

UNESP – UNIVERSIDADE ESTADUAL PAULISTA “JÚLIO DE MESQUITA
FILHO”
PROGRAMA DE PÓS-GRADUAÇÃO EM TELEVISÃO DIGITAL (PPGTVD)

Humberto Cecconi

UM SISTEMA DE RASTREAMENTO DE CÂMERAS PARA ESTÚDIOS VIRTUAIS

Bauru, SP
2013

Humberto Cecconi

UM SISTEMA DE RASTREAMENTO DE CÂMERAS PARA ESTÚDIOS VIRTUAIS

Trabalho de Conclusão de Mestrado apresentado ao Programa de Pós-Graduação em Televisão Digital (PPGTVD), da Faculdade de Arquitetura, Artes e Comunicação (FAAC), da Universidade Estadual Paulista “Júlio de Mesquita Filho” (UNESP), para obtenção do título de Mestre em Televisão Digital, sob orientação do Prof. Dr. Antonio Carlos Sementille.

Bauru, SP
2013

Humberto Cecconi

UM SISTEMA DE RASTREAMENTO DE CÂMERAS PARA ESTÚDIOS VIRTUAIS

Área de Concentração: Tecnologia e Televisão Digital
Linha de Pesquisa: 3 - Inovação tecnológica para televisão digital

Banca Examinadora:

Presidente/Orientador: Prof. Dr. Antonio Carlos Sementille
Instituição: Universidade Estadual Paulista Júlio de Mesquita Filho - UNESP - Bauru

Prof. Dr. João Fernando Marar
Instituição: Universidade Estadual Paulista Júlio de Mesquita Filho - UNESP - Bauru

Prof. Dr. Alexandre Cardoso
Instituição: Universidade Federal de Uberlândia - UFU

Resultado: _____

Bauru, _____/_____/_____

Dedico esta dissertação primeiramente a Deus e às pessoas que sempre estiveram ao meu lado: minha família, Patrícia, Priscila e Tayná que sempre me transmitiram fé, amor, alegria, paciência e coragem e souberam compreender as minhas ausências. Aos meus pais, Arnaldo e Sônia, pelo incentivo e força para alcançar o meu objetivo final.

AGRADECIMENTOS

Agradeço ao Prof. Dr. Antonio Carlos Sementille, orientador desta dissertação, por todo empenho, sabedoria, compreensão e dedicação.

À minha família, Patrícia, Priscila e Tayná, obrigado pela colaboração e por confiarem no meu trabalho e desempenho.

Aos meus pais, Arnaldo e Sônia, obrigado pelo carinho, incentivo, alegria e atenção sem reservas.

À minha Tia Angelina Cecconi, que sempre me teve como um filho e me apoiou nesta caminhada.

Aos coordenadores, professores e funcionários do Programa de Pós-Graduação em Televisão Digital, pela oportunidade de crescimento, aprendizado, realização profissional e pessoal e pela confiança em mim depositada.

Ao meu amigo e companheiro Sérgio Carlos Portari Júnior, que acompanhou de perto todo o desenvolvimento do meu trabalho. Convivi com ele boa parte desses últimos dois anos.

Aos meus colegas discentes do programa de pós-graduação em Televisão Digital, pela troca de experiências.

A todos que, em algum momento, dirigiram-me uma palavra de incentivo.

RESUMO

Cecconi, H. **Um Sistema de Rastreamento de Câmeras para Estúdios Virtuais**. Trabalho de Conclusão (Mestrado em TV Digital: Informação e Conhecimento)- FAAC - UNESP, sob a orientação do prof. Dr. Antonio Carlos Sementille, Bauru, 2013.

A produção virtual para a difusão via televisão digital é atualmente realizada por meio de estúdios virtuais. Para produções televisivas, que tenham elementos, tais como cenários virtuais ou objetos virtuais, é imprescindível mensurar, de forma precisa, a posição e orientação de cada câmera do estúdio, para que os elementos virtuais possam ser renderizados do ponto de vista correto. Considerando este contexto, o presente trabalho apresenta um sistema para rastreamento de câmeras que pode ser utilizado em estúdios de televisão, o qual baseia-se no reconhecimento óptico de marcadores fiduciais e dados obtidos de um sensor de rotação (giroscópio). No protótipo desenvolvido, foram utilizados os serviços da biblioteca de realidade aumentada conhecida como ARToolKit, bem como, o giroscópio presente no dispositivo Wiimote plus. Os resultados obtidos, a partir de testes com o protótipo, mostraram que o sistema proposto possui a acurácia necessária para ser efetivamente usado em produções televisivas.

Palavras-chaves: Televisão digital; Estúdios Virtuais; Rastreamento de Câmeras; Marcadores.

ABSTRACT

Cecconi, H. **A camera tracking system for virtual studios.** Dissertation (Master em Digital Television: Information and Knowledge) - FAAC - UNESP, guidance of professor Dr. Antonio Carlos Sementille, Bauru, 2013.

The virtual production for dissemination by way of digital television is currently performed by means of virtual studios. For television productions, which have elements such as virtual sets and virtual objects, it is essential to measure, precisely, the position and orientation of each camera studio, so the virtual elements can be rendered the right point of view. Considering this context, this work presents a system for tracking cameras that can be used in television studios, which is based on optical recognition of fiducial markers and data obtained from a rotation sensor (gyroscope). In the prototype, we used the services of the library called ARToolKit augmented reality, as well as the gyroscope present in the device Wiimote plus. The results from tests on a prototype have shown that the proposed system has the necessary accuracy to be effectively used in TV productions.

Keywords: Digital television; Virtual Studios; Tracking Cameras; Markers.

LISTA DE ILUSTRAÇÕES

Figura 2.1 - Exemplo de um giroscópio/acelerômetro em circuito integrado-----	20
Figura 2.2 - Estrutura básica de um acelerômetro.-----	21
Figura 2.3 - Composição de forças no estado de repouso.-----	22
Figura 2.4 - Composição de forças em queda livre. -----	22
Figura 2.5 - Efeito de Coriolis. -----	23
Figura 2.6 - Estrutura básica do giroscópio. -----	24
Figura 2.7 - Modelo de Câmera Pinhole.-----	25
Figura 2.8 – Exemplo de dispositivo utilizado em sistemas ópticos passivos, câmera de infravermelho. -----	27
Figura 2.9 – Conjunto de marcadores semiesféricos de 3mm. -----	27
Figura 2.10 Estrutura interna do Kinect, pode-se identificar o projetor infravermelho, a câmera colorida RGB e a câmera infravermelha. Konolige & Mihelich (2010) -----	29
Figura 2. 11- Visão superior do Leap Motion. -----	30
Figura 2. 12 - Área de movimento e toque do Leap Motion.-----	31
Figura 2. 13 - Imagem do Razer Hydra-----	31
Figura 3.1 – Protótipo de marcador preso a um stand.	35
Figura 3.2 - Arranjo utilizado na segunda abordagem para a medição da posição da câmera.	36
Figura 3.3 - Montagem experimental para medir a exatidão do sistema.	38
Figura 3.4 – Estrutura do sistema do projeto MATRIS.	40
Figura 3.5 - Câmera de TV com sensor inercial, câmera olho de peixe e sensor da lente.	41
Figura 3.6 - Vista da câmera principal (à esquerda) e da câmera olho de peixe (à direita) da mesma cena.	42
Figura 4.1 - ARToolkit em execução.	47
Figura 4.2 – Aquisição da Imagem e binarização do marcador	47
Figura 4.3 – Sistema de coordenadas do ARToolkit.	48
Figura 4.4 - Análise da precisão da biblioteca ARToolkit.	49
Figura 4.5 - Comparação das medidas do mundo real x capturadas pela câmera.	50
Figura 4.6 - Exemplos de diferentes resoluções de imagens.	51

Figura 4.7 - Controle remoto do Nintendo Wii.	52
Figura 4.8 - Circuito integrado do acelerômetro na placa do Wii.	53
Figura 4.9 - Eixos cartesianos e acelerações.	54
Figura 4.10 - Wii Motion Plus.....	55
Figura 4.11 - Câmera IR Wiimote.	56
Figura 5.1 - Ambiente virtual com webcam, wiimote e multimarcador.....	58
Figura 5.2 – Vista aproximada da webcam e o wiimote.....	58
Figura 5.3 - Diagrama de configuração do ambiente.	58
Figura 5.4 - Protótipo da webcam, Wiimote e laser de ponto.	61
Figura 5.5 - Multimarcador fixado no teto.	61
Figura 5. 6 - Diagrama de descrição do método.....	62
Figura 5.7 – Relação entre as coordenadas do marcador e da câmera, estimados pela análise da imagem.	63
Figura 5.8 – Layout dos marcadores múltiplos.	65
Figura 5.9 - Fluxograma do sistema desenvolvido.....	67
Figura 5.10 - Trecho de código do módulo de configuração.....	68
Figura 5.11 - Trecho de código do módulo de procura por marcadores.....	69
Figura 5.12 - Trecho de código do módulo de conversão de coordenadas.....	70
Figura 5. 13 - Trecho de código do módulo de obtenção dos quaternions.....	70
Figura 6.1 - Demarcação dos pontos no ambiente.....	73
Figura 6.2 - Aferição com trena metálica.	74
Figura 6.3 - Aferição com trena Ultrason.	75
Figura 6.4 – Trena de laser.	75
Figura 6.5 - Pontos demarcados no ambiente.	76
Figura 6.6 - Laser de linha.	76
Figura 6.7 - Ponto de marcação “ponto inicial”.	77
Figura 6. 8 - Tripé montado com o prumo.....	77
Figura 6.9 - Posicionamento no “marco 0” com o laser de linha.	78
Figura 6.10 - Demarcação dos 10 pontos em X e 10 pontos em Y.	79
Figura 6.11 - Multimarcador fixado no teto.	79
Figura 6.12 -Variação do resultado do eixo X entre os valores medidos, e o resultado obtido pelo software desenvolvido.	81

Figura 6.13 - Variação do resultado do eixo Y entre os valores medidos, e o resultado obtido pelo software desenvolvido.	82
Figura 6.14 - Variação do resultado do eixo Z entre os valores medidos, e o resultado obtido pelo software desenvolvido.	83
Figura 6.15 – Verificação dos ângulos.	84
Figura 6.16 - Orientação obtida pelo Wii Plus.	85
Figura 6.17 - Medidas dos ângulos da câmera (durante o movimento de rotação), representadas na forma de quaternion.	85
Figura 6.18 - Comparação do eixo X entre os quaternions do Wii e os quaternions da webcam.	86
Figura 6.19 - Comparação do eixo Y entre os quaternions do Wii e os quaternions da webcam.	87
Figura 6.20 - Comparação do eixo Z entre os quaternions do Wii e os quaternions da webcam.	87
Figura 10.1 - Distorção da lente e imagem do mundo real.....	94
Figura 10.2 - Imagem do arquivo calib_dist.pdf.	95
Figura 10.3 – Pontos interligados para cálculo da distorção calib_dist.pdf.	96
Figura 10.4 - Imagem do arquivo calicalib_cparam.pdf.....	97
Figura 10.5 - Posicionamento da câmera afastando de 10 em 10 cm.....	97
Figura 10.6 - Código do ARToolkit.	98
Figura 10.7 - Arquivo câmera_para.dat com erro em X 2,5 e Y 48.	98
Figura 10.8 – Código ARToolkit.....	99
Figura 10.9 - Correção do arquivo câmera_para.dat com os eixos ortogonais.....	99
Figura 11.1 – Novo marcador a partir do blankPatt.gif.	100
Figura 11.2 – Executando o programa de cadastro de marcadores.	101
Figura 11.3 – Cadastramento do marcador.....	101
Figura 12.1 - Tela inicial do software.....	102
Figura 12.2 - Resultados obtidos através do rastreamento.	103

TABELAS

Tabela 1 – Sistemas de rastreamento.....	32
Tabela 2 - Variação no Eixo X	81
Tabela 3 – Variação no Eixo Y.....	82
Tabela 4 – Variação no Eixo Z.....	83

SUMÁRIO

1	INTRODUÇÃO	15
1.1	OBJETO DA PESQUISA	16
1.2	OBJETIVOS	16
1.3	ORGANIZAÇÃO DA DISSERTAÇÃO	17
2	TECNOLOGIAS DE RASTREAMENTO.....	18
2.1	RASTREADORES ACÚSTICOS	18
2.2	RASTREADORES MAGNÉTICOS	19
2.3	RASTREADORES MECÂNICOS	19
2.4	RASTREADORES INERCIAIS.....	20
2.4.1	ACELERÔMETRO.....	21
2.4.2	GIROSCÓPIO	23
2.5	CÂMERA DIGITAL	24
2.6	MARCADORES	26
2.6.1	MARCADORES PASSIVOS	26
2.6.2	MARCADORES ATIVOS.....	28
2.7	AQUISIÇÃO DE DADOS DE CAPTURA.....	28
2.8	KINECT MICROSOFT	29
2.9	LEAP MOTION	30
2.10	RAZER HYDRA	31
2.11	CONSIDERAÇÕES FINAIS	32
3	TRABALHOS CORRELATOS	34
3.1	UM SISTEMA VERSÁTIL PARA RASTREAMENTO DE CÂMERA VOLTADO PARA PRODUÇÃO DE TV COM REALIDADE VIRTUAL.....	34
3.2	O RASTREAMENTO DE CÂMERAS EM TEMPO REAL NO PROJETO MATRIS.	39

3.2.1 MÓDULO INERCIAL.....	41
3.2.2 CÂMERA OLHO DE PEIXE	42
3.2.3 MODELAGEM DE CENAS OFF-LINE.....	42
3.2.4 ESTIMATIVA INICIAL.....	43
3.2.5 RASTREAMENTO BASEADO EM IMAGENS.....	43
3.2.6 SENSOR DE FUSÃO	44
3.3 CONSIDERAÇÕES FINAIS	45
4 FERRAMENTAS UTILIZADAS	46
4.1 A BIBLIOTECA ARTOOLKIT	46
4.1.1 ASPECTOS GERAIS DO FUNCIONAMENTO	47
4.1.2 PROCESSO DE RECONHECIMENTO	48
4.1.3 PRECISÃO DO PROCESSO DE DETECÇÃO	49
4.1.4 CONFIABILIDADE X DESEMPENHO.....	50
4.2 WII REMOTE.....	51
4.2.1 COMUNICAÇÃO BLUETOOTH.....	52
4.2.2 COMANDOS POR BOTÕES.....	53
4.2.3 ACELERÔMETRO.....	53
4.2.4 MOTION PLUS	55
4.2.5 CÂMERA DE INFRAVERMELHO	55
4.3 CONSIDERAÇÕES FINAIS	56
5 SISTEMA DESENVOLVIDO	57
5.1 ESTRUTURAÇÃO DO SISTEMA DE RASTREAMENTO DE CÂMERAS	57
5.2 ESTRUTURA DO SISTEMA	62
5.3 IMPLEMENTAÇÃO DO PROTÓTIPO DO SISTEMA DE RASTREAMENTO.....	66
5.3.1 IMPLEMENTAÇÃO DO TRATAMENTO DO FLUXO DE VÍDEO ..	67
5.4 CONSIDERAÇÕES FINAIS	72

6	TESTES E ANÁLISE DE RESULTADOS	73
6.1	TESTES REALIZADOS	73
6.2	PREPARAÇÃO DO AMBIENTE	73
6.3	TESTES EXECUTADOS	78
6.4	ANÁLISE DOS RESULTADOS EXTRAÍDOS	80
6.4.1	TESTES DE POSICIONAMENTO	80
6.4.2	TESTES RELATIVOS À ROTAÇÃO	84
6.5	CONSIDERAÇÕES FINAIS	88
7	CONCLUSÕES	89
	REFERÊNCIAS BIBLIOGRÁFICAS	90
	ANEXO	94
1.	CALIBRAÇÃO DE CÂMERAS	94
2.	CADASTRO DE MARCADORES	100
	APÊNDICE	102
1.	TELA DO INÍCIO DO SOFTWARE	102

1 INTRODUÇÃO

A ideia principal de produção virtual para televisão é a composição de elementos reais e virtuais em uma cena filmada. Para essa composição, diversos fenômenos ópticos devem ser harmonizados, utilizando-se diferentes tipos de mídias ou componentes de diversas fontes. Os mais importantes fenômenos são: combinar os parâmetros da câmera, no caso da composição de filmagem 2D, com componentes virtuais renderizados e estabelecer o apropriado sombreamento e iluminação dos diferentes componentes da cena (GRAU et al. 2002).

Atualmente, a produção virtual para difusão via televisão digital é realizada por meio de estúdios virtuais. Em um estúdio virtual, uma câmera filma um ou mais atores em um ambiente controlado. As imagens dos atores podem ser segmentadas em tempo real, usando-se técnicas de *chroma key*¹. A descrição resultante da forma dos atores é, neste caso, bidimensional. Para a realização de interações ópticas mais sofisticadas dos atores com o ambiente virtual, tais como oclusões e sombras, uma descrição 3D da cena é imprescindível.

Para produções de televisão, que tenham elementos tais como cenários virtuais ou objetos virtuais, é necessário, portanto, mensurar, de forma precisa, a posição e orientação de cada câmera do estúdio, para que os elementos virtuais possam ser renderizados do ponto de vista correto.

Um sistema de produção virtual 3D permite o movimento translacional da câmera em três dimensões, bem como *pan*, *tilt*, *roll* e *zoom*. Entretanto, isso demanda o uso de um sistema de rastreamento de câmera mais complexo.

Diversos tipos de rastreamento de câmera estão disponíveis no momento presente. Alguns são mecânicos, contando com câmeras em pedestais robóticos ou sobre trilhos. Outros dependem da análise de imagem para identificar marcadores colocados tanto na cena quanto no teto, (Thomas et al.1997), ou usam arranjos de marcadores, colocados na câmera e visualizados por outras câmeras existentes no estúdio, como o sistema da Orad, denominado Xync, (ORAD, 2012) que utiliza rastreamento por infravermelho, e o Free_D da Vinten, (Vinten, 2012) que utiliza marcadores passivos permanentes fixados fora da cena.

Segundo Gibbs & Baudisch (1996), nos estúdios virtuais são utilizados sistemas de rastreamento, que se dividem em duas categorias principais: eletromecânicos e óptica. Os

¹ *Chroma key* - O *Chroma key* substitui eletronicamente uma cor específica do fundo (em geral azul ou verde) por uma imagem vinda de outra fonte, como uma câmera, VT ou computador. Qualquer uma das três cores básicas pode ser usada para recorte, mas a azul e a verde são as mais utilizadas (Pizzotti, 2003).

sistemas eletromecânicos são usualmente mais utilizados, podendo ser divididos em duas subcategorias – ativa e passiva. No primeiro caso, para conduzir a câmera, são utilizados os conceitos de mecanismos de controle de servo. No segundo caso, os sensores detectam as ações do operador de câmera. Os sensores são muitas vezes "*encoders* ópticos" que detectam pequenas marcas (marcadores) em anéis ou em outras partes móveis. Apesar disso, referem-se a esses sistemas como eletromecânicos, para rastreamento de ativos e passivos.

A alternativa para rastreamento eletromecânico é o rastreamento óptico, que é baseado em reconhecimento de padrões. Ao colocar pontos de referência visíveis ou linhas de grade na tela azul do *chroma key*, utilizando-se de técnicas de processamento de imagem, pode-se extrair a posição, a orientação e o campo de visão da câmera.

Considerando o exposto, o presente trabalho tem como foco principal o estudo e o desenvolvimento de um sistema de rastreamento de câmeras para utilização em estúdios virtuais, visando as produções televisivas. O sistema desenvolvido utiliza-se dos serviços de reconhecimento de marcadores fiduciais passivos da biblioteca de Realidade Aumentada ARToolKit, com as informações obtidas de um giroscópio, para realizar o rastreamento da câmera de filmagem.

1.1 OBJETO DA PESQUISA

O objeto principal deste trabalho compreende o estudo de técnicas de rastreamento 3D, com ênfase naquelas aplicáveis ao rastreamento de câmeras em sets de filmagem, visando a inserção de objetos virtuais por meio do uso de Estúdios Virtuais.

1.2 OBJETIVOS

O objetivo principal deste trabalho é estruturar um sistema de rastreamento de câmeras de filmagem adequado à utilização em Estúdios Virtuais, valendo-se do estudo das tecnologias de rastreamento 3D existentes.

Como objetivos secundários tem-se:

- A implementação de um protótipo em software, elaborado de acordo com a estruturação proposta;
- A realização de testes utilizando o protótipo desenvolvido, visando a análise da acurácia.

1.3 ORGANIZAÇÃO DA DISSERTAÇÃO

No Capítulo 2, são apresentados os principais tipos de rastreadores com ênfase no princípio de rastreamento utilizado.

No Capítulo 3, é apresentado um levantamento dos principais trabalhos correlatos encontrados na literatura.

As principais características das ferramentas utilizadas na confecção do presente trabalho, a saber: a biblioteca ARToolkit e o controle remoto Wiimote são descritas no Capítulo 4.

O Capítulo 5 descreve, em detalhes, a estruturação e implementação do sistema desenvolvido.

Os componentes do ambiente experimental, os experimentos realizados e a análise dos resultados obtidos são apresentados no Capítulo 6.

Por fim, são apresentadas as conclusões e perspectivas para trabalhos futuros.

2 TECNOLOGIAS DE RASTREAMENTO

Neste capítulo, são apresentadas as principais tecnologias de rastreamento, bem como suas características e limitações.

O rastreamento tridimensional consiste, basicamente, em continuamente identificar os seis graus de liberdade (as translações e rotações nos três eixos coordenados), que definem a posição e orientação do objeto sendo rastreado, com relação a um determinado observador. Existem diversas tecnologias de rastreamento, porém todas têm suas fraquezas.

De acordo com Gomide (2006), os sistemas de captura de movimento podem ser divididos em quatro classes, dependendo do princípio físico empregado: sistemas ópticos, sistemas magnéticos, sistemas mecânicos e sistemas inerciais. Fundamentalmente, podem ser diferenciados pelo tipo de dados gerados, pelas restrições associadas e pelo custo total. As próximas seções descrevem as principais características de cada tipo de rastreador.

2.1 RASTREADORES ACÚSTICOS

Segundo Schreer et al. (2006), esse tipo de rastreamento utiliza fontes emissoras de som de alta frequência, que são captadas por microfones dentro do espaço de rastreamento. Ele tem duas configurações de implementação: na 1ª, a fonte de som é colocada no objeto a ser rastreado e os microfones estão espalhados na área de rastreamento (outside-in); na 2ª, pelo contrário, os microfones ficam no objeto e as fontes de som no espaço de rastreamento (inside-out). A forma de rastrear a posição dos objetos é medindo o tempo de voo dos pulsos de ultrassom, desde que saem dos emissores até sua chegada aos microfones. Esse tempo é então multiplicado pela velocidade do som, recuperando uma estimativa da distância dos receptores até a fonte. Com três ou mais receptores, fazendo triangulação, pode-se recuperar também a orientação do conjunto de receptores, como um único objeto rastreado.

Schreer et al. (2006) ressaltam, ainda, que as vantagens desses dispositivos são o baixo preço e o pouco peso que têm. Entre as desvantagens, tem-se a perda de precisão quando, no ambiente, existem objetos com superfícies que refletem as ondas de som, ou objetos que geram sons que atrapalham a identificação das ondas, como o som de telefones ou até do teclado. Esses dispositivos, também, apresentam uma baixa taxa de amostragem.

2.2 RASTREADORES MAGNÉTICOS

Esse tipo de tecnologia utiliza transmissores que emitem campos magnéticos de frequência baixa e pequenos refletores. De acordo com Schreer et al. (2006), a partir dos receptores, a posição e a orientação relativa do objeto rastreados são determinados em relação à fonte magnética. Lembrando que o alcance desse tipo de dispositivo varia de 1 a 10 metros. A precisão que se pode alcançar com ele é de 2 mm, em reconstrução da posição.

Os autores, também, afirmam que a desvantagem que tais dispositivos apresentam é que objetos condutores (metálicos) ou ferromagnéticos não podem estar presentes no espaço de rastreamento, no qual o dispositivo é colocado, porque eles causam distorção no campo magnético, tendo como consequência a perda de precisão na hora de recuperar os dados. Outra desvantagem é que os receptores, comumente, estão ligados por meio de cabos a um dispositivo que agrupa e gerencia seu funcionamento, o que cria desconforto para o usuário, especialmente quando ele se movimenta no espaço de rastreamento.

2.3 RASTREADORES MECÂNICOS

Para Schreer et al. (2006), esse tipo de rastreamento utiliza estruturas mecânicas rígidas e trabalha em conjunto com transdutores eletromagnéticos, como potenciômetros, codificadores de eixos e sensores de dobra para recuperar a posição e a orientação dos objetos rastreados. Alguns desses dispositivos devem ficar fixos no espaço de rastreamento, ou devem ser colocados no objeto a ser rastreado. No caso do rastreamento feito pelo usuário são geralmente colocados nas mãos ou na cabeça.

Segundo os autores, esses dispositivos oferecem duas vantagens: a precisão e a baixa latência. Além disso, alguns deles apresentam a característica de “*force feedback*”, que permite criar respostas táteis, por exemplo, com luvas que implementam força inversa para os movimentos das mãos. Isso ajuda a dar uma melhor sensação de imersão e realismo dentro do ambiente virtual onde se interage.

E a principal desvantagem é a restrição de mobilidade, que apresenta ao usuário, como consequência das estruturas rígidas, que compõem os dispositivos. Como exemplos, tem-se um dispositivo, baseado em uma estrutura fixa em forma de braço, que serve de ponto de referência para a recuperação da posição e rotação do usuário, e um dispositivo de resposta

tátil “*force feedback*”, apresentado como uma luva mecânica, que gera forças inversas nos dedos, como resposta para a sensação de pegar um objeto com volume.

2.4 RASTREADORES INERCIAIS

Os sistemas Micro-eleto-mecânicos (*Micro-Electro-Mechanical Systems*, ou *MEMS*) caracterizam uma tecnologia que, na sua forma mais geral, pode ser definida como miniaturas de elementos mecânicos ou eletromecânicos. São dispositivos e estruturas feitos utilizando técnicas de microfabricação. Da mesma forma, os dispositivos MEMS podem variar desde estruturas relativamente simples, sem elementos móveis, até elementos extremamente complexos de sistemas eletromecânicos, como ilustrado na (Figura 2.1) (*MEMS and Nanotechnology Exchange*).

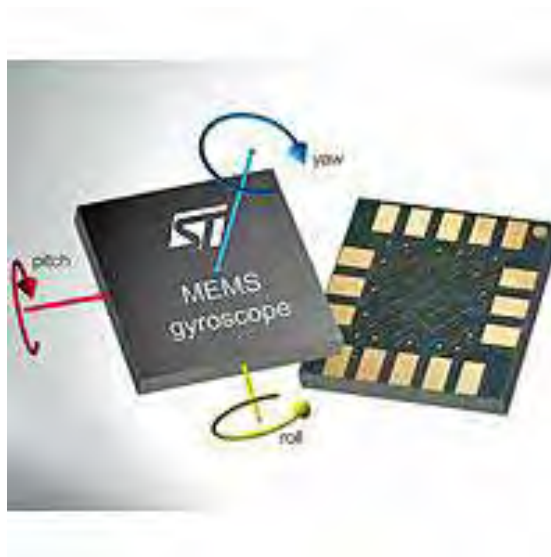


Figura 2.1 - Exemplo de um giroscópio/acelerômetro em circuito integrado

Fonte: Datasheet MENS, 2012.

Os dispositivos MEMS mais conhecidos são:

- Acelerômetro;
- Giroscópio;
- Magnetômetro;
- Micromotor;
- Interruptor óptico.

2.4.1 ACELERÔMETRO

Tal como o nome sugere, o acelerômetro serve para medir acelerações de um corpo. Ele pode ser construído de várias maneiras, mas todas mantendo uma estrutura básica que consiste em uma massa inercial suspensa por uma mola, (Figura 2.2). A diferença entre os vários acelerômetros está essencialmente na detecção da posição relativa da massa inercial, quando lhe é aplicada uma força exterior.

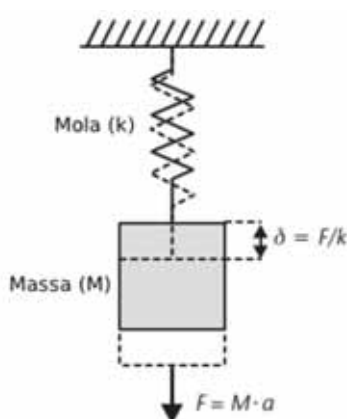


Figura 2.2 - Estrutura básica de um acelerômetro.

Um dos métodos mais comuns de detecção dessa posição relativa é o método capacitivo, que consiste em ter um condensador entre a massa e a estrutura do acelerômetro, para que, quando houver movimento, a massa faça mover a parte móvel do condensador e assim variar a sua capacitância. Esse tipo de sistema requer o uso de circuitos eletrônicos especiais para detectar mudanças minúsculas de capacitância ($<10^{-15}$ F) e convertê-lo em uma tensão de saída amplificada.

Outro método comum é o uso de piezorresistências para medir as tensões internas induzidas na mola que segura a massa. Uma variante desse tipo de aplicação é o material da mola ser de piezoelétrico ou conter um filme fino de piezoelétrico, fornecendo uma tensão em proporção direta ao deslocamento (Maluf & Willians, 2004).

Os acelerômetros, na realidade, não medem as acelerações nem as forças que lhes são diretamente aplicadas, mas a reação à força que lhe é aplicada, o que pode ser observada mediante dois exemplos práticos.

No primeiro exemplo, o sensor está em repouso, sobre uma mesa (Figura 2.3), e a informação enviada deverá ser uma aceleração de $9,81\text{m/s}^2$.

Para obter a aceleração provocada pela força, recorre-se à segunda lei de Newton.

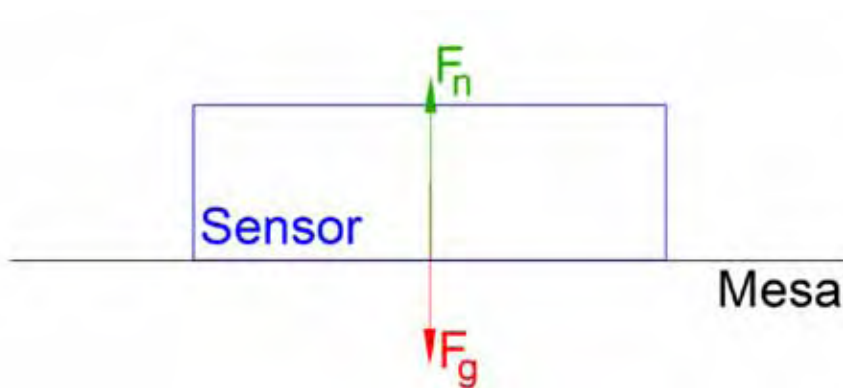


Figura 2.3 - Composição de forças no estado de repouso.

Fonte: adaptado de Beeby et al., 2004.

No segundo exemplo, o acelerômetro está em queda livre ideal (sem o efeito da resistência do ar), conforme as condições da (Figura 2.4).

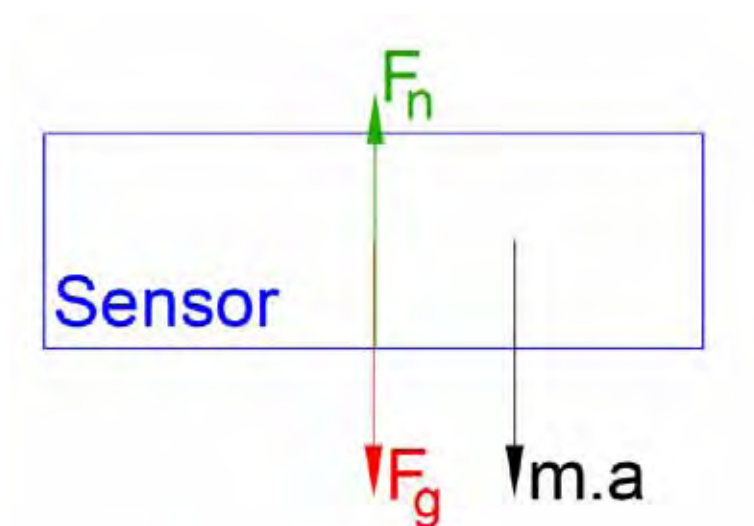


Figura 2.4 - Composição de forças em queda livre.

Como está em queda livre, a aceleração é igual à aceleração provocada pela força da gravidade, “g”.

2.4.2 GIROSCÓPIO

O método de cálculo da velocidade angular é semelhante para a maioria dos giroscópios e tem como base o efeito de Coriolis.

Quando uma massa (m) está com movimento, na direção da velocidade linear e com velocidade angular, ela vai sofrer uma força na direção da resultante da direção da velocidade linear com o eixo da velocidade angular. Essa força (figura 2.5) é intitulada de força de Coriolis (Beeby et al, 2004).

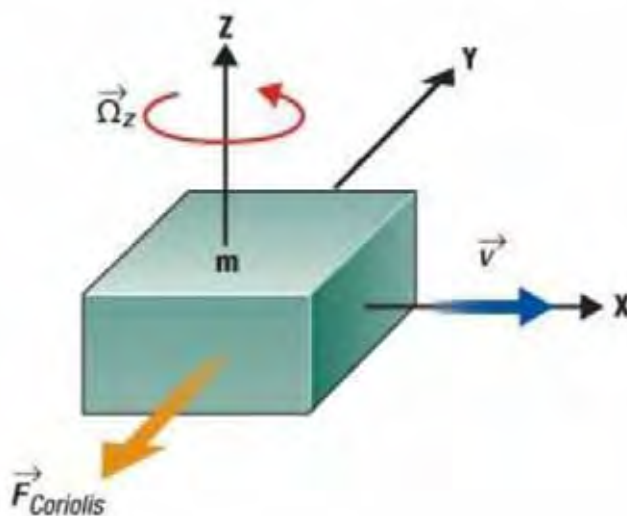


Figura 2.5 - Efeito de Coriolis.

A maioria dos giroscópios MEMS, disponíveis no mercado, têm em comum uma estrutura tipo forquilha. Existem duas massas a oscilar e a moverem-se constantemente em direções opostas (Figura 2.6). Quando a velocidade angular é aplicada, a força de Coriolis, em cada massa, também atua em direções opostas, resultando na mudança de capacitância. A diferença na capacitância é proporcional à velocidade angular (Beeby et al, 2004). Entretanto, se existir apenas movimento linear, as massas deslocar-se-ão para a mesma direção. Deste modo, a diferença na capacitância será zero.

do sensor, amenizando ao máximo a interferência gerada. Vale lembrar que apesar de o custo de produção de um sensor CMOS ser mais baixo, porém está mais suscetível a ruído.

O modelo de câmera *pinhole*, exibido pela Figura 2.7, explica, de maneira simplificada, o processo de obtenção de uma imagem em uma câmera. Na figura, a câmera é uma caixa escura, com um pequeno orifício que deixa passar alguns raios de luz. Desconsiderando a natureza atual da luz, tratando-a somente como partícula, os raios de luz propagam-se em linha reta, passando pelo centro de projeção e (orifício da caixa) formando a imagem invertida do objeto em um plano do lado oposto. A distância do objeto s e o comprimento da caixa c influenciam na altura h' da imagem formada. Por semelhança de triângulos, a Equação (Eq 1.1) quantifica essa influência. Quanto maior a distância do objeto, menor é sua imagem e quanto maior for a distância entre o orifício da caixa e seu fundo, maior será a imagem formada.

$$h' = h \frac{c}{s} \quad (\text{Eq. 1.1})$$

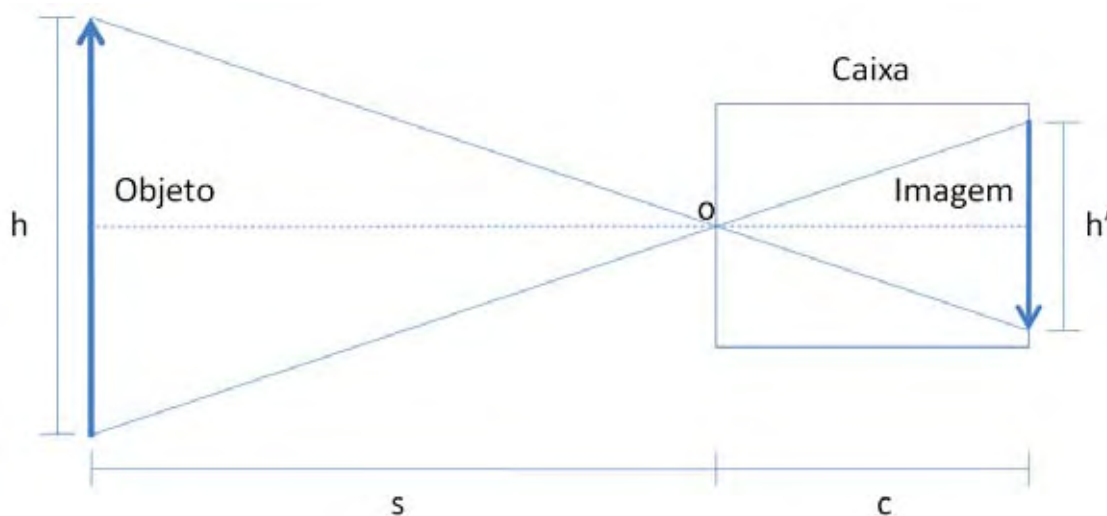


Figura 2.7 - Modelo de Câmera Pinhole.

Infelizmente, no mundo real, uma câmera pinhole não é prática. A principal razão é a pequena quantidade de luz, que passa pelo seu orifício. Ela é insuficiente para formar uma imagem de luminosidade adequada em um curto espaço de tempo. Logo, o tempo de exposição deve ser extremamente elevado para se obter uma imagem de qualidade. Certas aplicações se tornam impossíveis, como fotos de objetos em movimento ou vídeos; entretanto, é possível, com o uso de lentes, solucionar esse problema.

Lentes são objetos feitos geralmente de vidro ou plástico, capazes de divergir ou convergir raios de luz. Elas funcionam com base no conceito de refração, em que um raio de luz muda de velocidade ao entrar em um meio diferente do que está. O uso de uma lente convergente torna possível concentrar uma quantidade de luz suficiente para a obtenção de uma imagem, em um curto espaço de tempo. Idealmente, os raios de luz de um ponto do objeto devem convergir para um mesmo ponto de foco, ao passar pela lente, mas isso não acontece precisamente na realidade. Algumas aberrações e distorções podem surgir por causa desse fato e deterioram a qualidade da imagem final (Mahajan, 1998).

2.6 MARCADORES

A captura de movimento pode ser feita usando ou não marcadores. Marcadores são objetos que auxiliam na determinação ou efetivamente determinam os pontos a serem gravados ou analisados em uma seção de captura. É possível classificá-los em passivos ou ativos, dependendo de sua reatividade com o mundo externo.

2.6.1 MARCADORES PASSIVOS

Marcadores passivos são aqueles que apenas refletem algum tipo de onda, devido a uma propriedade natural do material do qual são constituídos. Eles não realizam qualquer tipo de comportamento específico baseado no impulso externo recebido, ou seja, somente auxiliam na segmentação, na separação de uma região de interesse do fundo.

O tamanho e o formato dos marcadores dependem, principalmente, do tamanho do volume de captura, da resolução das câmeras e do tipo de movimento a ser capturado (corporal, facial, etc.). Marcadores esféricos com cerca de 1,6 cm ou mais de diâmetro são comumente empregados na captura de movimentos de corpo inteiro, enquanto marcadores semiesféricos com aproximadamente 3 ou 4 mm de diâmetro são utilizados na captura de movimentos faciais. A Figura 2.8 ilustra uma câmera de infravermelho (comercializada pela Vicon).



Figura 2.8 – Exemplo de dispositivo utilizado em sistemas ópticos passivos, câmera de infravermelho.

Fonte: Vicon (2008).

O ponto de interesse em si é obtido por meio do cálculo do centroide da região por um software. É necessário ainda definir sua semântica em uma etapa de inicialização e rastreá-lo precisamente no tempo. Um exemplo bastante difundido dessa categoria (Figura 2.9) são bolinhas de borracha revestidas de um material reflexivo, iluminadas por lâmpadas que emitem luz no comprimento de onda infravermelho (Vicon Motion Systems, 2008).

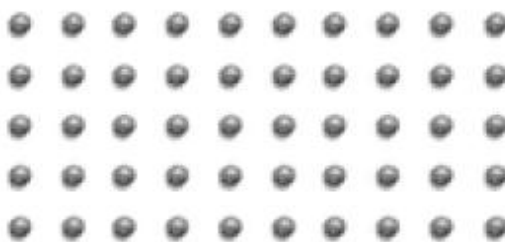


Figura 2.9 – Conjunto de marcadores semiesféricos de 3mm.

Fonte: Vicon (2008).

Em resumo, as principais vantagens dos sistemas ópticos passivos são:

- Possibilitam altas velocidades de captura e altos níveis de precisão e exatidão;
- Os movimentos de múltiplos sujeitos podem ser capturados simultaneamente;
- Um alto número de marcadores pode ser usado e a configuração dos mesmos pode ser alterada facilmente;
- O objeto ou indivíduo pode se movimentar livremente dentro do volume de captura, não sofrendo obstrução por fios ou equipamentos que limitam os movimentos;
- O volume de captura pode ser grande o suficiente para vários tipos de aplicações.

As principais desvantagens desses sistemas são:

- Um pós-processamento intenso dos dados pode ser necessário;

- Possibilidade de oclusão dos marcadores, resultando em perda de dados;
- Hardware e software geralmente mais caros do que os de outros sistemas;
- A maioria dos sistemas necessita que a luz do ambiente seja controlada.

2.6.2 MARCADORES ATIVOS

Marcadores ativos são aqueles que realizam algum tipo de processamento sobre impulsos externos recebidos ou que emitem impulsos. Necessariamente, possuem algum tipo de dispositivo, como acelerômetro, potenciômetro, giroscópio, sensor acústico, sensor de luz ou diodos emissores de luz (do inglês *Light Emitting Diodes* — LEDs) para poderem realizar cálculos e responder adequadamente ao estímulo recebido. No mínimo, são capazes de se identificar, como o sistema produzido pela *PhaseSpace* Inc. (2008), que modula singularmente a frequência de luz emitida por cada LED. Portanto, não requerem uma longa etapa de inicialização e o rastreamento é bastante simplificado. Além disso, marcadores ativos podem ser capazes de determinar sua posição tridimensional e rotações sofridas ao longo do tempo (Raskar et al., 2007).

Existem diversos tamanhos, tecnologias e modelos de marcadores. Entretanto, a entidade alvo da captura limita muitas vezes a escolha do tipo de marcador utilizado. Isso ocorre especialmente quando se deseja capturar o movimento detalhado de uma área relativamente pequena, como uma face. Atualmente, somente marcadores passivos podem ser usados nesse caso, tendo em vista seu tamanho e peso reduzidos. Outro fator restritivo é o ambiente em que a captura ocorre. Dependendo das condições existentes, como luz natural, interferências e meio físico, marcadores ativos com giroscópios e acelerômetros podem ser mais adequados. Ainda, outra limitação é em relação ao tipo do movimento capturado. Movimentos que envolvam quedas são mais adequados para marcadores passivos, pois não possuem frágeis componentes eletrônicos.

2.7 AQUISIÇÃO DE DADOS DE CAPTURA

Segundo Silva (1998), dois são os modos de se adquirir dados em uma captura de movimento: o direto e o indireto. Aquisição direta significa não necessitar de uma etapa de pós-processamento para realizar cálculos adicionais sobre os dados gerados. Ou seja, dados adquiridos desse modo podem ser usados exatamente da mesma forma que foram capturados,

sem qualquer intervenção. Sistemas desse tipo são próprios para aplicações de tempo real, apesar de serem mais intrusivos.

Dados adquiridos indiretamente precisam ser tratados por um software que calcula informações de rotação e de posição relativa dos pontos de interesse. Somente assim, os dados podem ser usados. As vantagens existentes em sistemas que capturam dados desse modo são, normalmente, a precisão e a velocidade de captura, já que parte das informações necessárias pode ser obtida por meio de processamento posterior.

2.8 KINECT MICROSOFT

Em Kühn (2011), o Kinect é descrito como sendo um dispositivo composto por três estruturas ópticas: um projetor laser infravermelho, uma câmera infravermelho e uma câmera colorida.

Uma característica específica do Kinect é a combinação do projetor e da câmera, ao invés da utilização de duas câmeras como é frequente acontecer em equipamentos para visão estereoscópica. A vantagem desta solução é a diminuição da quantidade de processamento necessário para o funcionamento do equipamento.

Seu custo é relativamente baixo e está disponível facilmente em lojas de eletrônicos do gênero.



Figura 2.10 Estrutura interna do Kinect, pode-se identificar o projetor infravermelho, a câmera colorida RGB e a câmera infravermelha. Kühn (2011)

2.9 LEAP MOTION

O Controlador de movimento Leap Motion, figura 2.11 é um pequeno dispositivo USB que é projetado para ser colocado em desktop ou notebook, para o funcionamento do dispositivo é necessário que o dispositivo fique em uma superfície plana e com a face voltada para cima.

Utilizando-se de 2 câmeras e três LEDs de infra vermelho, o dispositivo rastreia uma área aproximadamente hemisférica, a uma distância de cerca de 1 metro. O Leap Motion é projetado para rastrear os dedos, ou itens semelhantes, que se cruzem na área rastreada, com uma precisão de cerca de 0,01mm.



Figura 2. 11- Visão superior do Leap Motion.

Fonte: Leap Motion Inc.

A área de observação de menor e de maior resolução do dispositivo diferencia o produto a partir da cinética, que é mais apropriada para o seguimento de corpo inteiro, num espaço do tamanho de uma sala.

A sensibilidade de rastreamento do sensor é alta, permitindo movimentos rápidos com boa precisão, O dispositivo forma uma espécie de “bolha” virtual dividido em duas partes:

Hover Zone: é onde pode-se mover o cursor à vontade, sem se preocupar em com o clic, Localiza-se entre o usuário e o sensor.

Touch Zone: é onde se move as mãos para executar os movimentos podendo efetuar o clic. Esta área localiza-se depois do sensor, mais próximo do monitor. Figura 2.12

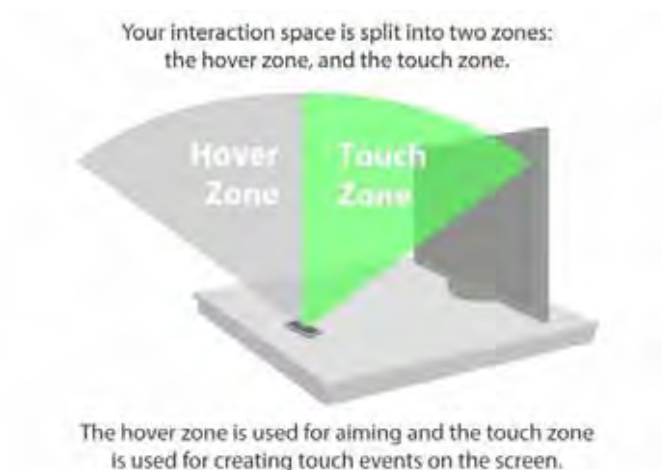


Figura 2. 12 - Área de movimento e toque do Leap Motion.

Fonte: Leap Motion Inc.

2.10 RAZER HYDRA

O Razer hydra é um joystick e foi desenvolvido pela Sixense Entertainment, com parceria da Razer EUA.

O Razer Hydra é composto de dois controladores analógicos, cinco botões de resposta rápida e dois botões de disparo na parte de traz, e uma estação base esférica Figura 2.13. A estação base tem uma esfera redonda que detecta os controles, a base se conecta ao PC através de uma interface USB, e os controles de movimentos se conectam a base através de fios.



Figura 2. 13 - Imagem do Razer Hydra.

Fonte: Sixense Entertainment

A detecção dos movimentos, da orientação e da posição é feita através de sensores magnéticos com uma precisão de 1mm e 1 grau, tornando o joystick bem preciso.

2.11 CONSIDERAÇÕES FINAIS

As características dos dispositivos de rastreamento estão na tabela 1.

Alguns dispositivos como Kinect, Leap Motion e Razer Hydra, não estão descritos na tabela, pois possuem diversas tecnologias no mesmo equipamento.

O Kinect possui rastreamento através de câmeras e LEDs infra vermelho, embora ter uma precisão alta, a utilização em estúdios virtuais o torna inviável pois os feixes infra vermelho são visualizados pela câmera principal, aparecendo os feixes na imagem.

O Leap Motion por sua vez tem uma área de rastreamento muito pequena, inviabilizando o rastreamento de câmeras em estúdios.

Já o Razer Hydra por trabalhar com sensores magnéticos, e presos a fios também se torna inviável para o rastreamento de câmeras em estúdios.

Dependendo da aplicação, alguns fatores deverão ser levados em consideração para o uso do equipamento.

Tabela 1 – Sistemas de rastreamento.

	Sistema Mecânico	Sistema Inercial	Sistema Óptico	Sistema Acústicos	Sistema Magnético
Resolução espacial	Independente do alcance, ~ 0.5 - 1 grau	3000 g for 0.5 ms	~ 0.0012 a 0.2 % do FOV ²	78+ [-66] V/(ms) dB	~ 0.005 a 8mm; ~ 0.025 a 0.1 graus
Acurácia espacial	<= 5 graus	10000 g for 0.1 ms	~ 0.004 a 0.5 % do FOV	1000 ft	~ 0.8 a 5mm; 0.1 a 3 graus
Precisão	de média elevada	elevada	~ 0.0055 a 0.02 % do FOV	média elevada	elevada
Taxa de amostragem	> 120 FPS	0 - 140 FPS	> 200 FPS	Frequência ultrassônica (10KHz a 2000 KHz)	~ 100 FPS
Campo de Trabalho	Ilimitado (depende apenas do comprimento dos cabos de transmissão)	de 1 a 10 m (conexão Bluetooth)	raio de 1 a 4 m	limitado ao tamanho do cabo	raio de 1 a 2 m
Exemplos de sistemas	Animatron, Digital Monkey, Power Glove	Free-D, Matris, Motion Capture, Siemens, Telefones, Tablets, Games	ExpertVision HiRES 3-D System, Multitrax Motion Capture System, OPTOTRAK	Sistema proprietário com interface externa	A Flock of Birds, ULTRATRAK, FASTRAK, INSIDETRAK
Softwares compatíveis	Alias Wavefront, Jack, 3D Studio Max SoftImage	Biblioteca para diversos sistemas operacionais, WDK,	ZoeTrax, The Creative Motion Editor, ARToolKit, PowerAnimator V7	_____	Alias Wavefront, 4Dvision, SoftImage, Kinemation 3.0, 3D Studio MAX, PowerAnimator V7, Photo 4D

Fonte: Adaptado de Velho et. al 1996.

A escolha por utilizar os sistemas ópticos e inerciais vem devido à precisão dos equipamentos e da facilidade de interligação com a estação de trabalho. Lembrando que as

conexões sem fio dão maior mobilidade aos equipamentos de rastreamento escolhidos. Outro ponto importante dos sistemas ópticos e inerciais é a compatibilidade com os softwares disponíveis no mercado e a acurácia do rastreamento.

Diferentes sistemas de rastreamentos, tais como mecânicos, acústicos e magnéticos têm como restrições equipamentos presos ao corpo do ator, limitação de cabos e interferências a ruídos elétricos, eletrônicos e magnéticos.

3 TRABALHOS CORRELATOS

Neste capítulo, são apresentados os principais trabalhos encontrados na literatura, os quais dizem respeito ao rastreamento de câmeras em produções televisivas.

3.1 UM SISTEMA VERSÁTIL PARA RASTREAMENTO DE CÂMERA VOLTADO PARA PRODUÇÃO DE TV COM REALIDADE VIRTUAL.

De acordo com Thomas et al. (1997), a diferença essencial entre um estúdio convencional e um estúdio com *chroma key* é a capacidade de mover a câmera, mantendo o registro correto, entre o primeiro plano e o fundo virtual.

Para os autores, um sistema de posicionamento de medição da posição da câmera, idealmente deve:

- Permitir a circulação sem restrição da câmera em uma área de cerca de 800 metros quadrados;
- Trabalhar com uma grande variedade de suportes de câmeras;
- Medir a posição e a orientação de uma precisão suficiente para não introduzir ruídos consideráveis nas posições dos elementos virtuais. Por exemplo, para manter as posições relativas de uma precisão de + ou - 0,5 pixel com um mínimo do campo de visão de 10 graus ou 50 cm, requer uma precisão de posicionamento na ordem de + ou - 1mm, e a precisão angular de + ou - 0,01 graus;
- Medir os parâmetros da câmera com o mínimo de atraso, pois longos atrasos tornam a navegação no mundo real muito difícil para o cinegrafista;
- Não colocar restrições significativas em cada cena de conteúdo ou no ambiente de estúdio.

Os autores ressaltam, ainda, que há no mercado inúmeros sistemas disponíveis, que fornecem dados de posição da câmera, mas nenhum deles cumpre todos os requisitos.

Neste trabalho, Thomas et al. (1997) descrevem duas abordagens que desenvolveram, sendo a primeira mais adequada para instalações temporárias e pré-gravação de produção, enquanto a segunda é mais adequada para programas ao vivo e produções de televisão, porém requer maior tempo para a instalação.

Após examinarem inúmeras possibilidades, os autores consideraram mais adequado um sistema óptico com base em marcadores colocados na cena. A imagem é analisada para localizar a posição exata de cada marcador. Só assim a posição e orientação da câmera podem ser calculadas, considerando-se as posições conhecidas dos marcadores no estúdio.

Na pós-produção, a posição e a orientação da câmera podem ser recalculadas por análise das imagens, proporcionando medições mais precisas. Ressalta-se, ainda, que os parâmetros intrínsecos da câmera não só devem ser conhecidos, como também podem ser determinados por um processo de calibração da câmera feito separadamente, uma vez que permanecem fixos a escolha da lente. Deve-se esclarecer que uma restrição a esse sistema deve ser levada em conta: é que pelo menos um e, de preferência, três marcadores precisam estar visíveis em todos os momentos.

Como os marcadores podem ser reposicionados para atender a cada cena, era importante incluir um método para medir, automaticamente, suas posições para se obter uma alta precisão. Isso foi conseguido por meio da captura de uma série de imagens, cada uma apresentando dois ou mais marcadores. Lembrando que as identidades dos marcadores, em cada imagem, são determinadas automaticamente.

Cada marcador compõe-se de oito tons de azul com uma cor ligeiramente diferente, ao fundo, de *chroma key*.

Os marcadores foram concebidos de modo que os círculos não sejam coplanares, o que permite que a posição da câmera possa ser calculada sem ambiguidade, com base em uma imagem que contém um único marcador e facilita também a medição automática das posições dos marcadores. O marcador, mostrado na figura 3.1, possui, no centro, quatro círculos montados a 10cm à frente dos círculos exteriores. As dimensões dos marcadores são 40x30x10cm. Cada marcador tem um código de barras para identificá-lo de forma exclusiva.

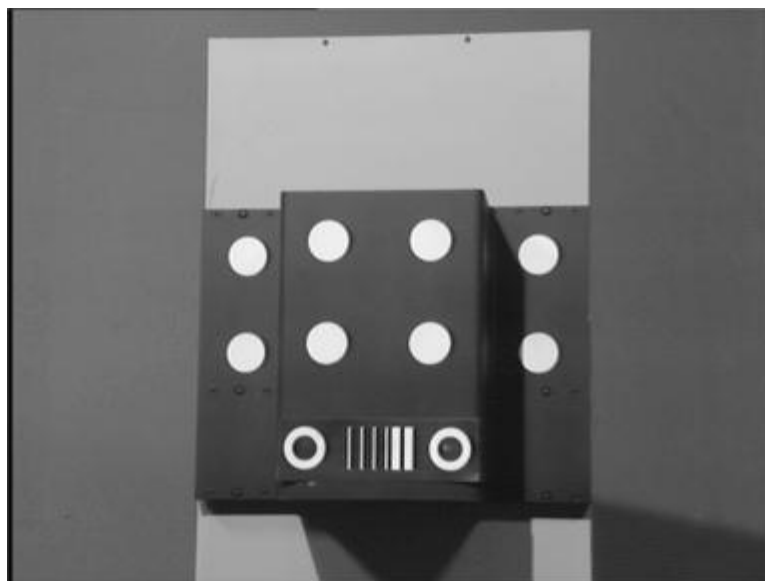


Figura 3.1 – Protótipo de marcador preso a um stand.

Fonte: Thomas et al. (1997).

Durante o desenvolvimento do sistema descrito, surgiu a necessidade de um sistema mais adequado para o uso em estúdios. Nessa abordagem, os marcadores, além de serem visíveis na cena, podiam ser inseridos, uma vez que se diferenciavam por tons de azul, sendo aceitáveis em produções de filmes com pós-produção, mas não sendo possível a sua utilização em transmissões ao vivo. Além disso, o sistema era prejudicado caso os marcadores fossem obstruídos, perdendo, assim, o controle da posição da câmera, demorando alguns segundos para serem recalculados. Em vista disso, um sistema alternativo foi concebido pelos autores. Aqueles marcadores foram substituídos por marcadores montados fora da cena, por exemplo, no teto, e uma pequena câmera auxiliar foi acoplada à câmera principal. Vale observar que a principal desvantagem desse sistema é o tempo de montagem pois, além de os marcadores precisarem ser posicionados no teto do estúdio, há, ainda, o aumento do custo e da complexibilidade, em virtude da inserção de uma câmera auxiliar.

No entanto, existem várias vantagens para a utilização do método com uma câmera auxiliar voltada para o teto. Como os marcadores não são mais visíveis pela câmera principal, as restrições sobre a cena são mais brandas, pois não é mais necessário a inserção de marcadores com tons diferentes de azul e nem mesmo a necessidade da utilização do *chroma key*. Além disso, os marcadores podem ser de alto contraste, tornando possível identificá-los facilmente.

A figura 3.2 mostra um exemplo da disposição de um estúdio. Alguns limites típicos entre a câmera e o teto do estúdio devem ser considerados; os marcadores podem ser posicionados de forma coplanar, porém a posição da câmera é significativamente melhor, se os marcadores forem colocados em distâncias diferentes da câmera.

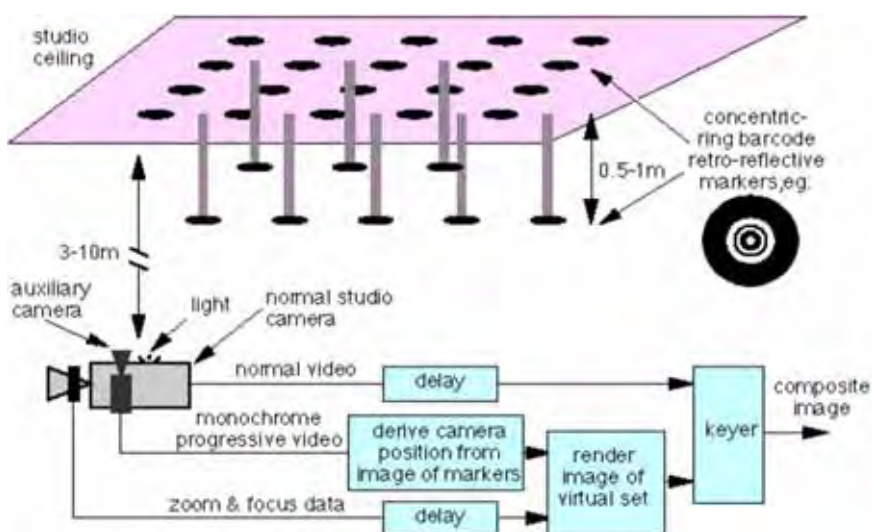


Figura 3.2 - Arranjo utilizado na segunda abordagem para a medição da posição da câmera.

Fonte: Thomas et al. (1997).

A câmera auxiliar precisa ter seus parâmetros intrínsecos medidos com precisão. Isso pode ser realizado quando o sistema estiver sendo instalado e calibrado, e somente precisa ser repetido, periodicamente, para garantir que a calibração não mudou, ou quando a lente é substituída. A orientação relativa, a posição da câmera auxiliar e a câmera principal, também, precisam ser medidas.

No sistema desenvolvido, foi demonstrado que o desenho do marcador utiliza círculos individuais em vez de grupos de oito, para simplificar a fabricação e a montagem. A codificação de cada marcador é incorporada dentro do círculo em si, sob a forma de um conjunto de anéis concêntricos. O número de bits, no código de barras, pode ser escolhido para se adequar ao tamanho do estúdio. Segundo os autores, um código de 9bits é suficiente para identificar, de modo único, cada marcador num estúdio de área de cerca de 50 metros quadrados. Se os estúdios forem maiores, podem ser acrescentados mais bits. Lembrando que o tamanho mínimo dos marcadores deve ser escolhido para assegurar que o código de barras possa ser lido com êxito pela câmera auxiliar.

Os autores, ainda descobriram que o desenho de código de barras e um algoritmo exigem um espaçamento mínimo, entre os anéis correspondentes a cerca de 1,5 pixel no sensor da câmera, para permitir uma leitura confiável. Assim, se os marcadores estavam a aproximadamente 4 m acima da câmera auxiliar, cada um contendo um código de 9 bits, e a câmera com resolução convencional com um mínimo de campo de visão de 30 graus, o marcador deve ser de aproximadamente 12cm.

Para que os autores avaliassem em um estúdio a precisão dos resultados experimentais, foi criado um arranjo com 24 marcadores montados em um lado do estúdio, 16 marcadores montados sobre a parede, e outros oito marcadores montados cerca de 40 cm em frente à parede. Suas posições foram medidas e, então, refinadas mediante a análise de 14 imagens, que mostram os marcadores vistos de várias posições e ângulos. A câmera auxiliar foi então montada numa bancada, de modo que ela pudesse ser movida em linha reta, mantendo o ângulo fixo, como é mostrado na (Figura 3.3).



Figura 3.3 - Montagem experimental para medir a exatidão do sistema.

Fonte: Thomas et al. (1997).

O ruído aleatório na posição do ângulo foi avaliado com a câmera em uma posição fixa, com 15 marcadores. Os desvios - padrão da posição e medidas de ângulo são calculados com base em 100 imagens consecutivas, conforme mostra a Tabela 1. Esses estão bem dentro dos requisitos de precisão, discutidos anteriormente pelos autores.

Tabela 1:

Desvios-padrão da posição medida e os ângulos de uma câmera estacionária.

	angle (degrees)	position (mm)
x	0.0020	0.073
y	0.0015	0.090
z	0.00072	0.030

Fonte: Thomas et al. (1997).

Foram utilizados dois métodos para calcular a posição de uma câmera para aplicações em estúdios virtuais. Ambos são adequados, segundo os autores, para câmeras de mão em estúdios grandes. Um método utiliza marcadores em cena, e outro utiliza marcadores fora de cena, no teto, visto por uma câmera auxiliar. De acordo com eles, o método utilizando uma câmera auxiliar parece ser mais adequado para a maioria das aplicações. O sistema permite que as técnicas de produção virtual sejam utilizadas em estúdios grandes, sem impor qualquer restrição sobre o tipo de montagem da câmera. Também, pode ser utilizado para outras aplicações que exijam a medição precisa da posição e orientação de um objeto.

3.2 O RASTREAMENTO DE CÂMERAS EM TEMPO REAL NO PROJETO MATRIS.

O trabalho, denominado Projeto MATRIS, foi desenvolvido por Chandaria et al. (2006). Segundo os autores:

“A fim de inserir um objeto virtual em uma imagem de TV, o sistema gráfico precisa saber exatamente como a câmera está em movimento, para que o objeto virtual possa ser processado no lugar correto em cada frame. Hoje em dia, isto pode ser conseguido com relativa facilidade na pós-produção, ou em um estúdio equipado com um sistema de rastreamento especial. No entanto, para a gravação ao vivo no local, ou em um estúdio que não está especialmente equipado, a instalação de um sistema deste tipo pode ser difícil ou não rentável.”

Com o propósito de superar essas limitações, desenvolveu-se o projeto MATRIS com a finalidade de medir o movimento de uma câmera em tempo real. Esse sistema não só utiliza a análise de imagens para acompanhar características que ocorrem na cena, como também faz uso de dados de um sensor inercial.

Convém ressaltar que nenhum marcador especial ou suporte de câmera é necessário. O que é interessante neste projeto é o uso da metáfora em relação aos sentidos humanos, exemplo, o labirinto seria o sensor inercial, e os olhos a imagem da câmera.

Para produções de televisão, que utilizam objetos virtuais em cena, é imprescindível medir, com precisão, a posição e a orientação de cada câmera de estúdio, de modo que os elementos virtuais possam ser processados exatamente a partir da perspectiva direta. Programas que inserem os objetos virtuais na pós-produção podem utilizar-se de recursos da imagem pronta, porém é um processo muito mais lento.

Também para programas de televisão e produções de filmes, a realidade aumentada tem muitas aplicações, tais como, indústria, medicina, educação e jogos. A ideia é adicionar objetos virtuais em uma cena, em tempo real. Para que os efeitos pareçam ser reais, os objetos virtuais devem aparecer rigidamente fixados no mundo real, o que requer a medição precisa da posição da câmera em tempo real.

Para Chandaria et al. (2006), o ambiente em que o sistema será utilizado é modelado para proporcionar um conjunto de imagens e as características que podem ser utilizadas como balizas pelo processo. Vale lembrar que o projeto MATRIS considerou as

texturas e as características de linha. As características, que são modeladas, são escolhidas para ser visíveis a partir de uma grande variedade de pontos de vista, e para conter detalhes suficientes para permitir a sua localização a ser determinada com acurácia.

Os autores ressaltam, ainda, que, as características naturais da cena nem sempre são suficientes para fornecer referências precisas para o rastreamento estável. O sistema também faz uso de um sensor inercial, um dispositivo acoplado à câmera, que incorpora acelerômetros, giroscópios em miniatura e magnetômetros.

Para a calibração inicial, é determinada a posição relativa e a orientação do sensor inercial em relação à câmera. A distorção da lente de distância focal, também, é calibrada, e no caso de uma lente com zoom, estes serão variáveis, de modo que o ajuste de zoom e foco são medidos com sensores mecânicos.

Durante o rastreamento, as imagens da câmera são capturadas e processadas ao vivo, a fim de localizar os recursos modelados anteriormente. Os dados do sensor inercial também são processados. No início do processo de controle, os dados da imagem e do sensor são utilizados pelo processo de registro inicial para calcular a posição inicial da câmera, o que pode levar vários segundos. Após a inicialização, o módulo de controle preditivo assume, acompanhando o movimento da câmera quadro a quadro. Se o algoritmo preditivo falhar, o processo de inicialização pode ser chamado novamente, para restabelecer o rastreamento. A figura 3.4 mostra os componentes principais da estrutura do sistema MATRIS.

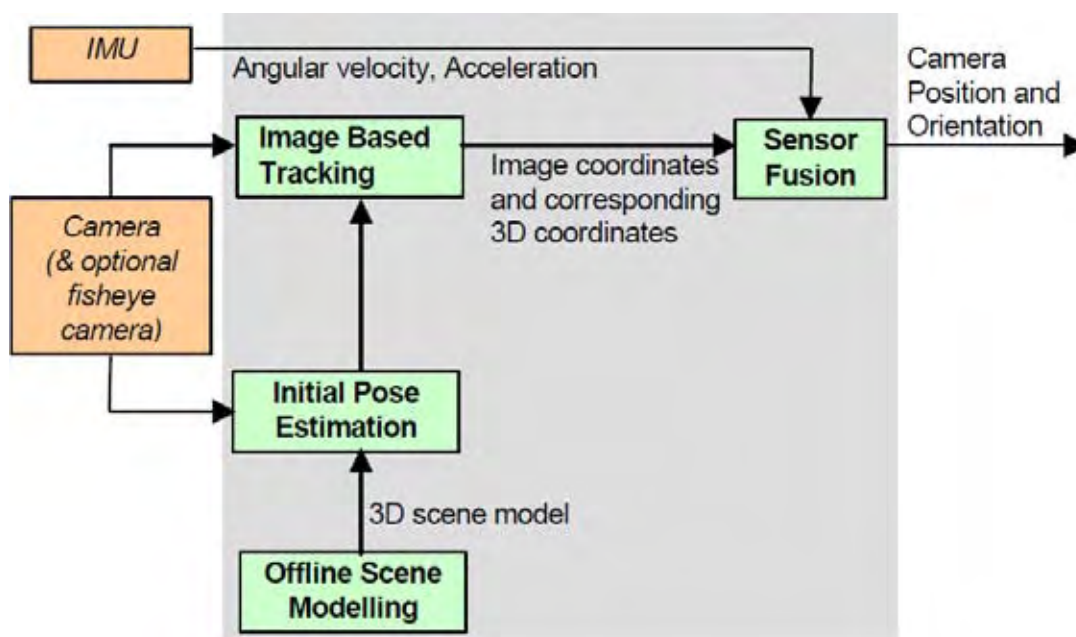


Figura 3.4 – Estrutura do sistema do projeto MATRIS.

Fonte: Chandaria et al. (2006)

Os componentes do sistema MATRIS consistem em uma câmera normal, um sensor inercial e um computador. Uma câmera com lente olho de peixe, um GPs e um sensor de lente são opcionais (Figura 3.5).



Figura 3.5 - Câmera de TV com sensor inercial, câmera olho de peixe e sensor da lente.

Fonte: Adaptado de Chandaria et al. (2006)

3.2.1 MÓDULO INERCIAL

O módulo inercial mede a velocidade angular e aceleração 3D linear, usando giroscópio de estado sólido e acelerômetros. O módulo, também, apresenta um sistema integrado de 3D de sensor magnético, que pode ser usado para detectar a posição. O DSP (*Digital Signal Processor*), processador de sinal digital, processa as informações do sensor a uma velocidade de até 512 Hz, compensando parâmetros do sensor, tais como ganho, nanoortogonalidade, dependência da temperatura, etc. Além disso o processador executa um algoritmo que permite que o sensor inercial estime a orientação 3D. O módulo inercial é sincronizado com a câmera de TV, usando o sinal de vídeo *genlock*² para gerar um sinal de disparo de 100Hz.

² *Genlock: Como um subconjunto de sincronização de quadro, genlock é comumente usado em vídeo pós-produção, edição não-linear (NLE) e estúdios de transmissão. Sua utilização garante que a saída gráfica de uma estação de trabalho está bloqueada (LOCK) para um sinal gerado externamente (GEN), garantindo que os dispositivos do estúdio (câmeras, gravadores de fitas de vídeo, personagem ou geradores de título, e assim por diante) trabalhem em conjunto de forma eficaz. (Nvidia manual Gforce 2012).*

3.2.2 CÂMERA OLHO DE PEIXE

Para garantir o rastreamento, foi acoplada, cobrindo um campo de 180°, uma câmera com lente olho de peixe, que pode ser utilizada como um dispositivo adicional de localização. A câmera olho de peixe está montada na parte superior dianteira da câmera principal e pode ver toda a parte dos arredores da cena e seu entorno, podendo ser utilizada para o rastreamento. A figura 3.6 compara a imagem da câmera principal (esquerda) com a câmera olho de peixe (à direita) da mesma cena, e mostra o campo muito maior de visão da câmera olho de peixe.



Figura 3.6 - Vista da câmera principal (à esquerda) e da câmera olho de peixe (à direita) da mesma cena.

Fonte: Chandaria et al. (2006)

3.2.3 MODELAGEM DE CENAS OFF-LINE

Na fase off-line, recursos 3D, que são adequados para o acompanhamento on-line, serão gerados automaticamente, tendo como ponto de partida a cena, o que permite a escolha das características para uma cena específica. As imagens da cena sem atores são capturadas, juntamente com os dados provenientes do sensor inercial e da lente, e insere o sistema de câmera que é estimado, usando uma estrutura robusta de abordagem de filmes.

De todas as diferentes imagens, um modelo de superfície 3D do conjunto é calculado automaticamente. Características da imagem 2D são rastreadas usando um modo de rastreamento rápido da imagem, e os pontos de função 3D são triangulados e automaticamente inseridos. Mapas de profundidade são calculados, com base em todas as visualizações de câmera, com análise estéreo *multi-view*. Os recursos 3D são selecionados de tal forma que facilitarão o acompanhamento online.

3.2.4 ESTIMATIVA INICIAL

Quando o sistema é iniciado pela utilização em tempo real, o processo de inicialização gera uma hipótese inicial da câmera. Ele compara a imagem atual da câmera com as imagens de referência, tiradas de posições conhecidas e orientações no banco de dados, e pode usar outras informações disponíveis, como a orientação a partir da posição do módulo do sensor inercial. A hipótese inicial para representar a câmera é então refinada por correspondências de características entre a imagem atual da câmera e a base de dados da imagem mais próxima. Essas características são extraídas da imagem real e comparadas com as características 3D, no banco de dados, associadas à imagem de referência. As resultantes "correspondências", características visíveis e suas correspondentes localizações 3D, são empregadas para refinar a câmera, utilizando o bem conhecido método *RANSAC*³, que é robusto contra valores atípicos.

3.2.5 RASTREAMENTO BASEADO EM IMAGENS

Esse módulo utiliza técnicas de visão computacional para localizar os recursos modelados em 3D. Ele usa uma previsão da câmera apresentada com base no processo de fusão do sensor para estimar as posições das características, e, em seguida, em torno de pesquisas desses locais estimados para gerar uma lista de correspondências.

Com a finalidade de combinar as características do trecho com as do modelo de cena, é necessário, primeiro, transformar cada textura do trecho armazenado, de modo que sua escala, orientação e posição adeque-se com o previsto pela câmera. Lembrando que se deve utilizar uma interpolarização de subpixel para aumentar a precisão do resultado final.

A Rejeição isolada do *RANSAC*, segundo os autores, é realizada no conjunto inicial de correspondências, e as correspondências válidas são, então, passadas para o processo de fusão de sensores, juntamente com as suas incertezas. As correspondências são usadas como medição do módulo de fusão de sensores para corrigir a estimativa atual.

³*RANSAC* : (Random Sample Consensus) É um método iterativo para estimar os parâmetros de um modelo matemático de um conjunto de dados, dado um modelo que requer um mínimo de n pontos para instanciar seus parâmetros livres, e um conjunto de pontos P , tal que o número de pontos em P é maior que n , seleciona randomicamente um subconjunto S_1 de n pontos de P e instancia o modelo (Jesse Russell, Ronald Cohn 2012).

Esclarecendo que o método foi desenvolvido para estimar a pose da câmera, combinando segmentos de linha com linhas conhecidas em um modelo 3D. Melhor dizendo, o sistema faz uso de uma técnica de minimização dos mínimos quadrados para encontrar a câmera.

A câmera é utilizada para selecionar as regiões da imagem, que são pesquisadas para localizar cada linha. O detector de linha pode ser configurado para ignorar pixels que não são cercados por uma determinada cor, o que ajuda a evitar falsas partidas, como a detecção de bordas de painéis publicitários, quando as linhas marcadas em um campo de futebol são as características desejadas.

3.2.6 SENSOR DE FUSÃO

O processo de fusão de sensores leva as correspondências a partir da fase de visão computacional e combina-os com a estimativa de representar o sensor inercial. Ao modelar o sistema como um todo, incluindo informação de confiança, a estimativa otimizada da câmera pode ser feita.

O projeto MATRIS adotou uma abordagem inversa, inspirado por algoritmos de alto desempenho de navegação da aeronave. Em tais aplicações, o conjunto inercial é o sensor de navegação principal para a estimativa de altitude e posição. Usando os dados do grupo de sensores inerciais, permite rápida mudança de movimentos para ser monitorado, mas sofre de desvio de longo prazo. Por essa razão, os sensores de apoio, tais como balizas terrestres e satélites do sistema de posicionamento global (GPS) são utilizados sempre que possível. Essas informações externas devem ser comparadas com o modelo de cena utilizado no sistema aqui descrito.

Os pontos importantes são: 1º) movimentos rápidos podem ser manuseados sem dificuldade; 2º) nem todos os ângulos de liberdade para a estimativa de pose tem que estar disponíveis em cada quadro de imagem. Em vez disso, os recursos correspondentes em 2D e 3D, com base na fase da visão computacional, são usados para corrigir o grupo de sensores inerciais.

Nesse projeto, os autores demonstraram que um sistema de rastreamento de câmera em tempo real não necessita de marcadores ou infraestrutura especial de estúdio. Constatou-se que, mediante a combinação de análise de imagens e sensores inerciais, ocorre

um melhor resultado do que pode ser conseguido com a análise de imagens sozinhas, podendo ser incorporado uma câmera olho de peixe para melhorar a robustez.

3.3 CONSIDERAÇÕES FINAIS

Neste capítulo, Thomas et al. (1997) demonstram dois métodos para calcular a posição da câmera em estúdios virtuais: um método utiliza marcadores em cena, e o outro utiliza marcadores fora de cena e uma câmera auxiliar voltada para eles.

Já Chandaria et al. (2006) utilizaram a combinação de análise de imagens da cena através de uma câmera olho de peixe, e o acoplamento de um sensor inercial à câmera principal.

A combinação desses dois artigos vem de encontro com a proposta deste projeto, a qual utiliza-se de uma webcam apontada para os marcadores fora de cena.

4 FERRAMENTAS UTILIZADAS

Neste capítulo, são apresentadas as principais ferramentas utilizadas no desenvolvimento deste trabalho. A biblioteca ARToolkit (Kato et al., 1999), a qual permite a detecção e o cálculo do posicionamento e orientação de marcadores fiduciais passivos. Para a utilização dos serviços dessa biblioteca, necessita-se apenas de uma webcam convencional para a captura de imagens do ambiente real. Por sua vez, a segunda ferramenta refere-se ao controle remoto Wiimote, acoplado com o Motion Plus, o qual possui, entre outros componentes, um sensor inercial (giroscópio), capaz de fornecer, de forma precisa, informações de orientação nos três eixos coordenados. Uma descrição mais detalhada dessa ferramenta encontra-se nas seções seguintes.

4.1 A BIBLIOTECA ARTOOLKIT

A biblioteca ARToolkit (ARToolkit, 2012) utiliza a linguagem C e C++ e oferece um conjunto de funções que realizam algumas das tarefas mais complexas das aplicações de realidade aumentada. Sua implementação é baseada em algoritmos de visão computacional para resolver o problema da estimativa do campo de visão do usuário (Kato et al., 1999).

Distribuída livremente para fins não comerciais sob a licença GNU e suportada por várias plataformas, a biblioteca ARToolkit possui uma excelente documentação, sendo utilizada em conjunto com alguns softwares gráficos.

A biblioteca ARToolkit tem como principal funcionalidade o rastreamento da posição e da orientação da câmera, uma vez que simples quadrados são utilizados como marcadores para localização dessas coordenadas. A configuração da câmera é fácil e tem uma execução rápida o suficiente para aplicações de tempo real, conforme mostra a (Figura 4.1).

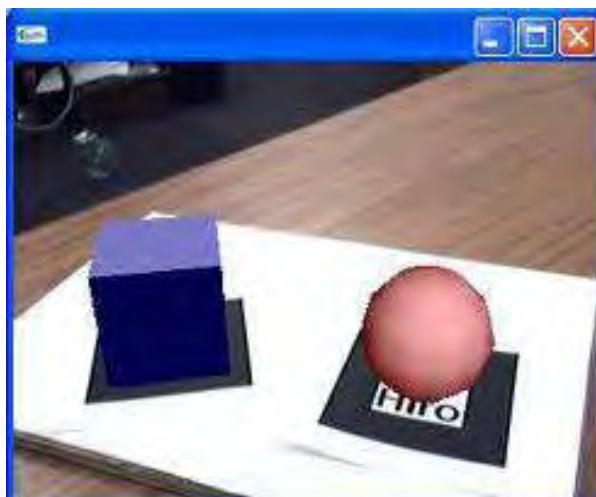


Figura 4.1 - ARToolkit em execução.

Fonte: KATO et al. (1999)

4.1.1 ASPECTOS GERAIS DO FUNCIONAMENTO

O processo de reconhecimento executado pela biblioteca Artoolkit inicia-se com a aquisição de imagens por meio de uma câmera (normalmente uma webcam), a qual é convertida em imagem binária, obedecendo a um valor limite (thresholding) estabelecido, para transformar as cores da imagem em preta ou branca. Essa primeira etapa do processo é ilustrada pelas (Figura 4.2)



Figura 4.2 – Aquisição da Imagem e binarização do marcador

Fonte: KATO et al. (1999)

4.1.2 PROCESSO DE RECONHECIMENTO

O limiar de binarização (thresholding), segundo Kato et al. (1999), pode ser alterado, manualmente, em tempo de execução, provocando uma interrupção no programa. A variação desse valor, muitas vezes é necessária, principalmente em ambientes em que a iluminação não é constante. Um *thresholding* adaptativo para a biblioteca ARToolkit foi desenvolvido por Pintaric (2003), na tentativa de automatizar esse ajuste.

Os autores consideram, também, que, na imagem binária, são identificadas regiões quadradas, que passam por um processo de comparação com modelos de marcadores, previamente, inseridos no sistema. Havendo sucesso na comparação, identifica-se um marcador.

O fato de as dimensões do marcador estarem armazenadas no sistema, permite à biblioteca ARToolkit calcular a posição da câmera de vídeo em relação a esse marcador físico. Esses valores são utilizados para encontrar as coordenadas para projeção dos objetos virtuais na tela, que deve ser feita precisamente sobre o marcador real por meio de algum software gráfico (Kato et al.1999).

Os princípios matemáticos, envolvidos nas estimativas de posicionamento e orientação dos marcadores, baseiam-se em cálculos que geram valores para matrizes de transformação. O processo de mudança entre sistemas de coordenadas, realizado pela biblioteca ARTollKit, pode ser observado na (Figura 4.3).

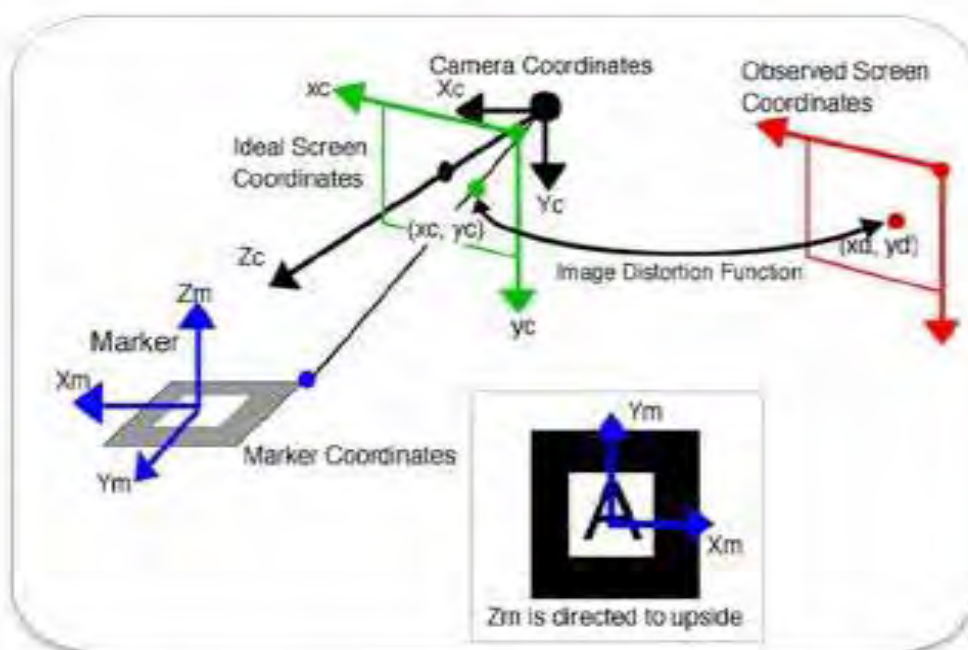


Figura 4.3 – Sistema de coordenadas do ARToolkit.

Fonte: Kato et al. (1999)

4.1.3 PRECISÃO DO PROCESSO DE DETECÇÃO

Conforme Piekarski et al. (2002), para que se possa analisar com acurácia as estimativas de posição e orientação de marcadores, foi definida uma função que calcula as áreas em que os erros são menores. Realizou-se um experimento no qual uma câmera foi fixada de modo que a altura da lente e o centro do marcador ficassem no mesmo nível. O ângulo em torno dos eixos X, Z e a angulação permaneceram fixas em 0°, e as variações foram medidas em relação ao eixo Y.

Os resultados da experiência mostram que os erros da biblioteca ARTToolkit, em relação à distância do marcador na faixa entre 20 e 70 cm, são baixos, e o desvio padrão entre, 20 e 50 cm, é pequeno. Em relação aos ângulos de rotação, os erros são menores na faixa entre 30° e 40°, e o desvio padrão é pequeno na faixa entre 40° e 85° (Figura 4.4).

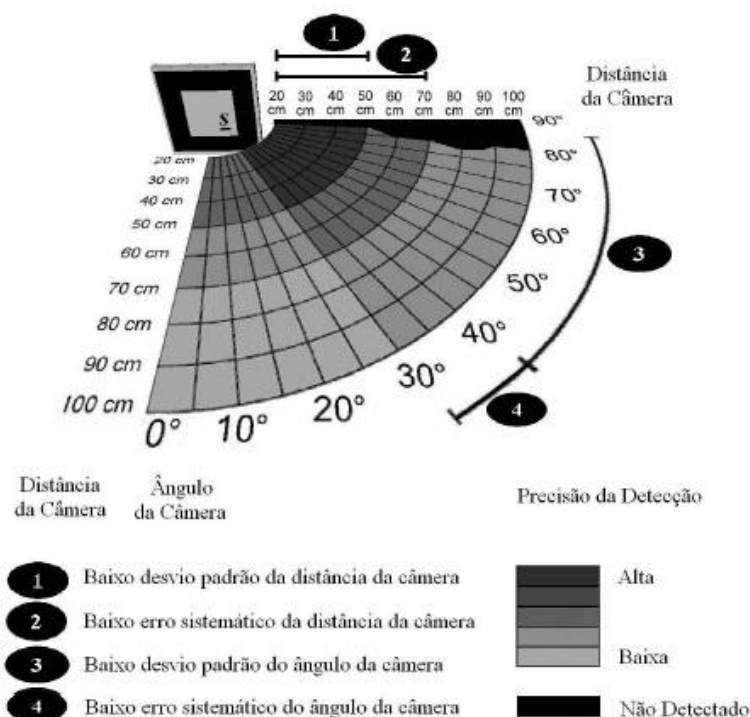


Figura 4.4 - Análise da precisão da biblioteca ARTToolkit.

Fonte: Piekarski et al. (2002)

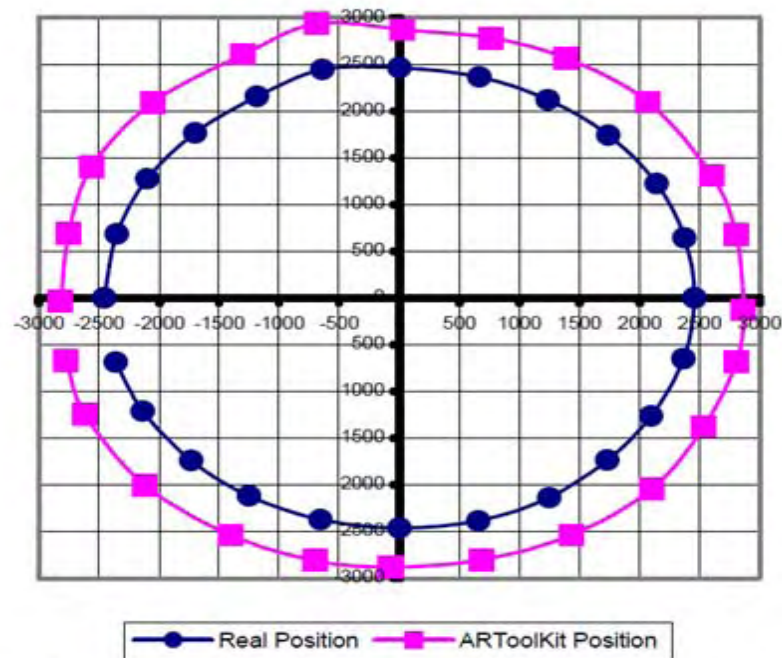
De acordo com os experimentos realizados por Malbezin (2002), o erro aumenta com a distância dos valores entre 1m a 3m, entre 9% a 18% de Y e 6% a 12% de X (Figura 4.5). Como um primeiro passo para a correção, pode ser desenvolvido um filtro para medir o erro de 1m a 2,5.

Distance (m)	1	1.5	2	2.5
Error (mm)	± 14	± 18	± 22	± 27

Maximum error values for the four distances



X and Y mean error values for the four radius values



ARToolKit radius estimates are always larger than actual physical measurements (units are millimetres)

Figura 4.5 - Comparação das medidas do mundo real x capturadas pela câmera.

Fonte: Malbezin et al. (2002)

4.1.4 CONFIABILIDADE X DESEMPENHO

A resolução do padrão inserido no marcador influencia diretamente o reconhecimento do mesmo.

No processo de identificação, quanto maior for a resolução, maior será o número de *pixels* analisados com mais precisão (Figura 4.6).



Figura 4.6 - Exemplos de diferentes resoluções de imagens.

Fonte: Kato, (2007)

Kato (2007) destaca que, com essa precisão, haverá um prejuízo do desempenho do sistema. Também salienta que o problema fica mais evidente quando existe a necessidade de inserção de muitos marcadores na cena. Nesse caso, uma resolução muito alta, certamente, provocará atrasos significativos.

A principal função para detecção de marcadores da biblioteca ARToolkit é armazenar valores em um histórico para maior confiabilidade em suas detecções.

4.2 WII REMOTE

Um dispositivo muito conhecido hoje no mercado é o videogame Nintendo Wii, cujos controles remotos são utilizados com a movimentação no espaço. Os movimentos são “capturados” através de componentes eletrônicos, dispostos dentro do controle remoto (Figura 4.7), fazendo as deslocamentos dentro do jogo, utilizando amplamente os sensores de movimentação, como se descreve a seguir.



Figura 4.7 - Controle remoto do Nintendo Wii.

Fonte: Manual do VídeoGame Nintendo Wii, (2006).

O *Wii remote*, informalmente conhecido como *Wiimote*, é um dispositivo de entrada principal, sem fio, que, para se comunicar com o videogame, usa tecnologias *Bluetooth*.

O wii remote possui vários componentes internos, como comunicação *bluetooth*, comandos por botões, acelerômetro, câmera de infravermelho, botão de liga e desliga, alto-falante, *led's* de sinalização e status de informações. Seus componentes principais são descritos nas seções que seguem.

4.2.1 COMUNICAÇÃO BLUETOOTH

Como mencionado anteriormente, o Wiimote comunica-se com o videogame ou com o computador, ambos denominados como host, através de um protocolo Bluetooth padrão. O wiimote pode ser colocado em modo de descoberta por 20s, pressionando o botão de sincronização em suas costas, (junto com o compartimento das pilhas), ou pressionando simultaneamente, os botões 1 e 2 continuamente, forçando o modo de descoberta sem desligar. Esse procedimento é facilmente visualizado, visto que os *led's* indicadores, na frente do wiimote, começam a piscar.

No videogame, cada controle é reconhecido e numerado através de seus *led's*, sendo que o *led* 1 aceso indica o jogador 1, o *led* 2 aceso indica o jogador 2, assim sucessivamente.

A sincronização do wiimote com o computador (ou o videogame) dá-se por meio do envio do código PIN (endereço físico do *bluetooth* em hexadecimal). Exemplo do número PIN - 00:1D:14:9F:5C:6D.

Ao ligar o wiimote pressionando as teclas 1 e 2, o wiimote envia o código PIN para o host, fazendo, assim, a sincronização dos dispositivos.

4.2.2 COMANDOS POR BOTÕES

O wiimote possui um conjunto de 12 teclas, com funções diferentes, a saber:

Tecla de liga e desliga;

Teclas de movimentação (setas esquerda e direita, para cima e para baixo);

Tecla botão A;

Teclas + e -;

Tecla “home”;

Teclas 1 e 2.

Essas teclas, junto com o console, têm suas funções designadas para o jogo; entretanto, para a utilização em um computador, elas podem ser programadas de acordo com a necessidade do usuário.

4.2.3 ACELERÔMETRO

O controle remoto do wii possui um acelerômetro linear de três eixos, localizado na placa de circuito impresso, ao lado da Tecla A. A figura 4.8 ilustra o posicionamento do acelerômetro na placa de circuito impresso.

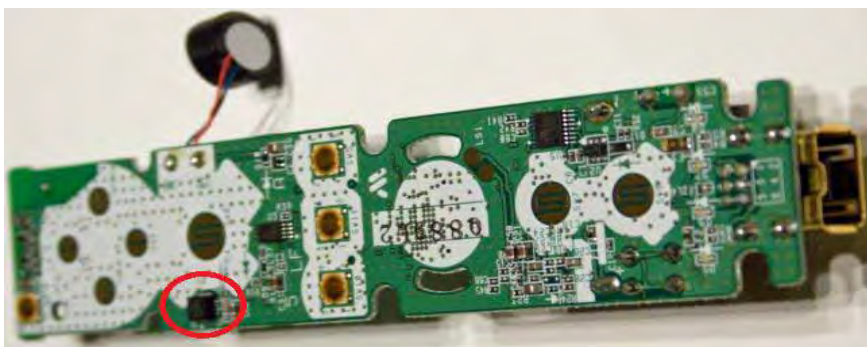


Figura 4.8 - Circuito integrado do acelerômetro na placa do Wii.

Fonte: Manual do Videogame Nintendo Wii, (2006).

Esse dispositivo está dimensionado para medir as acelerações acima de uma gama de pelo menos + ou - 3g, com sensibilidade de 10%.

O acelerômetro mede a força exercida por um conjunto de pequenas massas em relação ao seu invólucro, além da aceleração linear em um quadro de referência em queda livre. Se o *wii mote* está em queda livre, ele irá informar aceleração zero. Em repouso, irá reportar uma aceleração para cima igual à aceleração da gravidade g (cerca de $9,8 \text{ m/s}^2$), mas no sentido oposto.

Em todos os modos de relatórios, que incluem os dados do acelerômetro, os dados são relatados com três bytes consecutivos. Exemplo: (a1) RR BB BB XX YY ZZ [...].

XX YY e ZZ são bytes sem sinal representando a aceleração de cada um dos três eixos, em que zero é cerca de $0x80$ da aceleração; os itens RR, BB, BB são os valores de aceleração.

Na Figura 4.9, são apresentados os eixos e os sentidos de suas acelerações.

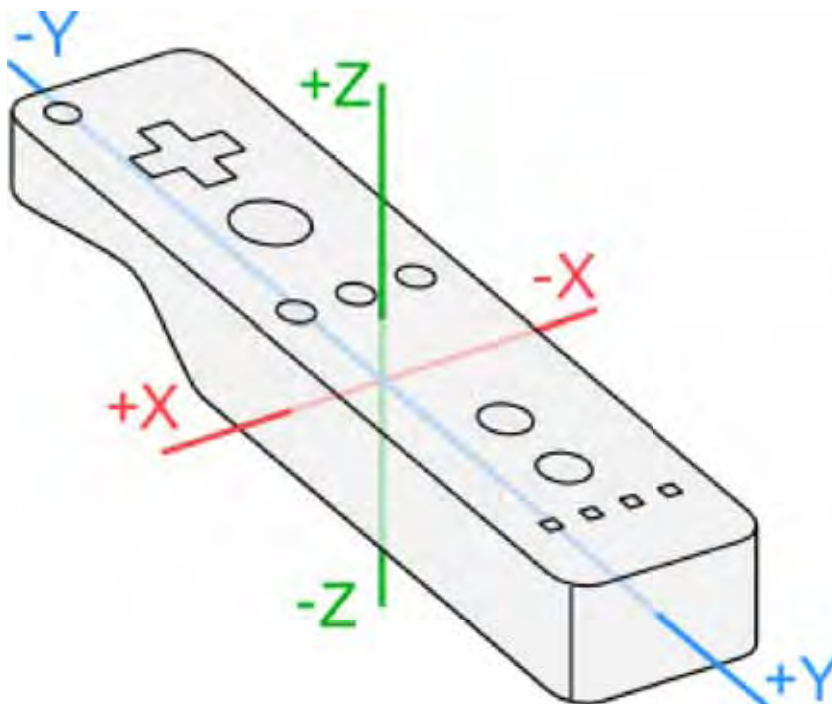


Figura 4.9 - Eixos cartesianos e acelerações.

Fonte: Manual do Videogame Nintendo Wii, (2006).

4.2.4 MOTION PLUS

O acessório Motion Plus é conectado diretamente ao controle Wii Remote (Figura 4.10). Ele adiciona três giroscópios ao wiimote, dispostos ortogonalmente entre si.

O Motion Plus utiliza o giroscópio IDG 600, que possui dois giroscópios integrados, dispostos em dois eixos; para o sensoriamento do terceiro eixo, é utilizado o giroscópio X3500W, produzido pela EPSON TOYOCOM.



Figura 4.10 - Wii Motion Plus

Fonte: Manual do Videogame Nintendo Wii, (2008).

4.2.5 CÂMERA DE INFRAVERMELHO

O Wii Remote inclui uma câmera monocromática 128x96 com processamento de imagem (Figura 4.11). A câmera possui um filtro de passagem de infravermelho o qual barra as luzes ambiente e deixa passar apenas a luz infravermelha do *Bar Sensor*. O processamento de imagem da câmera é capaz de rastrear até quatro objetos em movimento, e esses dados são os únicos dados disponíveis para o equipamento. A imagem processada utiliza uma análise de 8x subpixel para fornecer resolução de 1024x768 para os pontos monitorados. O *Bar Sensor*, que vem com o Wii, inclui dois *clusters* de LED IR em cada extremidade, que são controlados pelo Wii. A distância entre os centros dos agrupamentos dos LED's são de 20 cm.

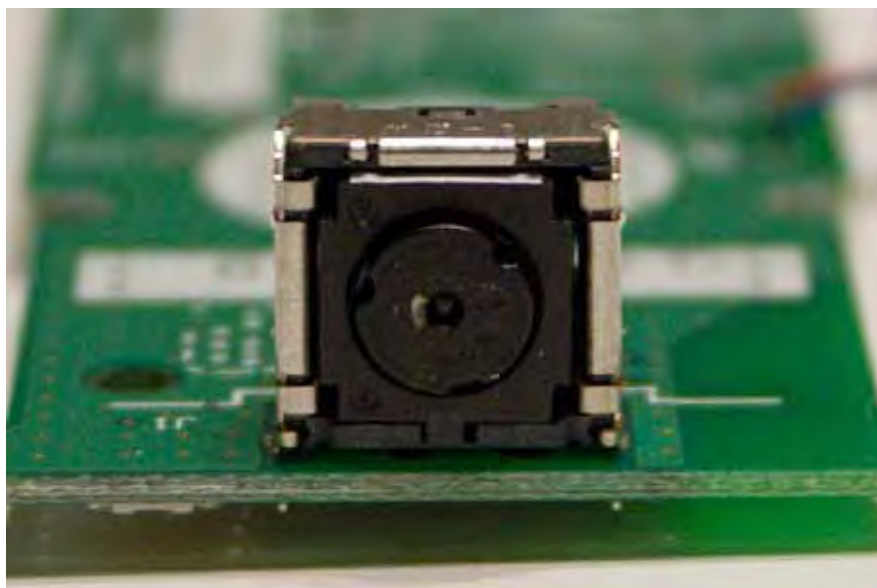


Figura 4.11 - Câmera IR Wiimote.

Fonte: Manual do Videogame Nintendo Wii, (2006).

4.3 CONSIDERAÇÕES FINAIS

Levando-se em conta o do estudo das características da biblioteca de realidade aumentada ARToolkit, ficou evidente que a ferramenta possui a funcionalidade necessária para ser utilizada na elaboração de um sistema de rastreamento tridimensional.

O controle remoto do Nintendo Wii possui dois acelerômetros como *default*, porem acoplando-se o acessório Motion Plus, o qual possui 3 giroscópios têm-se os sensores inerciais (giroscópios) que serão utilizados para a orientação da câmera.

Utilizando-se a câmera com a biblioteca ARToolkit em conjunto com o Wii Plus, pode-se obter a posição e a orientação com maior acurácia.

No capítulo a seguir, serão explanados os passos do desenvolvimento do sistema de rastreamento de câmera.

5 SISTEMA DESENVOLVIDO

No presente capítulo, apresenta-se a estruturação do sistema de rastreamento de câmeras idealizado, bem como procura-se detalhar a sua implementação.

5.1 ESTRUTURAÇÃO DO SISTEMA DE RASTREAMENTO DE CÂMERAS

Este trabalho, de acordo com o mencionado no Capítulo 1, tem como objetivo principal o desenvolvimento de um sistema de rastreamento de câmeras de filmagem adequado à utilização em Estúdios Virtuais, valendo-se do estudo das tecnologias de rastreamento 3D existentes. Sua utilização permitirá obter um arquivo que relacione as coordenadas da câmera com suas coordenadas relativas ao mundo real.

O sistema foi desenvolvido para realizar o rastreamento de câmeras em um estúdio real. Para isso, utilizou-se o método de rastreamento óptico, baseado em marcadores fiduciais, juntamente com os serviços da biblioteca de Realidade Aumentada, conhecida como ARToolkit (KATO et al. (2006)). Esclarecendo, ainda, que o sistema utiliza multimarcadores previamente cadastrados, posicionados no teto do estúdio de filmagem, capturados por uma câmera auxiliar (webcam). Esses marcadores, uma vez reconhecidos e identificados pelo sistema, viabilizam o cálculo de suas posições e orientações com relação à câmera. Através do cálculo da matriz de transformação, será possível realizar a conversão dos sistemas de coordenadas entre a câmera e o marcador.

A câmera auxiliar obtém as coordenadas relativas ao multimarcador do teto. Juntamente com a matriz de transformação obtida e conhecendo uma relação fixa entre o marcador de cena e o multimarcador, pode-se estimar a posição da câmera no sistema de coordenadas do mundo real.

A sequência de figuras 5.1 (a) e (b) exibe o ambiente virtual experimental com os marcadores no teto, e a webcam acoplada à câmera principal.

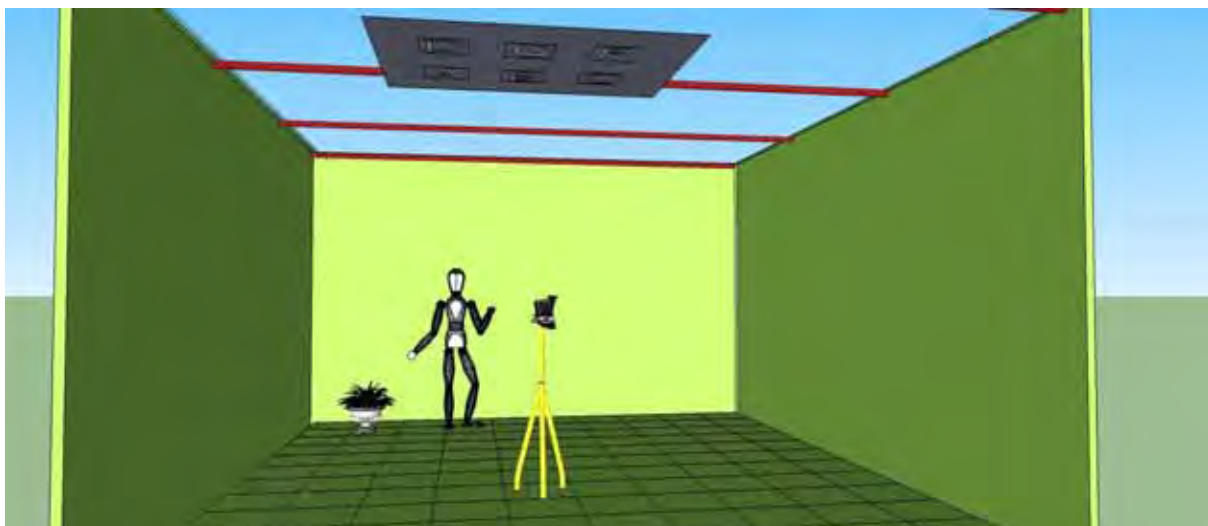


Figura 5.1 - Ambiente virtual com webcam, wiimote e multimarcador.

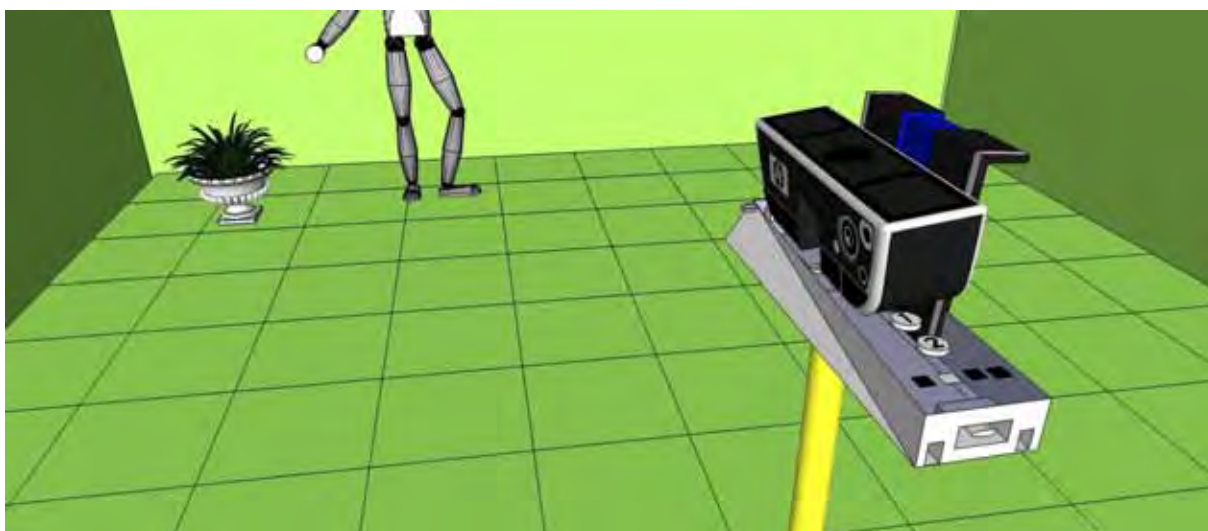


Figura 5.2 – Vista aproximada da webcam e o wiimote.

Para que o sistema funcione corretamente, alguns passos devem ser seguidos. O diagrama demonstra item a item os procedimentos a serem feitos (Figura 5.3).

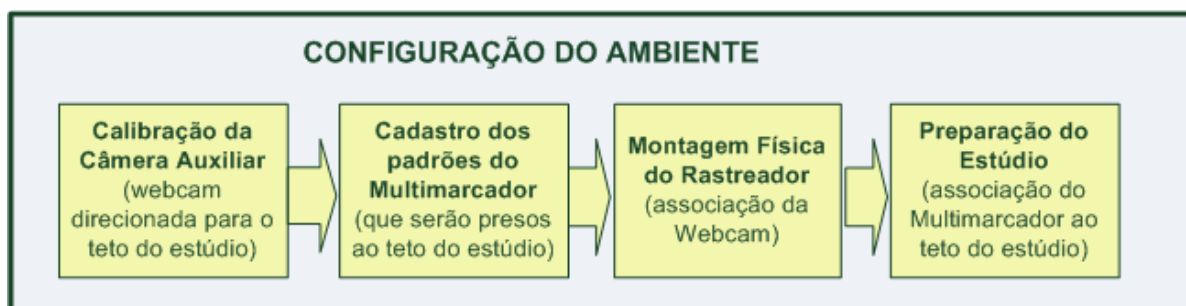


Figura 5.3 - Diagrama de configuração do ambiente.

a) Calibração

O primeiro passo na etapa da configuração é a calibração da câmera auxiliar (Webcam). O processo de calibração busca a correspondência entre os pontos tridimensionais com suas respectivas projeções bidimensionais no plano da imagem; desta forma, podemos identificar 2 sistemas de coordenadas, de modo que a correspondência entre esses pontos possa ser expressa como uma composição de transformações, que associe os pontos 3D em 2D correspondentes.

Em Kato e Billinghamurst (1999), os autores usaram um simples marcador em forma de grade para realizar a calibração da câmera. As coordenadas de todos os pontos de cruzamento de uma malha são conhecidas nas coordenadas 3D da grade de calibração. Os pontos de intersecção das coordenadas da tela podem ser obtidos por processamento de imagem, após a obtenção da imagem do cartão de calibração. Muitos pares de coordenadas 3D locais do cartão de calibração (X_t, Y_t, Z_t) e as coordenadas da câmera (X_c, Y_c) são usados para encontrar matriz de transformação.

As relações entre as coordenadas da tela da câmera (X_c, Y_c) as coordenadas de câmera (X_c, Y_c, Z_c) e as coordenadas do cartão de calibração (X_t, Y_t, Z_t) podem ser representadas como na equação Eq. 5.1.

$$\begin{bmatrix} hx_c \\ hy_c \\ h \\ 1 \end{bmatrix} = P \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = P \cdot T_{ct} \begin{bmatrix} X_t \\ Y_t \\ Z_t \\ 1 \end{bmatrix} = C \begin{bmatrix} X_t \\ Y_t \\ Z_t \\ 1 \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} \\ C_{21} & C_{22} & C_{23} & C_{24} \\ C_{31} & C_{32} & C_{33} & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_t \\ Y_t \\ Z_t \\ 1 \end{bmatrix}$$

$$P = \begin{bmatrix} S_{xf} & 0 & x_0 & 0 \\ 0 & S_{yf} & Y_0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, T_{ct} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_x \\ R_{21} & R_{22} & R_{23} & T_y \\ R_{31} & R_{32} & R_{33} & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{Eq. 5.1})$$

Onde P é a matriz de transformação de perspectiva, que deve ser encontrada, f é a distância focal, S_x é o fator de escala na direção do eixo X, S_y é o fator de escala na direção do eixo y, (X_0, Y_0) é a posição em que o eixo Z vê as coordenadas de passagens do frame, T_{ct} representa a translação e transformação de rotação a partir das coordenadas dos cartões para as coordenadas da câmera, e C é a matriz de transformação obtida, combinando P e T_{ct} .

Uma vez que muitos pares de (X_c, Y_c) e (X_t, Y_t, Z_t) foram obtidos pelo procedimento mencionado acima, a matriz C pode ser estimada. No entanto, a matriz de C

não pode ser decomposta em P e T_{ct} em geral, devido a matriz C ter 11 variáveis independentes das matrizes P e T_{ct} , tendo 4 e 6, respectivamente; então, a soma das variáveis independentes de P e T_{ct} não é igual a um dos C . Uma variável escalar k é adicionada em P para tornar esses números iguais, segundo equação Eq. 5.2.

$$\begin{bmatrix} hx_c \\ hy_c \\ h \\ 1 \end{bmatrix} = \begin{bmatrix} S_{xf} & k & x_0 & 0 \\ 0 & S_{yf} & Y_0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_x \\ R_{21} & R_{22} & R_{23} & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_t \\ Y_t \\ Z_t \\ 1 \end{bmatrix} \quad (\text{Eq. 5.2})$$

Como resultado, a matriz C pode ser decomposta em P e T_{ct} . A variável k significa a inclinação entre o eixo x e eixo y , e idealmente deve ser zero, mas pode ser um pequeno valor de ruído.

b) Cadastro dos Padrões do Multimarcador

No passo seguinte, realiza-se o cadastramento dos marcadores que serão presos no teto. Esses marcadores devem ter as dimensões 8 x 8 e serem fixados distantes 20 cm na vertical e 20 cm na horizontal um dos outros, em relação ao centro de cada marcador. Os marcadores reconhecidos pelo ARToolKit consistem em figuras geométricas quadradas, que contém, no seu interior, símbolos para identificá-los. Os marcadores deverão ser cadastrados previamente, como explicado no apêndice 2 do cadastro de marcadores.

c) Montagem Física do Rastreador

O hardware, montado neste projeto, consiste em uma webcam e um tripé, para fazer o rastreamento da câmera. Foi acoplado, também, um laser de ponto com a finalidade de se observar para onde a câmera principal estava apontando (Figuras 5.4 (a) e (b)).



(a)

(b)

Figura 5.4 - Protótipo da webcam, Wiimote e laser de ponto.

d) Preparação do Estúdio

O último passo dessa etapa de configuração do ambiente consiste na preparação do estúdio, em que será feito o posicionamento do multimarcador no teto (Figura 5.5). Esse procedimento deve ser realizado marcando o centro dos marcadores em um ponto conhecido nos eixos X, Y e Z. Esse ponto deve ser anotado em milímetros, visto que o sistema vai solicitar ao usuário esses dados no início do rastreamento.



Figura 5.5 - Multimarcador fixado no teto.

5.2 ESTRUTURA DO SISTEMA

Os componentes principais do sistema de rastreamento de câmeras desenvolvido são mostrados na (Figura 5.6).

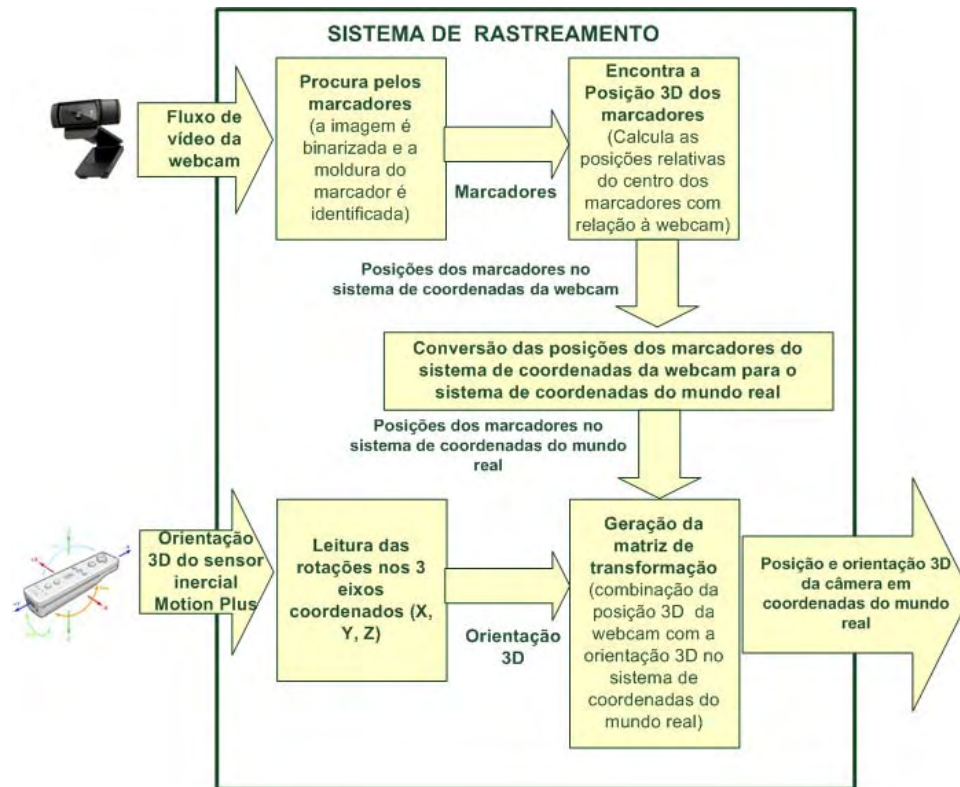


Figura 5. 6 - Diagrama de descrição do método.

O sistema, conforme ilustrado na Figura 5.6, possui um fluxo de imagens de vídeo proveniente da Webcam.

Considerando o fluxo de imagens da Webcam, o qual inicia-se com a procura dos marcadores, Kato e Billinghurst (1999) descrevem o reconhecimento das coordenadas do marcador como sendo possíveis devido ao fato de terem uma borda quadrada, de dimensões fixas e conhecidas, como visto na Figura 5.7. Com os valores das coordenadas do marcador já determinados, é possível descrever uma matriz de transformação das coordenadas do marcador para as coordenadas da câmera, demonstrada na equação Eq. 5.3.

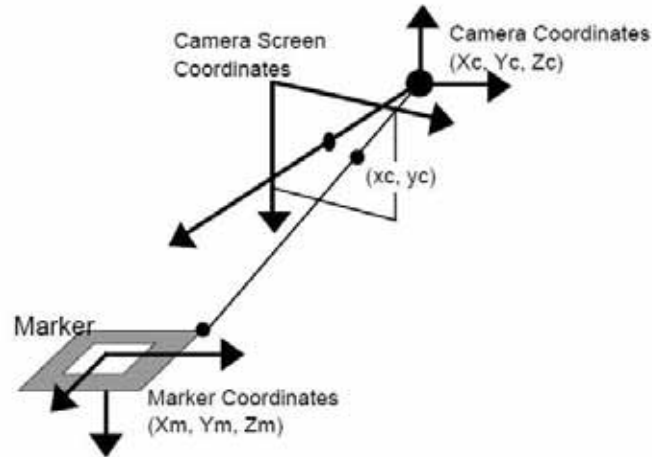


Figura 5.7 – Relação entre as coordenadas do marcador e da câmera, estimados pela análise da imagem.

Fonte: Kato e Billinghurst (1999).

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} V_{11} & V_{12} & V_{13} & W_X \\ V_{21} & V_{22} & V_{23} & W_Y \\ V_{31} & V_{32} & V_{33} & W_Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} = \begin{bmatrix} V_{3 \times 3} & W_{3 \times 1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} = T_{cm} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} \quad (\text{Eq. 5.3})$$

Logo após carregar uma imagem no *buffer*, busca-se, nesta imagem, pelos quatro segmentos de retas, que formam o quadrado do marcador. As coordenadas dos vértices das intersecções são armazenadas, pois elas serão utilizadas em seguida para determinar o centro do marcador.

Detectada, então, a moldura do marcador na imagem, realiza-se uma normalização na sub-imagem, contida dentro da moldura, que é então comparada com os modelos existentes, previamente cadastrados no sistema.

Para realizar essa normalização, utiliza-se a equação Eq.5.4. Todas as variáveis existentes na matriz de transformação são determinadas, substituindo-se as coordenadas da tela pelas coordenadas dos quatro vértices do marcador detectado (X_c, Y_c por X_m, Y_m , respectivamente). Após essa etapa, conclui-se o processo de normalização, utilizando-se a matriz de transformação.

$$\begin{bmatrix} hx_c \\ hy_c \\ h \end{bmatrix} = \begin{bmatrix} N_{11} & N_{12} & N_{13} \\ N_{21} & N_{22} & N_{23} \\ N_{31} & N_{32} & 1 \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \\ 1 \end{bmatrix} \quad (\text{Eq. 5.4})$$

Quando dois lados paralelos de um marcador quadrado são projetados sobre a imagem, as equações desses segmentos de reta nas coordenadas de tela da câmera são, equação Eq. 5.5.

$$a_1x + b_1y + c_1 = 0, \quad a_2x + b_2y + c_2 = 0 \quad (\text{Eq. 5.5})$$

Para cada marcador detectado, o valor desses parâmetros foi obtido no processo de linha de montagem da equação Eq.5.5. Com a perspectiva de projeção da matriz P, que é obtida pelo processo de calibração da câmera (visto na equação Eq. 5.6), as equações dos planos, que incluem essas duas coordenadas, respectivamente, podem ser representadas, como a equação Eq. 5.7, obtida com a troca das coordenadas da câmera, substituindo X_c e Y_c , em equação Eq. 5.6, para x e y em equação Eq. 5.5.

$$P = \begin{bmatrix} P_{11} & P_{12} & P_{13} & 0 \\ 0 & P_{22} & P_{23} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} hx_c \\ hy_c \\ h \\ 1 \end{bmatrix} = P \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \quad (\text{Eq. 5.6})$$

$$\begin{aligned} a_1P_{11}X_c + (a_1P_{12} + b_1P_{22})Y_c + (a_1P_{13} + b_1P_{23} + c_1)Z_c &= 0 \\ a_2P_{11}X_c + (a_2P_{12} + b_2P_{22})Y_c + (a_2P_{13} + b_2P_{23} + c_2)Z_c &= 0 \end{aligned} \quad (\text{Eq. 5.7})$$

A mesma técnica empregada para detectar o rastreamento de um único marcador descrito pode ser utilizada para rastrear um sistema composto por seis marcadores, por exemplo, para aumentar a área de referência de coordenadas disponíveis. Esses seis marcadores estarão dispostos, como na Figura 5.8, em forma de um tabuleiro, alinhados por suas bordas.

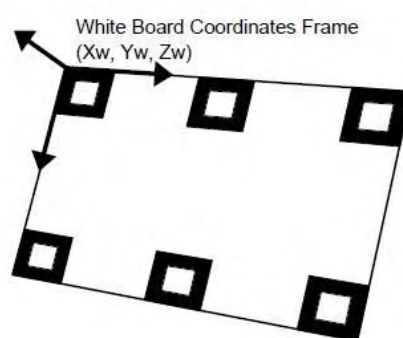


Figura 5.8 – Layout dos marcadores múltiplos.

Fonte: Kato e Billinghurst (1999).

Utiliza-se todos os seis marcadores para determinar a orientação e as coordenadas como um único sistema de multimarcadores. Durante o reconhecimento dos marcadores, algum ou alguns marcadores podem ser obstruídos pelo usuário. Enquanto os marcadores estiverem visíveis, o sistema continuará funcionando e fornecendo as informações das coordenadas do sistema. A matriz de transformação dos multimarcadores, pode ainda ser estimada a partir dos marcadores visíveis. Alguns pequenos erros podem acontecer se os marcadores visíveis diminuïrem. Para que esses erros sejam reduzidos, a equação de transformação (Eq. 5.7) deve considerar as coordenadas do sistema de multimarcadores e dos marcadores individualmente detectados na imagem.

Cada marcador possui um padrão único no interior de sua moldura, que possibilita ao sistema identificar os marcadores e sua posição na horizontal e vertical, dentro do sistema, e estimar sua rotação. Embora o sistema funcione como um único marcador, com uma coordenada única de saída, cada marcador possui as suas coordenadas individuais independentes, e as equações de linha do multimarcador são praticamente idênticas.

Portanto, extraindo-se todos os lados alinhados com base nos marcadores visíveis, cada equação da linha de marcadores é calculada utilizando todas as informações extraídas do sistema de multimarcadores. Além disto, utilizando todas as equações das linhas paralelas de marcadores do sistema, um vetor de direção pode ser estimado, e a orientação do sistema de multimarcadores é determinada.

5.3 IMPLEMENTAÇÃO DO PROTÓTIPO DO SISTEMA DE RASTREAMENTO

Nesta seção, é apresentada a implementação do sistema, cuja estruturação foi descrita nas seções anteriores.

Os componentes de hardware e software, utilizados neste trabalho, foram os seguintes:

A) COMPONENTES DE SOFTWARE

Para o desenvolvimento do software foram utilizados:

- ARToolkit 2.72.1
- O Microsoft Visual C++ 2010
- Windows 7 professional 64 bits
- *Netbeans* 7.2.1 (criação da interface).

B) COMPONENTES DO HARDWARE

- 01 - Note book com Processador core I7 2.8GHz, memória de 6GB, disco rígido 500GB e placa aceleradora de vídeo Nvidia com 1GB de memória
- Sistema Operacional Windows 7 64 bits
- 02 – Webcam Logitech HD C270 com Resolução de 3 MP, correção de iluminação, com foco automático
- 01 – Controle Remoto do Nintendo Wii
- 01 – Wii motion Plus
- 01 – Interface Bluetooth com chipset Broadcom
- 01 - Hub USB 04 portas para interligação das câmeras
- Marcadores Fiduciais (HIRO E MULTIMARCADOR)
- 01 Tripé com altura máxima de 1530 mm, inclinação de 90°
- Trena mecânica 5m
- Trena ultrassom
- Trena laser Bosh Glr 225 70m
- Laser de linha até 10m

- Laser de ponto até 10 m
- 02 luminárias fluorescentes com 2 lâmpadas de 40W cada, totalizando 160W de luz branca.

5.3.1 IMPLEMENTAÇÃO DO TRATAMENTO DO FLUXO DE VÍDEO

A Figura 5.9 ilustra o fluxograma geral do protótipo.

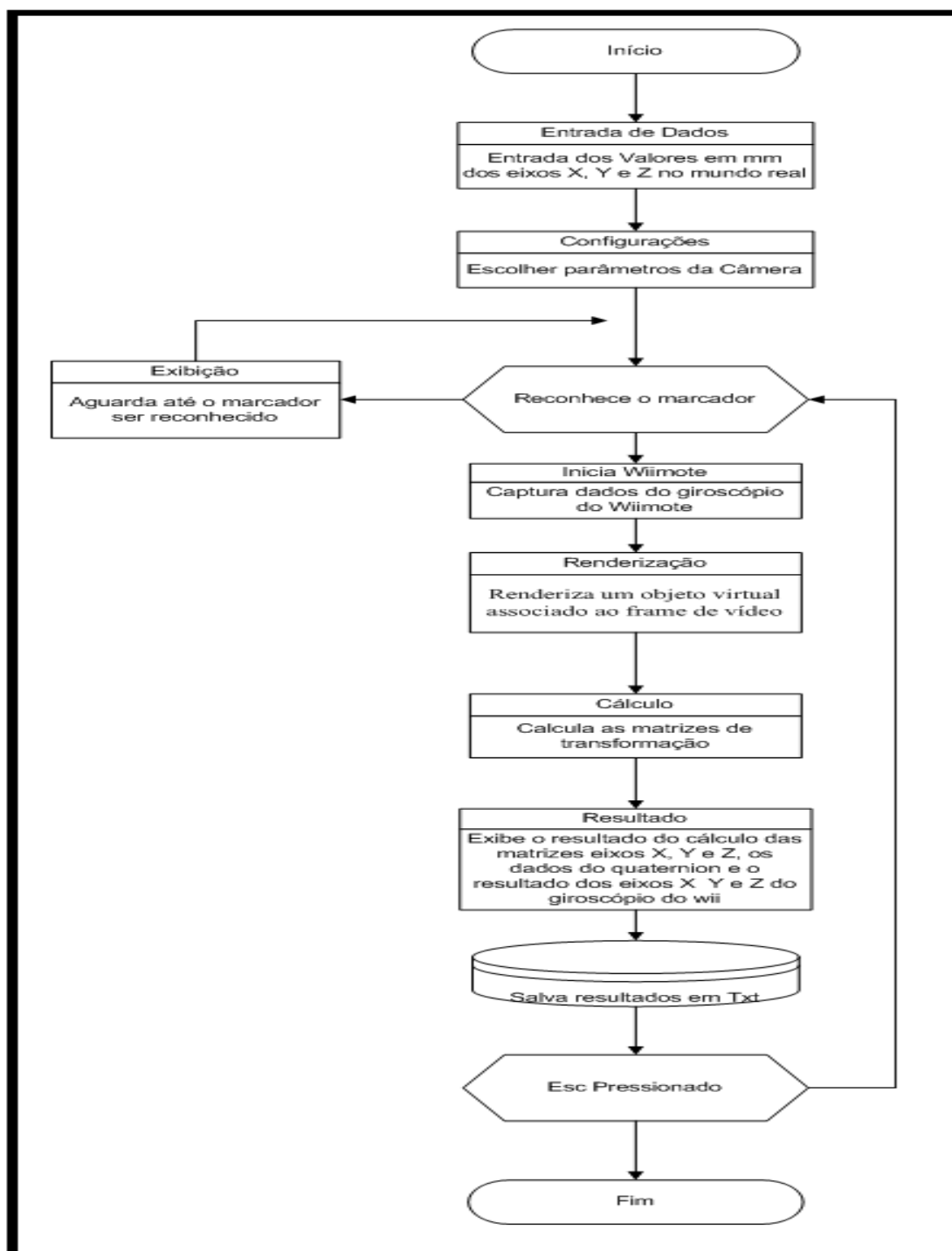


Figura 5.9 - Fluxograma do sistema desenvolvido.

Inicialmente, o sistema solicita ao usuário os parâmetros de configurações iniciais para o funcionamento do sistema. O usuário deve informar os valores das coordenadas X, Y e Z do mundo real, definidas previamente. Esses valores serão utilizados para se obter a matriz de transformação das coordenadas do sistema da câmera para o sistema de coordenadas do mundo real (Figura 5.10).

```
int at=255;
printf("\t\tRASTREADOR DE CAMERA\n\n\n");
printf("Digite o valor do x do marcador no mundo real: ");
scanf("%lf",&pontos[0]);
//target_center[0]=pontos[0];
printf("Digite o valor do y do marcador no mundo real: ");
scanf("%lf",&pontos[1]);
//target_center[1]=pontos[1];
printf("Digite o valor do z do marcador no mundo real: ");
scanf("%lf",&pontos[2]);
pontos[3]=1;

at=arParamLoad("Data/camera_para.dat",1,&mparam);
mparam.dist_factor[0]=mparam.xsize/2;
mparam.dist_factor[1]=mparam.ysize/2;
at=255;
at=arParamSave("Data/camera_para2.dat",1,&mparam);
//fptr=fopen("Data/camera_para.dat","rb"); //rb significa abrir
arquivo binário para leitura
//fread(&Binario, sizeof(Camera_dat_Binario), 1, fptr);
printf("\nX center of cam: %d\n",mparam.dist_factor[0]);
printf("\nY center of cam: %d\n",mparam.dist_factor[1]);

glutInit(&argc, argv);
init();

arVideoCapStart();
argMainLoop( NULL, keyEvent, mainLoop );
return (0);
```

Figura 5.10 - Trecho de código do módulo de configuração.

Com esses dados armazenados, inicia-se a captura, quadro a quadro, das imagens do fluxo de vídeo através da webcam que aponta para o marcador do teto, iniciando o primeiro módulo.

Em cada imagem obtida da etapa anterior, será efetuada a detecção dos marcadores (Figura 5.11). Se não conseguir encontrar nenhum marcador na cena, o sistema obterá um novo frame e irá repetir o processo. Caso encontre um ou mais marcadores na cena, essa imagem será binarizada (convertida para preto e branco), e o sistema irá procurar a identificação de cada marcador, que será importante para determinação da matriz de

coordenadas do marcador em relação à câmera. Quando o processo estiver todo concluído, passa-se ao módulo seguinte.

```

/* grab a wide frame */
if( (dataPtr = (ARUint8 *)arVideoGetImage()) == NULL ) {
    arUtilSleep(2);
    return;
}
if( count == 0 ) arUtilTimerReset();
count++;

/* detect the markers in the video frame */
if( arDetectMarkerLite(dataPtr, thresh, &marker_info, &marker_num) < 0 ) {
    cleanup();
    exit(0);
}

argDrawMode2D();
if( !arDebug ) {
    argDispImage( dataPtr, 0,0 );
}
else {
    argDispImage( dataPtr, 1, 1 );
    if( arImageProcMode == AR_IMAGE_PROC_IN_HALF )
        argDispHalfImage( arImage, 0, 0 );
    else
        argDispImage( arImage, 0, 0 );

    glColor3f( 1.0, 0.0, 0.0 );
    glLineWidth( 1.0 );
    for( i = 0; i < marker_num; i++ ) {
        argDrawSquare( marker_info[i].vertex, 0, 0 );
    }
    glLineWidth( 1.0 );
}

arVideoCapNext();

```

Figura 5.11 - Trecho de código do módulo de procura por marcadores.

No segundo módulo (Figura 5.12), com o marcador já detectado e identificado, o sistema buscará a matriz de posição relativa do marcador em relação ao sistema de multimarcadores, obtendo a primeira matriz de transformação de coordenadas, das coordenadas do marcador em relação à câmera.

Com essa matriz obtida, efetua-se a conversão de sistema de coordenadas, transformando a matriz em uma matriz de coordenadas da câmera, em relação ao marcador.

```

if( (err=arMultiGetTransMat(marker_info, marker_num, config)) < 0 ) {
    argSwapBuffers();
    return;
}
printf("err = %f\n", err);
if(err > 100.0 ) {
    argSwapBuffers();
    return;
}
/*
for(i=0;i<3;i++) {
    for(j=0;j<4;j++) printf("%10.5f ", config->trans[i][j]);
    printf("\n");
}
printf("\n");
*/
argDrawMode3D();
argDraw3dCamera( 0, 0 );
glClearDepth( 1.0 );
glClear(GL_DEPTH_BUFFER_BIT);
for( i = 0; i < config->marker_num; i++ ) {
    if( config->marker[i].visible >= 0 ) {
        printf("%i: X=%.11f\tY=%.11f\t",i,config->marker[i].trans[0][3],config-
>marker[i].trans[1][3]);
        printf("\nZ=%.11f\n",config->trans[2][3]);
        arUtilMatInv(config->trans, cam_trans); //calcula matriz da camera
        //matriz de transformação
        pontosc[0] = cam_trans[0][0]*config->trans[0][3]+cam_trans[1][0]*config-
>trans[1][3]+cam_trans[2][0]*config->trans[2][3];
        pontosc[1] = cam_trans[0][1]*config->trans[0][3]+cam_trans[1][1]*config-
>trans[1][3]+cam_trans[2][1]*config->trans[2][3];
        pontosc[2] = cam_trans[0][2]*config->trans[0][3]+cam_trans[1][2]*config-
>trans[1][3]+cam_trans[2][2]*config->trans[2][3];
        pontosc[3] = 1;
        printf("\n\n");
        //mostra pontos convertidos:
        printf("\n\nCoordenadas da camera em relação ao marcador:\n X = %.11f\tY = %.11f\tZ =
%.11f",pontosc[0],pontosc[1],pontosc[2]);
    }
}

```

Figura 5.12 - Trecho de código do módulo de conversão de coordenadas.

Em seguida, o sistema realiza a conversão do sistema das coordenadas da câmera para o sistema de coordenadas do mundo real, utilizando os parâmetros, informados pelo usuário no módulo de configuração inicial, informados no início da execução (Figura 5.13).

Com o sistema de coordenadas da câmera no mundo real determinado, é possível obter os valores dos quaternions⁴, através da conversão de seus valores pela matriz de transformação.

```
//realiza conversao
printf("\n\nCoordenadas da camera no mundo real:\n");
printf("X = %.11f cms\n", (pontosc[0]-pontos[0])*(-1)/1);
printf("Y = %.11f cms\n", (pontosc[1]-pontos[1])*(-1)/1);
printf("Z = %.11f cms\n", (pontosc[2]-pontos[2])*(-1)/1);

//Quaternos - início
arUtilMat2QuatPos(cam_trans, quat, pos);
printf(" QUAT: Pos x: %3.1f y: %3.1f z: %3.1f\n", pos[0],
pos[1], pos[2]);
printf("      Quat qx: %3.2f qy: %3.2f qz: %3.2f qw: %3.2f\n", quat[0], quat[1], quat[2], quat[3]);

//fim quaternos

draw( config->trans, config->marker[i].trans, 0 );

    }
else
draw( config->trans, config->marker[i].trans, 1 );
}
argSwapBuffers();
```

Figura 5.13 - Trecho de código do módulo de obtenção dos quaternions.

Esse processo repetir-se-á desde o primeiro módulo para cada frame obtido até o fim da execução do sistema.

Para finalizar o sistema, basta pressionar “ESC” ou fechar a janela.

⁴ Um quaternion é um número complexo de quatro dimensões, que pode ser utilizado para representar a orientação de um frame de coordenada no espaço tridimensional (Craig, 2005).

5.4 CONSIDERAÇÕES FINAIS

A estruturação do sistema de rastreamento de câmeras idealizado foi desenvolvido, implementado e descrito neste capítulo.

Para a implementação do protótipo foi acoplado ao controle remoto do Nintendo Wii, o acessório Wii Plus que possui três giroscópios necessários para a orientação. Junto ao Wii também foi acoplado uma Webcam HD. Esse conjunto foi instalado em um tripé.

Para os testes de posicionamento do protótipo, utilizou-se a biblioteca ARToolkit, com multimarcadores presos no teto. Através do sistema desenvolvido foi possível encontrar as coordenadas da posição do protótipo.

Foram demarcados no ambiente real pontos cujas posições X, Y, e Z são conhecidas.

O software desenvolvido retornou o posicionamento do protótipo confrontando com os posicionamentos do mundo real.

Os dados do giroscópio do Wii Plus exibem os ângulos dos eixos X, Y para conhecer a orientação do protótipo.

A biblioteca ARToolkit foi utilizada, também para encontrar a orientação do protótipo no mundo real, utilizando-se dos quaternions. Foi feita, uma comparação entre os quaternions do ARToolkit e os quaternions do giroscópio do Wii Plus.

Uma vez que os dois equipamentos não possuem um hardware de sincronismos, manteve-se os dois equipamentos (câmera e Wii) ligados simultaneamente, por alguns segundos, na posição aproximada de (0,0,0). Após a coleta dos dados foi feito, manualmente, o ajuste do tempo de ambos os equipamentos.

Foi apresentada, também, a fundamentação matemática necessária para o correto funcionamento do sistema.

Os resultados desses estudos serão apresentados e comentados no capítulo 6.

6 TESTES E ANÁLISE DE RESULTADOS

Neste capítulo serão abordados os testes e resultado de posicionamento e orientação da câmera no ambiente de teste.

6.1 TESTES REALIZADOS

Neste capítulo, são descritos os testes realizados com o protótipo do software de rastreamento de câmera, e os resultados obtidos.

Deve-se esclarecer que os testes foram realizados com a finalidade de comprovar que, com base em tecnologias de rastreamento, e de acordo com Piekarski et al. (2002), e Malbezin (2002), descritos na seção 4.1.3, é possível realizar o rastreamento de câmera dentro dos padrões estabelecidos pelos autores, com resultados satisfatórios.

Para uma maior confiabilidade e acurácia nos testes, foram feitas as calibrações e testes nos seguintes modelos de Webcam, Multilaser WC044, Speed C130, Logtech quickcam e Logtech C270 HD. A escolha pelo modelo Logtec C270 deu-se pela menor distorção e a distancia focal da lente e o leitor de CCD.

6.2 PREPARAÇÃO DO AMBIENTE

Para verificar a eficácia do estudo, utilizou-se um ambiente de 4,50 m de comprimento, 4,0 m de largura e 2,90 m de altura, no qual foi demarcado um espaço de 3x3x2,90m para serem efetuados os testes (Figura 6.1).

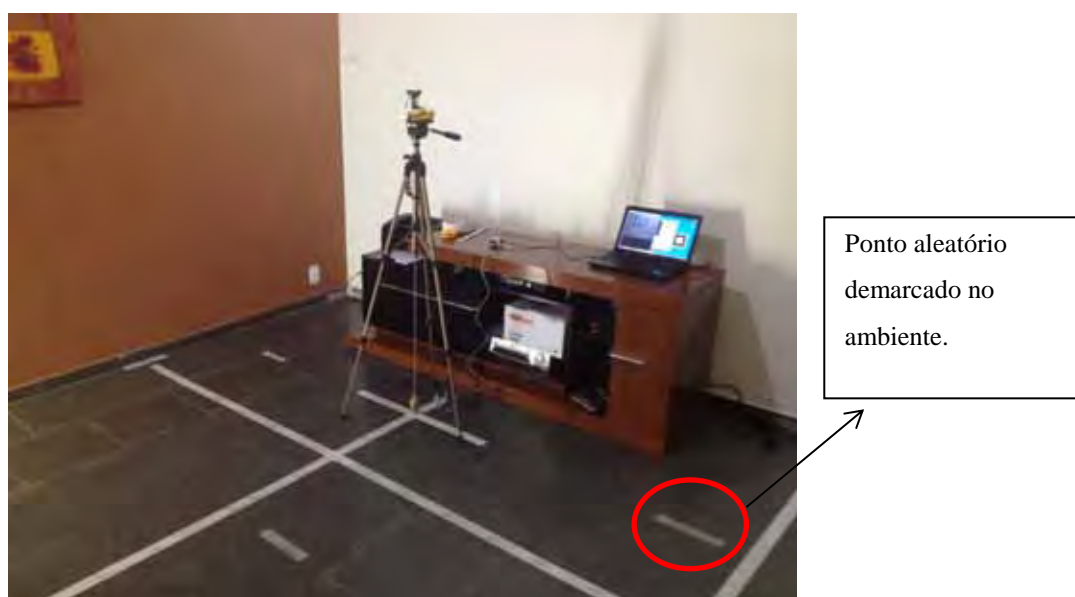


Figura 6.1 - Demarcação dos pontos no ambiente.

Para maior acurácia dos testes, foram demarcados, no ambiente real, 8 (oito) pontos aleatórios, usados como calibração, verificação e ajustes dos equipamentos. Esses pontos foram aferidos com diversos equipamentos diferentes. O procedimento para a criação de cada ponto é descrito a seguir:

- Para uma melhor movimentação e para se obter um ambiente mais controlado, medindo 4,5x4,0x2,90m, fez-se a demarcação do ambiente para 3x3x2,90m. Escolheram-se 8 (oito) pontos aleatoriamente, dentro dessa nova demarcação, os quais foram aferidos com vários equipamentos diferentes, tais como: uma trena metálica, Figura 6.2, uma trena ultrassônica, Figura 6.3, e uma trena a laser, Figura 6.4.
- Todos os 8 (oito) pontos foram demarcados no chão do ambiente, sabendo-se, em milímetros, os eixos X, Y e Z de cada um deles (Figura 6.5).

Um laser de linha foi utilizado para garantir o alinhamento dos pontos e para que os mesmos não ficassem fora do esquadro (Figura 6.6).



Figura 6.2 - Aferição com trena metálica.



Figura 6.3 - Aferição com trena Ultrason.



Figura 6.4 – Trena de laser.



Figura 6.5 - Pontos demarcados no ambiente.



Figura 6.6 - Laser de linha.

Foi demarcado, em uma parede à frente da câmera, um ponto denominado ponto “inicial”, Figura 6.7, que está em $X=1500\text{mm}$, $Y=1500\text{mm}$ e $Z=0\text{mm}$, no ambiente real, alocado em uma parede logo à frente da câmera, para facilitar os ajustes e primeiros testes.

Nesse ponto, foi inserido um marcador fiducial de $8\text{cm} \times 8\text{cm}$, cujo centro do marcador coincida, exatamente, com o ponto “inicial”. O propósito de um único marcador inicialmente foi para facilitar os ajustes e os primeiros testes. Esses primeiros ajustes foram

feitos não só para aprimorar os testes e as aferições das medidas apresentadas pelo software, mas também para compará-las com as medidas do mundo real.

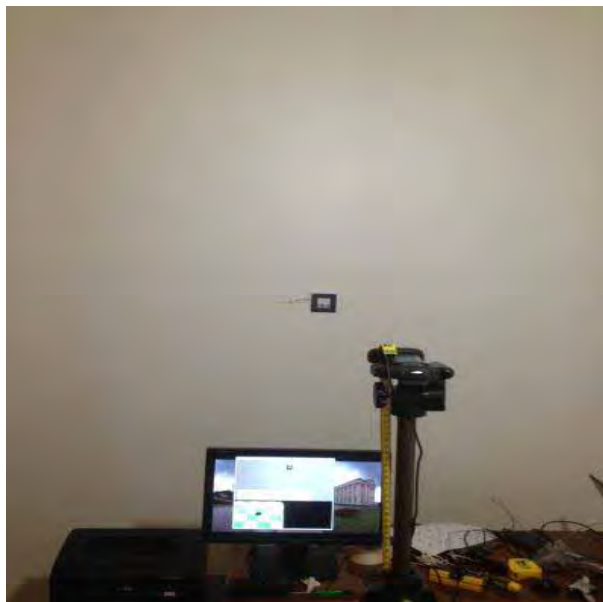


Figura 6.7 - Ponto de marcação “ponto inicial”.

A câmera está posicionada em um tripé, com um prumo fixado exatamente ao centro do tripé e ao centro focal da câmera, para uma melhor acurácia dos testes (Figura 6.8).



Figura 6. 8 - Tripé montado com o prumo.

O tripé, também, foi posicionado em um ponto denominado marco “0 “ (zero), Figura 6.9, que está colocado no ambiente real em $X=1500\text{mm}$, $Y=1500\text{mm}$ e $Z=1500\text{mm}$. Para facilitar as calibrações neste ponto, um laser de linha foi utilizado para uma maior precisão do posicionamento do mesmo.



Figura 6.9 - Posicionamento no “marco 0” com o laser de linha.

6.3 TESTES EXECUTADOS

No segmento da demarcação dos 8 (oito) pontos no ambiente, verificou-se a acurácia do software em relação aos pontos demarcados, utilizando o marcador à frente.

Após essa calibração e verificação, foram criadas duas faixas com 10 (dez) pontos demarcados no eixo X, e outra faixa demarcada com 10 (dez) pontos no eixo Y (Figura 6.10), e inserido o multimarcador no teto (Figura 6.11). O multimarcador foi posicionado em $X=1500\text{mm}$, $Y=1500\text{mm}$, e $Z=0\text{mm}$, lembrando que esses dados são referentes ao centro dos 6 (seis) marcadores, que formam o multimarcador, para finalmente definir a acurácia dos testes.



Figura 6.10 - Demarcação dos 10 pontos em X e 10 pontos em Y.



Centro do
multimarcador.

Figura 6.11 - Multimarcador fixado no teto.

Os primeiros dez testes foram efetuados variando-se a posição do equipamento apenas com relação ao eixo X, mantendo a câmera sempre com um ângulo reto, entre o multimarcador e o ator.

Para efetuar os próximos 10 testes, manteve-se os eixos X e Z fixos, variando-se, apenas, o eixo Y e não alterando os ângulos da orientação.

O mesmo procedimento foi executado com relação ao eixo Z.

Outro teste efetuado foi em relação à inclinação da câmera. Para esses testes, foram acoplados, junto à câmera, o controle do Wii com o Wii Plus, e um laser de ponto para mostrar o apontamento exato da linha de visão.

As inclinações feitas para os testes ficaram sempre entre $+30^\circ$, -30° em X e em Y para que a câmera sempre tenha uma linha de visão com o ator.

A comparação das informações de rotação foram realizadas adotando-se o formato dos quaternions.

Deve-se ressaltar que a iluminação do ambiente foi controlada com duas fluorescentes de 40w cada uma, num total de 80w, localizadas, sempre, do lado oposto dos marcadores.

6.4 ANÁLISE DOS RESULTADOS EXTRAÍDOS

6.4.1 TESTES DE POSICIONAMENTO

Ao término dos testes da posição, utilizando-se uma variação de dez pontos em cada eixo, mantendo-se os outros dois pontos fixos e conservando-se, também, fixos a orientação X (pitch) e Y (roll), os dados do Wii plus foram obtidos. A seguir, os resultados serão mostrados nas tabelas e figuras.

Variando o eixo X e mantendo os eixos Y e Z fixos, tem-se uma média de erro de 3,65%. A tabela 2 e a figura 6.12 exibem os resultados obtidos.

Tabela 2 - Variação no Eixo X

Ponto	Eixo X			Eixo X			Eixo X			Variação no Eixo X		
	Valores medidos			Valores obtidos do sistema			Correção segundo Piekarski (2002)			Porcentagem		
	X	Y	Z	X	Y	Z	X	Y	Z	X	Y	Z
1	700	1500	1500	739	1444	1518	39	56	18	5,57%	3,73%	1,20%
2	900	1500	1500	947	1467	1512	47	33	12	5,22%	2,20%	0,80%
3	1100	1500	1500	1150	1478	1525	50	22	25	4,55%	1,47%	1,67%
4	1300	1500	1500	1319	1518	1492	19	18	8	1,46%	1,20%	0,53%
5	1500	1500	1500	1478	1527	1515	22	27	15	1,47%	1,80%	1,00%
6	1700	1500	1500	1731	1553	1487	31	53	13	1,82%	3,53%	0,87%
7	1900	1500	1500	1963	1562	1535	63	62	35	3,32%	4,13%	2,33%
8	2100	1500	1500	2186	1568	1549	86	68	49	4,10%	4,53%	3,27%
9	2300	1500	1500	2399	1573	1561	99	73	61	4,30%	4,87%	4,07%
10	2500	1500	1500	2618	1582	1579	118	82	79	4,72%	5,47%	5,27%

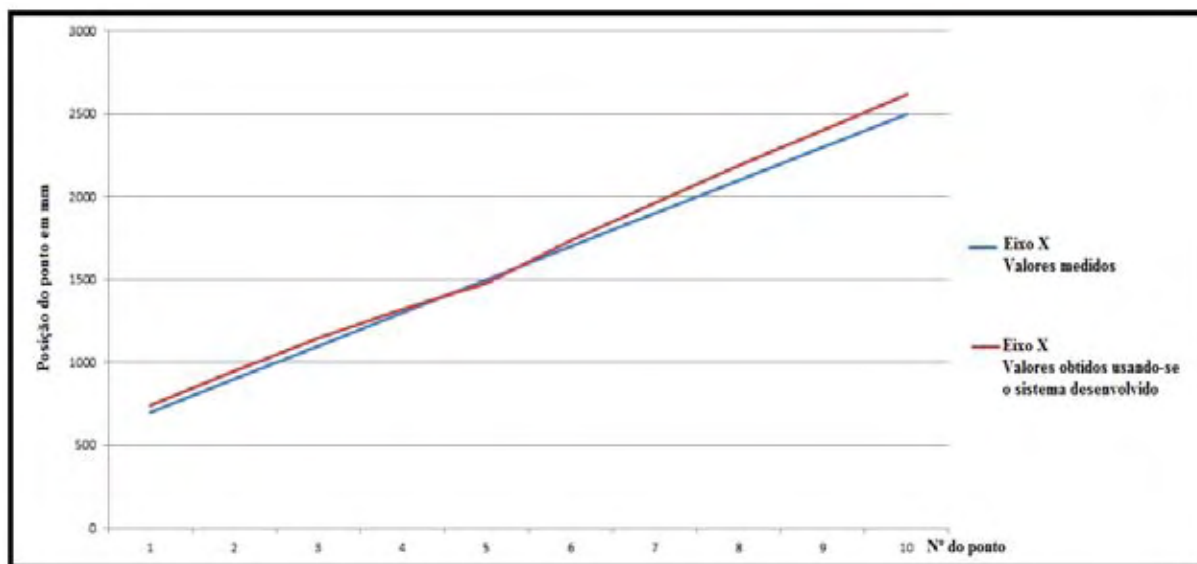


Figura 6.12 -Variação do resultado do eixo X entre os valores medidos, e o resultado obtido pelo software desenvolvido.

Variando o eixo Y e mantendo os eixos X e Z fixos, tem-se uma média de erro de 11,18%. A tabela 3 e a figura 6.13 exibem os resultados obtidos.

Tabela 3 – Variação no Eixo Y

Ponto	Eixo Y			Eixo Y			Eixo Y			Variação no Eixo Y		
	Valores medidos			Valores obtidos do sistema			Correção segundo Piekarski (2002)			Porcentagem		
	X	Y	Z	X	Y	Z	X	Y	Z	X	Y	Z
1	1500	1000	1500	1463	876	1471	37	124	29	2,47%	12,40%	1,93%
2	1500	1300	1500	1481	1218	1483	19	82	17	1,27%	6,31%	1,13%
3	1500	1500	1500	1519	1532	1527	19	32	27	1,27%	2,13%	1,80%
4	1500	1700	1500	1560	1598	1547	60	102	47	4,00%	6,00%	3,13%
5	1500	1900	1500	1597	1670	1579	97	230	79	6,47%	12,11%	5,27%
6	1500	2000	1500	1618	1764	1601	118	236	101	7,87%	11,80%	6,73%
7	1500	2300	1500	1634	1987	1640	134	313	140	8,93%	13,61%	9,33%
8	1500	2500	1500	1652	2139	1654	152	361	154	10,13%	14,44%	10,27%
9	1500	2700	1500	1679	2277	1677	179	423	177	11,93%	15,67%	11,80%
10	1500	3000	1500	1698	2481	1699	198	519	199	13,20%	17,30%	13,27%

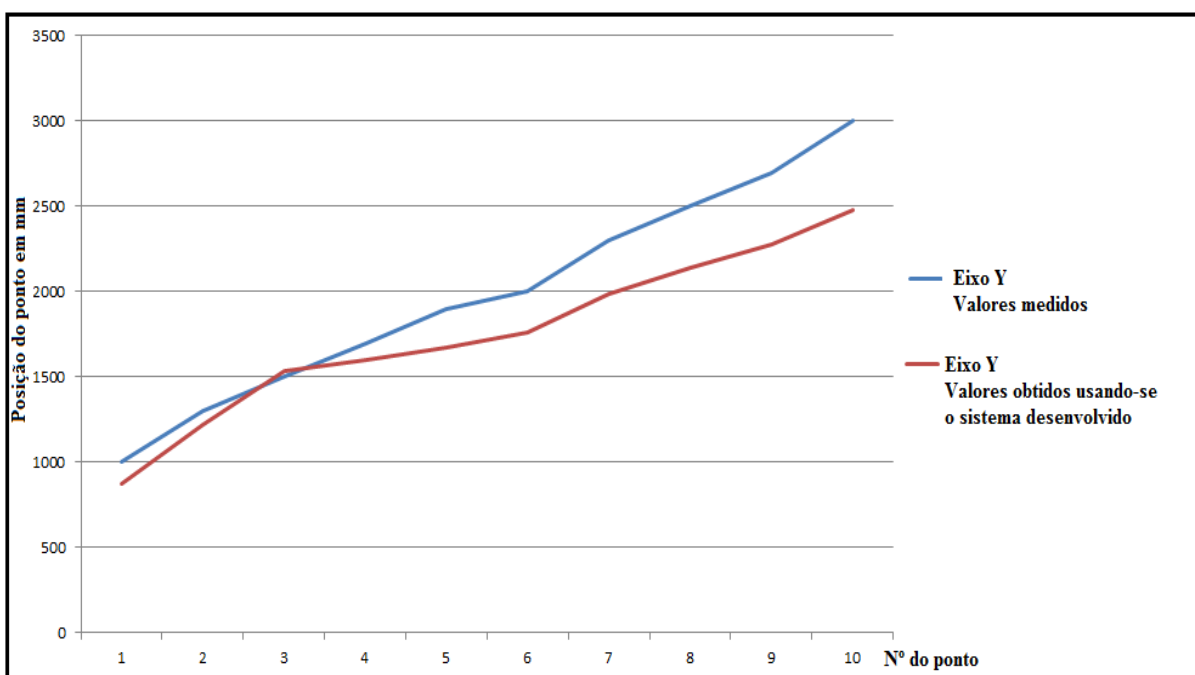


Figura 6.13 - Variação do resultado do eixo Y entre os valores medidos, e o resultado obtido pelo software desenvolvido.

Variando o eixo Z e mantendo os eixos X e Y fixos, tem-se uma média de erro de 4,54%. A tabela 4 e a figura 6.14 exibem os resultados obtidos.

Tabela 4 – Variação no Eixo Z

Ponto	Eixo Z			Eixo Z			Eixo Z			Variação no Eixo Z		
	Valores medidos			Valores obtidos do sistema			Correção segundo Piekarski (2002)			Porcentagem		
	X	Y	Z	X	Y	Z	X	Y	Z	X	Y	Z
1	1500	1500	900	1514	1477	911	14	23	11	0,93%	1,53%	1,22%
2	1500	1500	1000	1518	1495	1027	18	5	27	1,20%	0,33%	2,70%
3	1500	1500	1100	1533	1507	1143	33	7	43	2,20%	0,47%	3,91%
4	1500	1500	1200	1552	1540	1255	52	40	55	3,47%	2,67%	4,58%
5	1500	1500	1300	1564	1553	1340	64	53	40	4,27%	3,53%	3,08%
6	1500	1500	1400	1532	1569	1478	32	69	78	2,13%	4,60%	5,57%
7	1500	1500	1500	1524	1531	1579	24	31	79	1,60%	2,07%	5,27%
8	1500	1500	1600	1579	1552	1687	79	52	87	5,27%	3,47%	5,44%
9	1500	1500	1700	1557	1566	1803	57	66	103	3,80%	4,40%	6,06%
10	1500	1500	1800	1581	1573	1937	81	73	137	5,40%	4,87%	7,61%

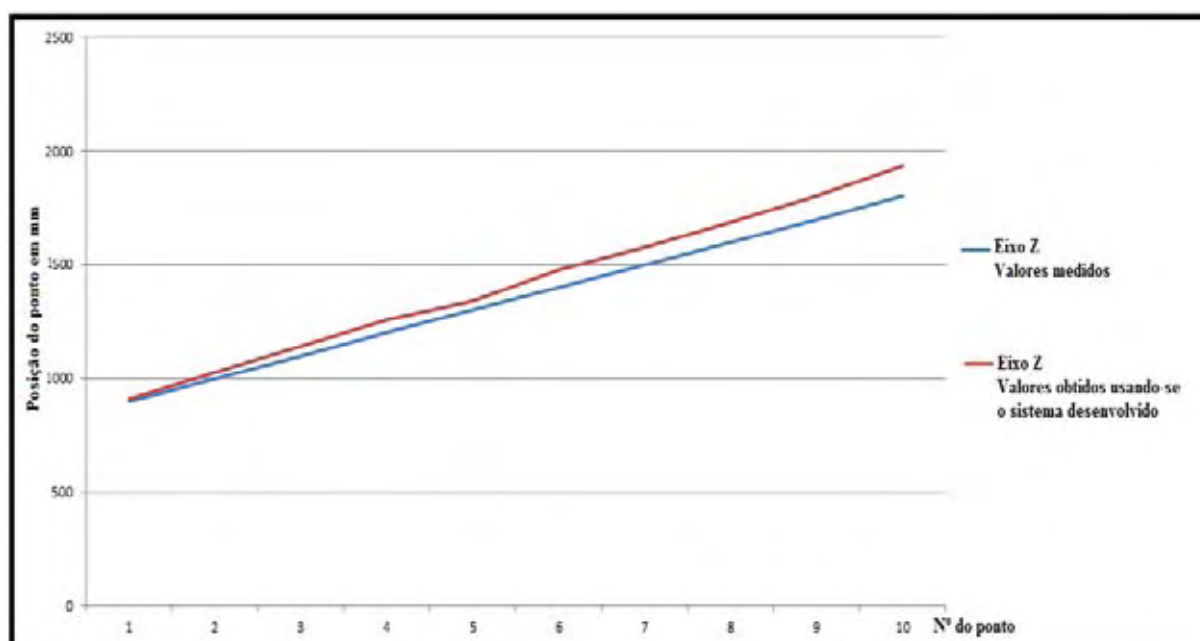


Figura 6.14 - Variação do resultado do eixo Z entre os valores medidos, e o resultado obtido pelo software desenvolvido.

6.4.2 TESTES RELATIVOS À ROTAÇÃO

Para o teste de orientação, foram utilizados os dados provenientes do sensor óptico provindos da câmera, (quaternion) e comparados com os dados do sensor inercial (Wii plus). Um iPhone com um software baseado no giroscópio foi acoplado ao sistema de rastreamento para auxiliar a demonstração dos ângulos X e Y (Figura 6.15).

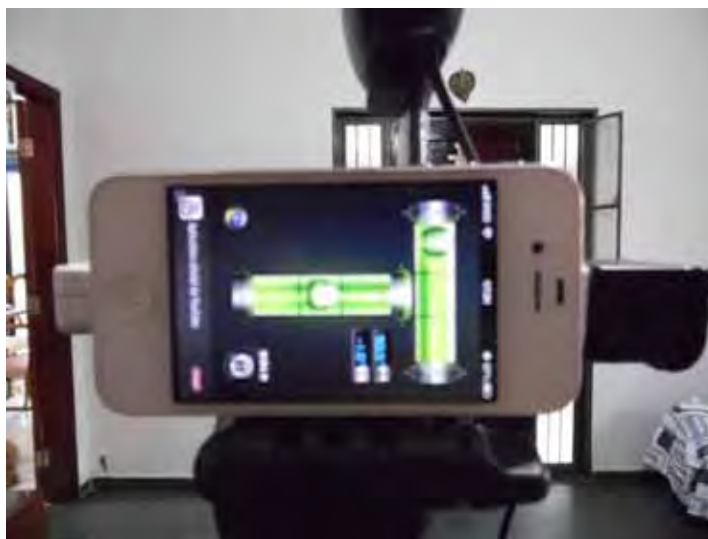


Figura 6.15 – Verificação dos ângulos.

Para efetuar as leituras dos dados da orientação da câmera, X (Pitch), Y (Roll) e Z, (Yaw), foram utilizados os mesmos pontos demarcados no teste de posicionamento, porém sempre com a câmera em movimento, ao contrário dos testes de posicionamento, que foram efetuados com a câmera parada no ponto demarcado.

A figura 6.16 demonstra os resultados obtidos através do sensor do Wii plus, cuja câmera foi movimentada e inclinada em $+30^\circ$ a -30° em X (Pitch), Y (Roll), e mantendo o Z (Yaw) estabilizado para sempre manter o foco no ator.

A Figura 6.16 exibe os dados do quaternion do Wii Plus com uma inclinação de 30° positivo e negativo para X e Y durante aproximadamente 1 minuto o eixo Z não foi alterado durante os testes.

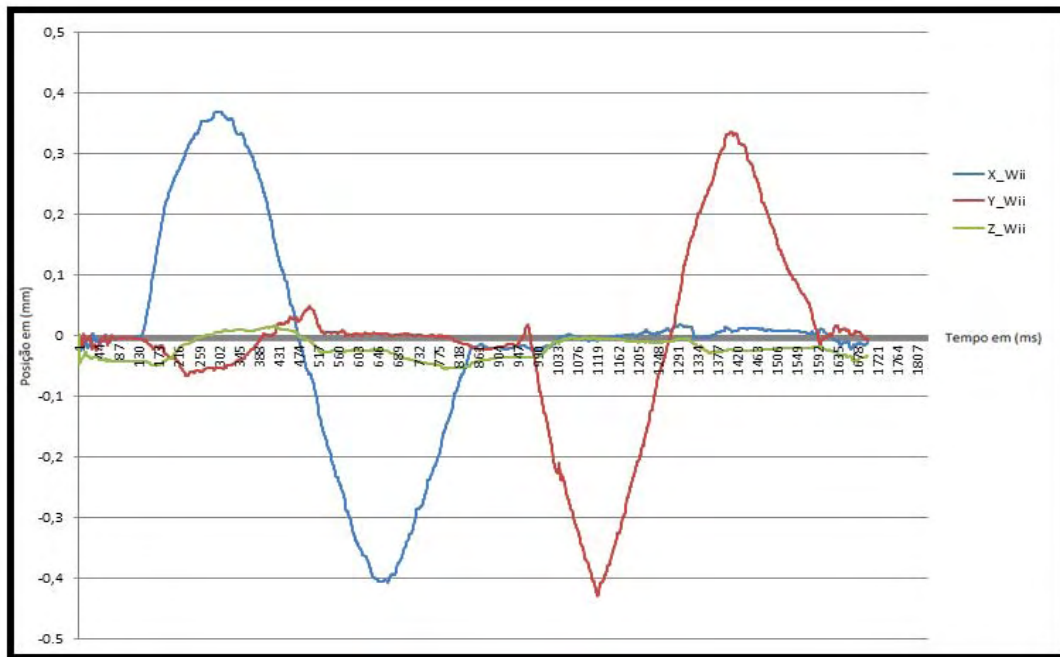


Figura 6.16 - Orientação obtida pelo Wii Plus.

Já a figura 6.17 exibe os dados do quaternion da câmera com uma inclinação de 30° positivo e negativo para X e Y capturados ao mesmo tempo do Wii Plus.

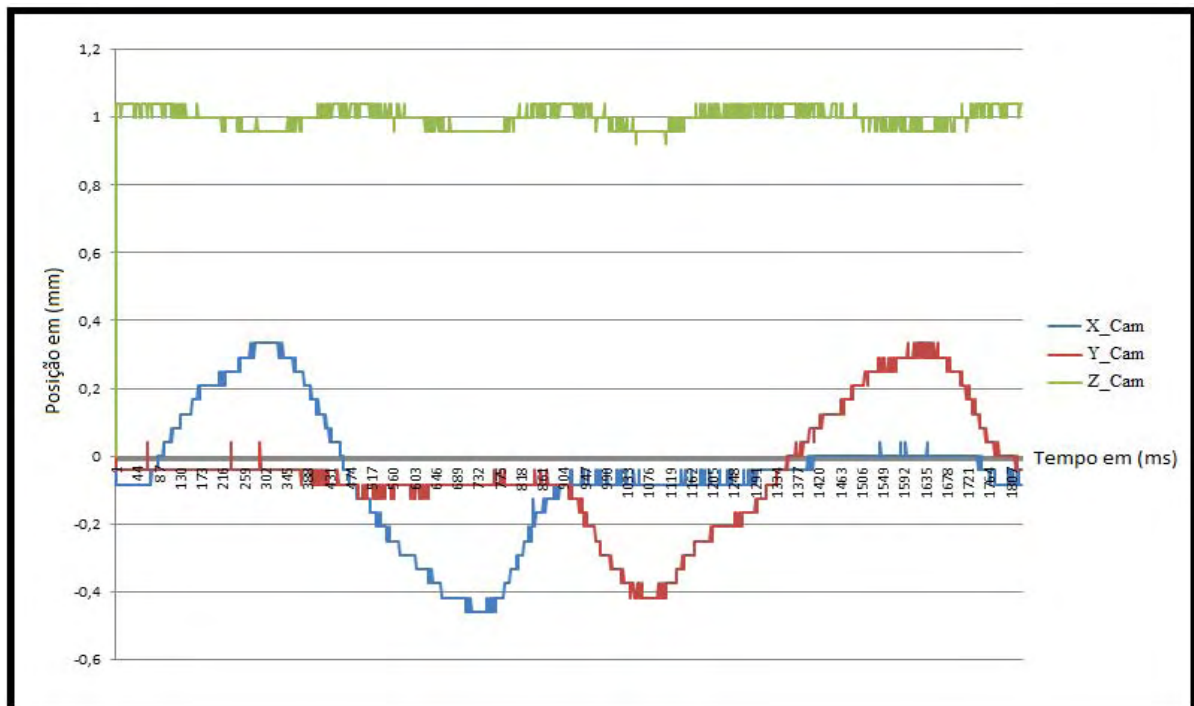


Figura 6.17 - Medidas dos ângulos da câmera (durante o movimento de rotação), representadas na forma de quaternion.

Para uma melhor comparação entre os dois métodos, a figura 6.18 demonstra o comparativo entre os quaternions do Wii e os quaternions da câmera para o eixo X (Pitch), os quais foram efetuados simultaneamente para que os pontos e o tempo sejam exatamente os mesmos.

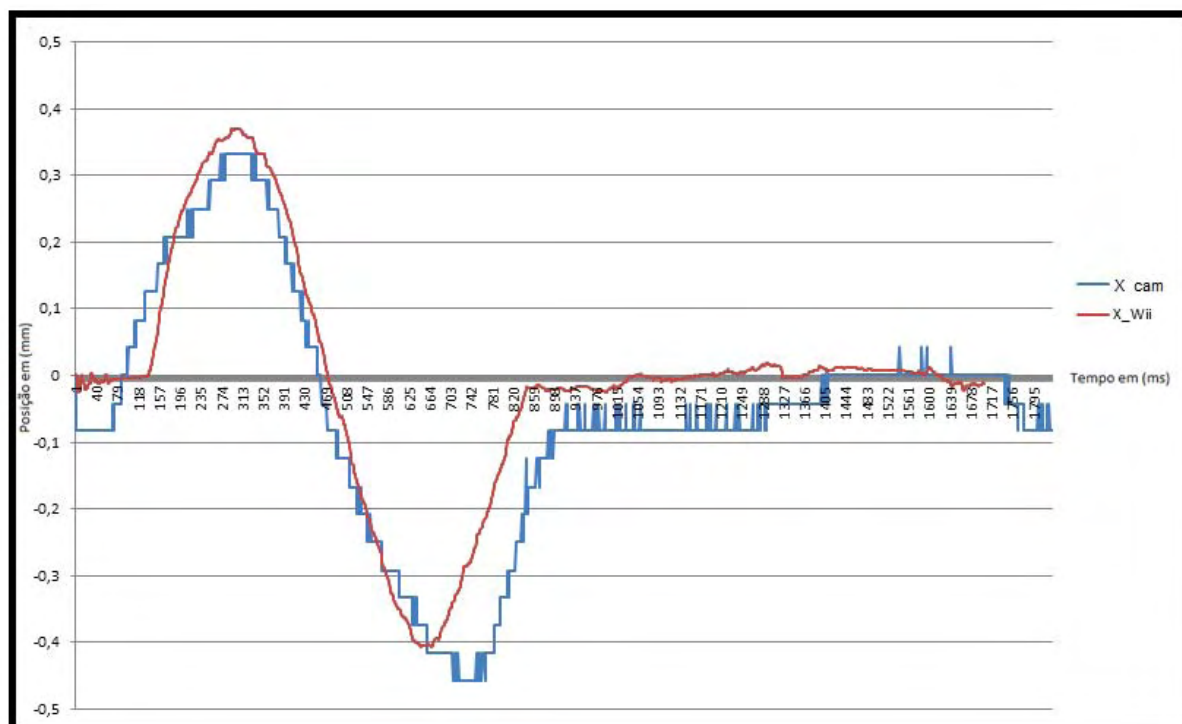


Figura 6.18 - Comparação do eixo X entre os quaternions do Wii e os quaternions da webcam.

A figura 6.18 exibe uma variação de 0,08m na posição e de 36ms para o eixo do tempo, uma variação considerável segundo Thomas et al. (1997) capítulo 3.

O mesmo ocorreu com a figura 6.19 eixo Y (Roll).

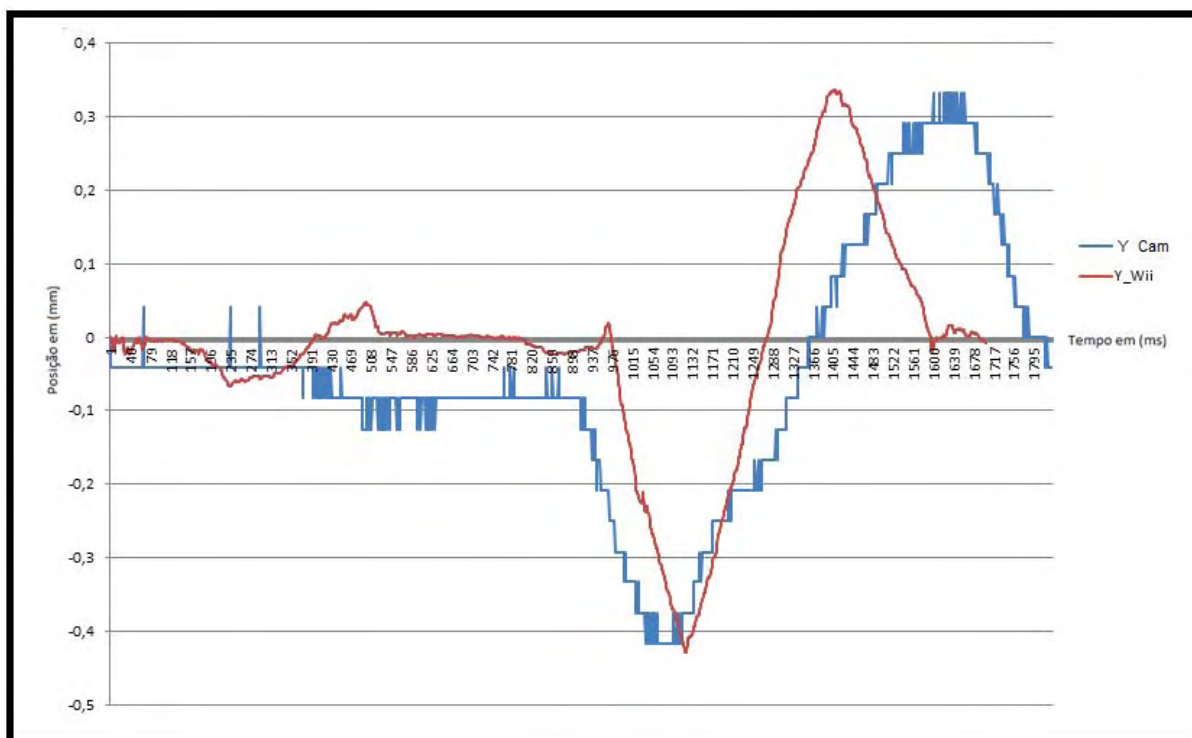


Figura 6.19 - Comparação do eixo Y entre os quaternions do Wii e os quaternions da webcam.

A figura 6.20 exibe os resultados do eixo Z (Yaw) do quaternion do Wii em comparação com o quaternion da câmera. Os resultados apresentados demonstram que a diferença constatada, deve-se ao posicionamento de 90° existente entre eles.

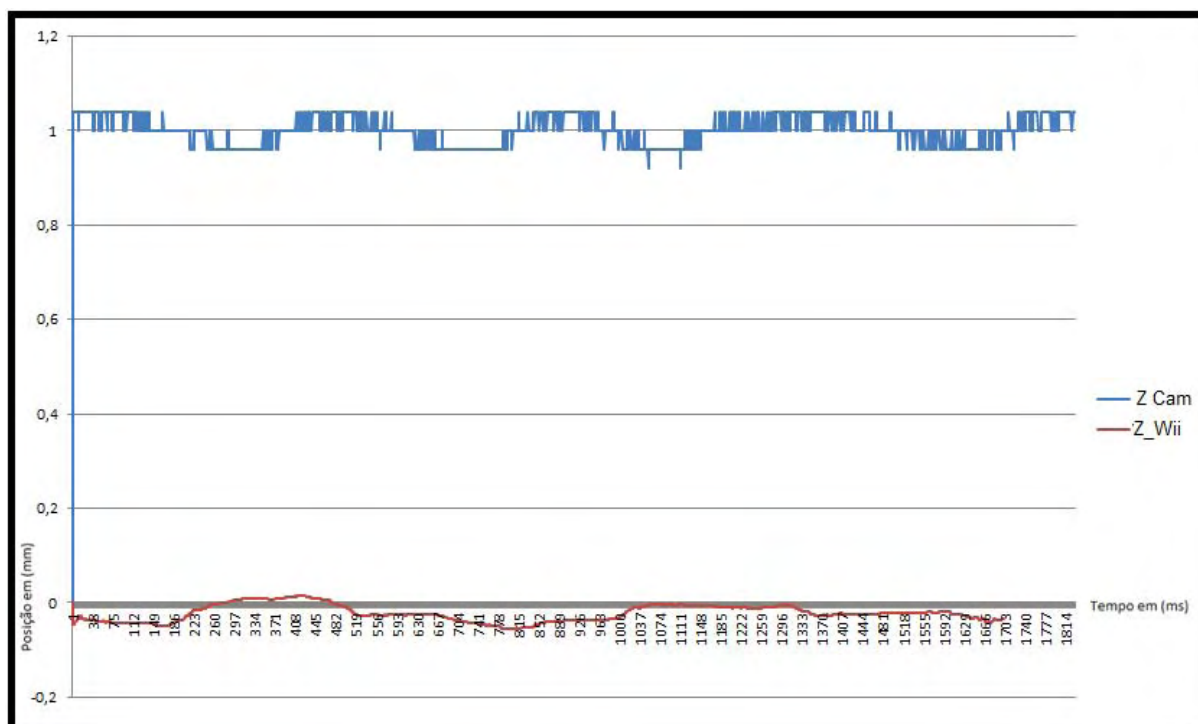


Figura 6.20 - Comparação do eixo Z entre os quaternions do Wii e os quaternions da webcam.

6.5 CONSIDERAÇÕES FINAIS

Os resultados dos testes de posicionamento analisados obtiveram uma média de 3,65% de variação no eixo X; 11,18% no eixo Y; 4,54% no eixo Z, para uma distância média de 1,5m e inclinação de 30°, estando, portanto, dentro dos padrões estabelecidos por Malbezin et. al (2002), os quais apresentaram, em sua pesquisa, uma variação de erro de 9% a 18% no eixo Y e de 6% a 12% no X, considerando-se uma distância de 1m a 2,5m e uma inclinação de 30° a 40°.

Os testes de comparação entre os quaternions da Webcam e os quaternions do wii plus demonstraram uma pequena variação nos eixos X e Y entre os dois sistemas de rastreamento (sistema óptico e sistema inercial), já o eixo Z demonstra muito pequena, porem com uma diferença de 90° entre elas, pois os equipamentos estão direcionados desta forma, a qual não foi avaliada nos testes de Malbezin et. al (2002), comprovando, assim, a eficácia dos dois métodos aplicados.

7 CONCLUSÕES

Embasado nos testes realizados no protótipo desenvolvido, é possível concluir que o protótipo do software e hardware de rastreamento de câmeras mostrou-se viável para o rastreamento de câmera em estúdios virtuais.

Os testes realizados mostraram, no cap. II, uma visão geral dos tipos de rastreamento. Devido a precisão, alcance e por não ser suscetível a ruídos magnéticos, optou-se por utilizar o rastreamento óptico.

Com relação à mensuração da rotação da câmera, fez-se uso do rastreador inercial, baseado no controle remoto do Nintendo Wii acoplado com o Wii Plus.

Tendo como ponto de partida os trabalhos correlatos citados no capítulo 3, foi possível chegar a um resultado não apenas satisfatório, mas que também possibilitou criar um protótipo com baixo custo.

Tendo por base os testes realizados, demonstrou-se que o rastreamento de câmera em estúdio funciona com uma acurácia adequada, podendo ser inseridos objetos 3D na cena.

Deve-se esclarecer que os testes realizados mostram que a acurácia dos valores obtidos em cada eixo x, y e z aumentam com a quantidade de marcadores espalhados pelo teto, tornando o rastreamento cada vez mais preciso.

Como trabalhos futuros, pretende-se utilizar os resultados obtidos a partir do sistema desenvolvido para auxiliar a inserção de objetos 3D em tempo real, com relação à cena capturada em estúdio.

Um outro desdobramento desta pesquisa seria a utilização de equipamentos portáteis, tais como smartphones e tables, os quais possuem câmeras e giroscópios embutidos, como substitutos da webcam e do Wii Motion Plus.

REFERÊNCIAS BIBLIOGRÁFICAS

ARTOOLKIT HOMEPAGE, 2007. Disponível em <http://www.hitl.washington.edu/artoolkit/>. Acesso em 21 de fevereiro de 2012.

BEEBY, S.; ENSELL, G.; KRAFT, M.; WHITE, N.; **MEMS Mechanical Sensors.**: Artech House, Inc. Norwood, Massachusetts, USA. 2004.

CHANDARI, J, et al. **Real time camera tracking in the MATRIS Project.** IBC, set. Amsterdam, 2006.

CRAIG, J. J. **Introduction to Robotics Mechanics and Control.** Pearson Education International ,2005.

BUILDING, Willian V.; **Free-D Virtual Studio System** (2009). Documento eletrônico disponível em <http://www.vintenradamec.com>. Acessado em 20 de janeiro de 2013.

FURNISS, M. **Motion Capture.** 2009. Disponível em <http://web.mit.edu/comm-forum/papers/furniss.html>. Acesso em 15 de Novembro de 2012.

GIBBS, Simon; BAUDESCH, Patrick; **Interaction in the Virtual Studio.** German National Research Center for Information Technology, pp. 29-32. Sankt Augustin, Germany, 1996.

GRAU, O.; PRICE, M.; THOMAS, G. A. **Use of 3-D Techniques for Virtual Production.** BBC Research & Development, White Paper WHP 033, Londres, 2002.

GOMIDE, J. V. B. **Captura Digital de Movimento no Cinema de Animação.** Dissertação (Mestrado em Arte e Tecnologia da Imagem.), Escola de Belas Artes Universidade Federal de Minas Gerais, (EBA/UFGM). Belo Horizonte, 2006.

HIT LAB. **ARToolKit.** Disponível em: <http://www.hitl.washington.edu/artoolkit> Acesso em: 21 de setembro. 2012.

JESSE, Russell; RONALD, Cohn; **Ransac**. Book on demand, 2012.

KATO, H. **Inside Artoolkit**. Disponível em :
<www.hitl.washington.edu/artoolkit/Papers/ART02-Tutorial.pdf> Acesso em: 4 de outubro de 2012.

KATO, H.; BILLINGHURST, M. **Marker tracking and hmd calibration for a video-based augmented reality conferencing system**. In: IWAR '99: proceedings of the 2nd, IEEE and ACM International Workshop on Augmented Reality. IEEE Computer Society, p; 85 Washington, DC, USA 1999.

KÜHN, T. (2011). **The Kinect Sensor Plataform**. *Advances in Media Technology*, pp. 1-4, 2011.

LEAP MOTION. Manual do fabricate. Disponível em <www.leapmotion.com>. Acessado em 22 de julho de 2013.

LIVERMAN, M.; **The Animator's Motion Capture Guide**: Organizing, Managing and Editing. Charles River Media, Burlington, MA, USA, 2004.

MALUF, N.; WILLIAMS, K.; **An Introduction to Microelectromechanical Systems Engineering**, Second Edition. Artech House Norwood, Massachusetts, USA. 2004.

NINTENDO. **Manual do Nintendo Wii**. Disponível em <<http://wiibrew.org/wiki/Wiimote>>. Acesso em 20 de setembro de 2012. 2006.

NINTENDO. **Manual do Nintendo Wii**. Disponível em <<http://rosseto.wordpress.com/page/2/>>. Acesso em 10 de março de 2013. 2008.

MAHAJAN, V.; N.; **Optical Imaging and Aberrations - Part I: Ray Geometrical Optics**. SPIE Press, Bellingham, WA, USA, 1998.

MALBEZIN, Pierre, et al. **Measuring ARToolKit accuracy in long distance tracking experiments**. Austrália, 2002.

MENACHE, A.; **Understanding Motion Capture for Computer Animation and Video Games**. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002.

ORAD. **Mastering VideoGraphics** Disponível em <<http://www.orad.tv/pt/xync>>. Acesso em 15 de dezembro de 2012. 2012.

PIEKARSKI, W.; THOMAS, B. H.; **The tinmith system: demonstrating new techniques for mobile augmented reality modelling**. In: Proceedings of the third Australasian Conference on User Interfaces. p 48; p. 61-70. Darlinghurst, Australia. 2002,

PINTARIC, T.; **An adaptive thresholding algorithm for the augmented reality toolkit**. In: Augmented reality toolkit workshop. IEEE International, p. 71. 2003.

PIZZOTTI, Ricardo; **Enciclopédia básica da mídia electronica**. Editora Senac. São Paulo, 2003.

RASKAR, R. et al. **Lighting aware motion capture using photosensing markers and multiplexed illuminators**. ACM Transactions on Graphics, 26(3):36. Disponível em <<http://doi.acm.org/10.1145/1276377.1276422>>. Acessado em 15 de fevereiro de 2013. 2007.

SILVA, F. W. S. V.; **Um Sistema de Animação Baseado em Movimento Capturado**. Dissertação (Mestrado em Computação Gráfica), COPPE/UFRJ, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 1998.

SIXENSE. **Manual do Razer Hydra**. Disponível em <<http://www.sixense.com>> Acesso em 22 de Julho de 2013.

STEGER, C.; Ulrich, M.; WIEDEMANN, C. **Machine Vision Algorithms and Applications**. Wiley-VCH, Weinheim, BW, DE., 2008.

THOMAS, G. A.; JIN, J., NIBLETT, T.; URQUHART, C.; **Versatile camera position measurement system for virtual reality TV productio**. IEE Conference Publication, p. 284-289. Amsterdam 12-16, Sep, 1997.

THOMAS, G. A.; **Real time camera pose estimation for augmenting sports scenes.** Conference on Visual Media Production, 2006.

VELHO, L.; JONAS, M. G.; ROMA, P. C., WAGNER, F. S. V. S., **Motion capture.** Apostila de referências para consulta interna do IMPA '96.

VICON Motion Systems. **Motion Capture Systems from Vicon.** Disponível em <<http://www.vicon.com>>. Acesso em 13 de Novembro de 2012. 2008.

ANEXO

1. CALIBRAÇÃO DE CÂMERAS

O processo de calibração, inicialmente, busca corresponder os pontos tridimensionais com suas respectivas projeções bidimensionais no plano da imagem. Desta forma, podemos identificar 2 sistemas de coordenadas, de modo que a correspondência entre esses pontos possa ser expressa como uma composição de transformações, que associe os pontos 3D aos 2D correspondentes.

Kato (2007) diz que é desejável saber, para uma maior precisão do rastreamento, os parâmetros da câmera. Para isso, é necessário efetuar alguns procedimentos de calibração com as bibliotecas ARToolkit.

Visando aumentar a precisão do rastreamento, algumas propriedades da calibração devem ser extraídas, tais como:

- Distorção da lente;
- Ponto central da câmera;
- Distância focal da câmera.

Essas calibrações devem ser executadas para aproximar a imagem da câmera com a imagem ideal, como mostra a (Figura 10.1).

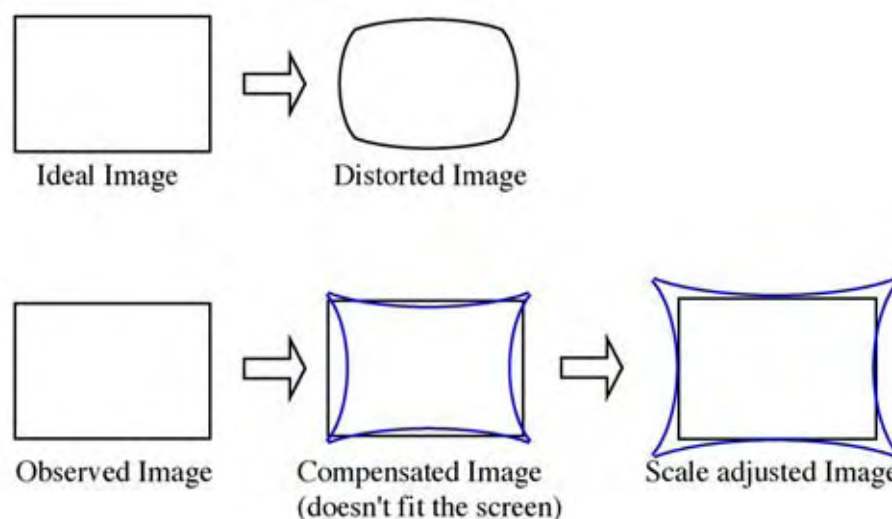


Figura 10.1 - Distorção da lente e imagem do mundo real.

Fonte: Kato, (2007)

Para uma correta calibração da câmera, os procedimentos estão descritos, passo a passo, a seguir:

➤ **Passo 1**

- Imprimir a imagem calib_dist.pdf. (Figura 10.2);
- Executar o calib_dist.exe;
- Capturar a imagem e marcar os 24 pontos em ordem da esquerda para direita e de cima para baixo;
- Repetir o último procedimento de 5 a 10 vezes em diferentes ângulos (quanto mais procedimentos maior a precisão);
- Clicar no botão do mouse para calcular a distorção (Figura 10.3);
- Salvar os resultados em um arquivo txt.

A figura 10.2 mostra os pontos que devem ser marcados em ordem de cima para baixo e da esquerda para a direita, somando um total de 24 pontos.



Figura 10.2 - Imagem do arquivo calib_dist.pdf.

Fonte: Kato (2007)

Após a marcação dos pontos em diversos ângulos mínimo de 5, o sistema gera a interligação dos pontos como mostra a Figura 10.3.

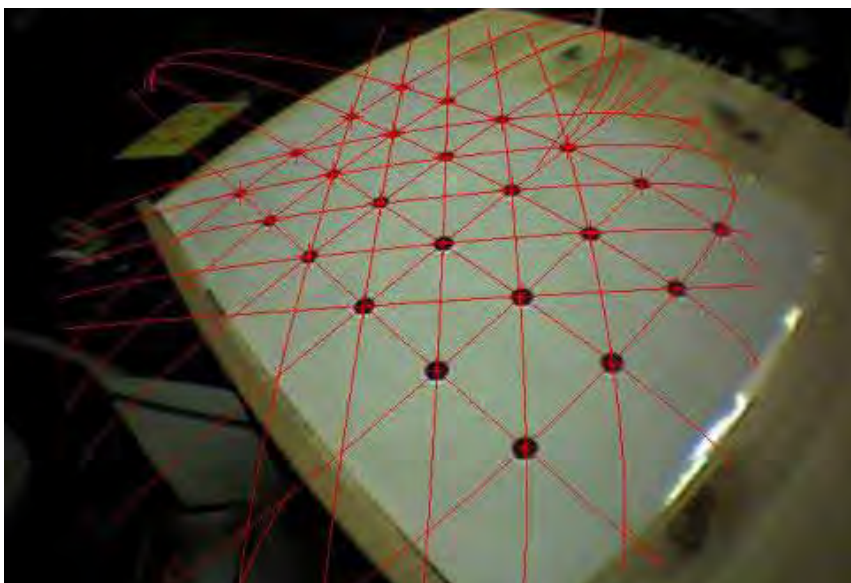


Figura 10.3 – Pontos interligados para cálculo da distorção `calib_dist.pdf`.

Fonte: Kato (2007)

➤ **Passo 2**

- Imprimir a imagem `calib_cparam.pdf`. (Figura 10.4);
- Rodar o executável `calib_cparam.exe`;
- Entrar com os parâmetros guardados no txt anterior, e com os parâmetros de linhas e colunas do arquivo `calib_cparam.pdf`;
- Capturar a imagem e definir as linhas horizontais e verticais sequencialmente, utilizando as setas do teclado e fazendo coincidir a linha branca com a azul;
- Repetir os testes do procedimento 5 vezes, afastando a cada interação a câmera em 10 cm cada vez (Figura 10.4);
- Definir o nome do arquivo como `câmera_para.dat`;
- Substituir o arquivo criado no diretório `bin/data`.

Para a Calibração da grade, é necessário enquadrar as linhas azuis exatamente sobre as linhas pretas, utilizando as setas do teclado, como mostra a (Figura 10.4).

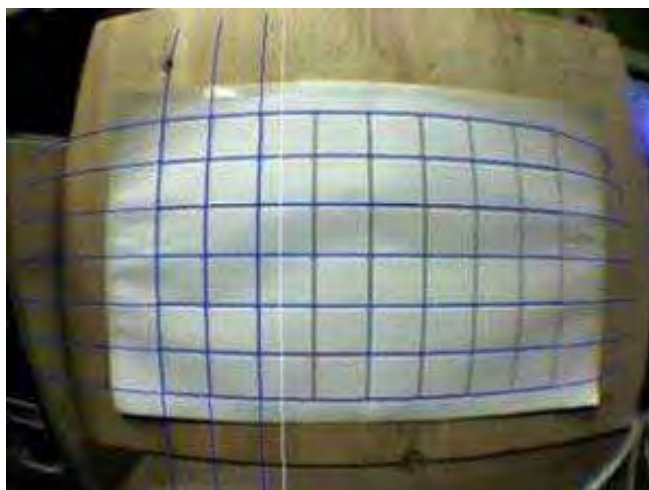


Figura 10.4 - Imagem do arquivo calicalib_cpam.pdf.

Fonte: Kato (2007)

Como dito no passo 2 de calibração das câmeras, deve-se afastar a câmera de 10 em 10 cm, para efetuar a calibração das grades, ressaltando, ainda, que esse procedimento deve ser feito 5 vezes, como mostra a (Figura 10.5).

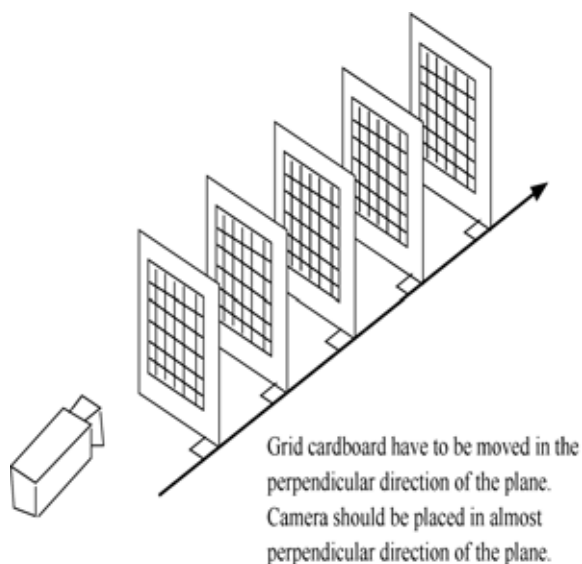


Figura 10.5 - Posicionamento da câmera afastando de 10 em 10 cm.

Fonte: Kato (2007)

Mesmo após a calibração das câmeras, é preciso ainda corrigir os parâmetros manualmente para que o centro da imagem seja o centro da câmera e melhorar a precisão do rastreamento, segundo Piekarski (2001).

Na Figura 10.6 (a), temos a definição em C par ARParam do Artoolkit, do modelo de distorção da câmera; na Figura 10.6 (b), acrescentadas as variáveis para centralizar a imagem real ao centro da câmera:

<pre>/* Actual struct */ typedef struct { int xsize, ysize; double mat[3][4]; double dist_factor[4]; } ARParam;</pre>	<pre>/* Components explained */ typedef struct { int cam_width, cam_height; double matrix [3][4]; double centre_x, centre_y; double focal, size; } ARParam;</pre>
---	---

(a)

(b)

Figura 10.6 - Código do ARToolkit.

(a) Definição do código C ARParam para o modelo de distorção, (b) Criação das variáveis para o centro da câmera em x e y.

Fonte: Piekarski (2001)

A Figura 10.7 mostra os parâmetros do arquivo câmera_para.dat gerado pelo padrão ARToolkit, logo após a calibração da câmera, com um erro de x=2,5 e y=48,0:

camera = (640, 480)		780.54	0.54	304.64	0.00	
centre = (317.5, 192.0)		0.00	762.30	208.68	0.00	
focal = 26.300000		0.00	0.00	1.00	0.00	
size = 1.009989		0.00	0.00	0.00	1.00	

Figura 10.7 - Arquivo câmera_para.dat com erro em X 2,5 e Y 48.

Fonte: Piekarski (2001)

Criou-se um código para corrigir a câmera ortogonal para um modelo de câmera perfeitamente simétrico com a finalidade de coincidir os eixos (Figura 10.8):

```

centre_x = cam_width / 2.0;
centre_y = cam_height / 2.0;
matrix[0][2] = cam_width / 2.0;
matrix[1][2] = cam_height / 2.0;

```

Figura 10.8 – Código ARToolkit

Fonte: Piekarski (2001)

Código para correção dos eixos da câmera para um modelo perfeitamente simétrico.

A figura 10.9 mostra a matriz após a correção do arquivo câmera_para.dat para o uso do rastreador com o frustum.

camera = (640, 480)		780.54	0.54	320.00	0.00	
centre = (320.0, 240.0)		0.00	762.30	240.00	0.00	
focal = 26.300000		0.00	0.00	1.00	0.00	
size = 1.009989		0.00	0.00	0.00	1.00	

Figura 10.9 - Correção do arquivo câmera_para.dat com os eixos ortogonais.

Fonte: Piekarski (2001)

2. CADASTRO DE MARCADORES

Para o cadastramento de novos marcadores, é preciso observar os seguintes passos: a) Criar os novos marcadores. Para isso, abra o arquivo blankPatt.gif, contido na pasta “patterns” do ARToolKit, e adicione um símbolo na parte branca central, posicionado de forma assimétrica, conforme a Figura 11.1. Faça uma letra (H), por exemplo, imprima em seguida o marcador.

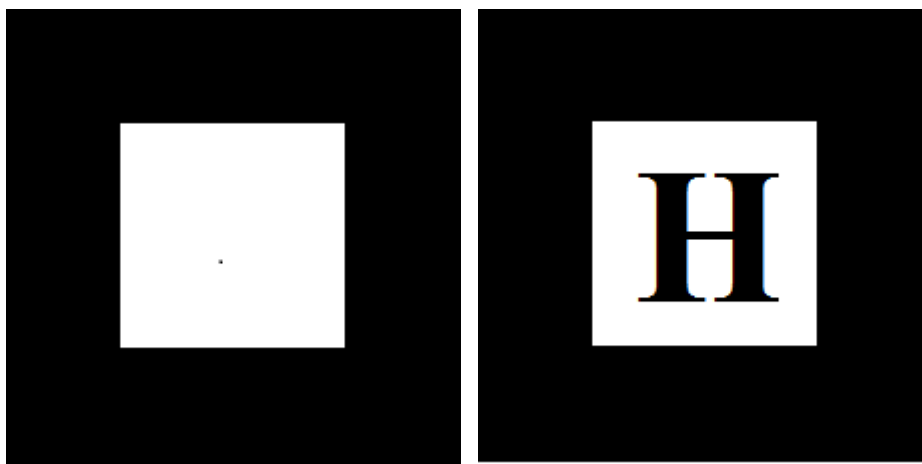


Figura 11.1 – Novo marcador a partir do blankPatt.gif.

b) Executar o programa "mk_patt.exe", contido na pasta "bin". Será pedido para que você entre com um nome de arquivo de parâmetros de câmera. Entre com o nome do arquivo "camera_para.dat", ou simplesmente tecle "*enter*". Este é o nome *default* para o arquivo de parâmetros de câmera. Aparecerá, então, a tela mostrada na Figura 11.2.

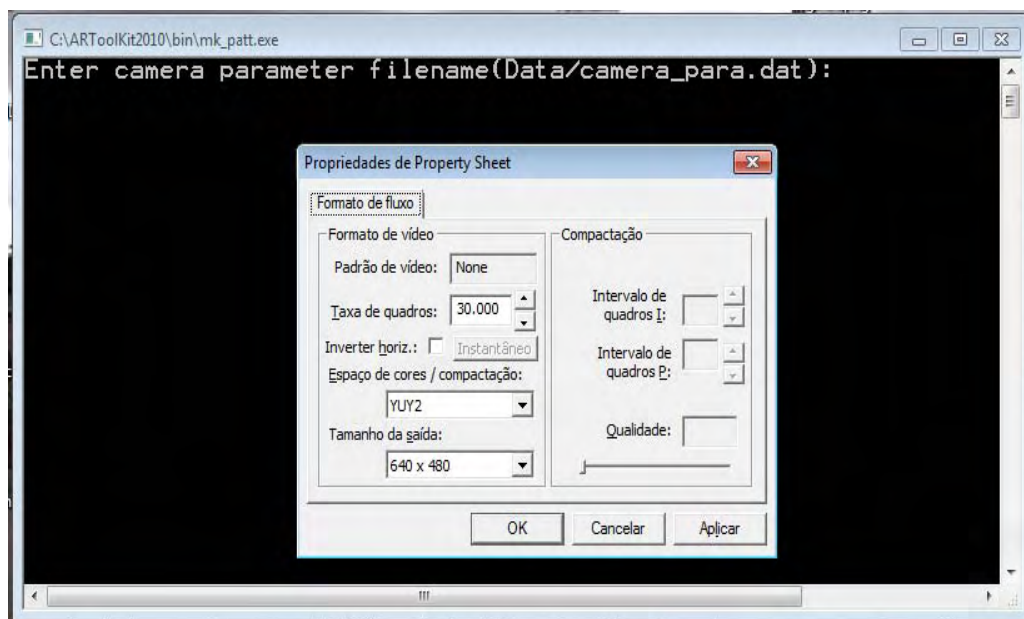


Figura 11.2 – Executando o programa de cadastro de marcadores.

c) Enquadrar a câmera de vídeo, apontando diretamente para o marcador. Surgirão, então, bordas vermelhas e verdes em torno do marcador. Isso indica que o software "mk_patt" encontrou o quadrado em torno do marcador que está sendo cadastrado. O marcador deve ser movimentado até que os lados vermelhos do quadrado estejam no topo e à esquerda do quadrado na imagem de vídeo (a figura 11.3 mostra uma situação intermediária, não é a situação final). Uma vez que o quadrado encontrado esteja orientado corretamente, clique no botão esquerdo do mouse. Será então pedido um nome de arquivo para o marcador (Por exemplo: marcador h).



Figura 11.3 – Cadastramento do marcador.

d) Copiar os novos arquivos de padrões de marcador "bin" para a pasta "Data", antes de usá-los.

APÊNDICE

1. TELA DO INÍCIO DO SOFTWARE

A seguir, apresentam-se as telas do software desenvolvido (Figura 12.1), (Figura 12.2).

Executar o arquivo rastreamento.jar para o início do rastreamento.

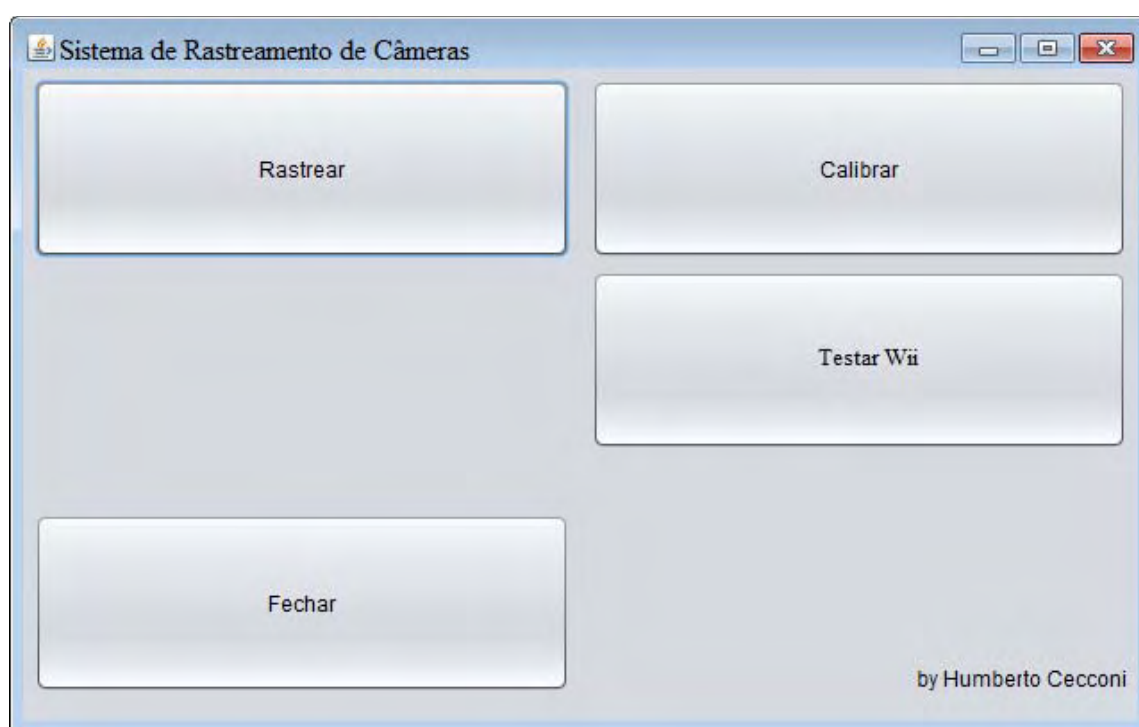


Figura 12.1 - Tela inicial do software.

Entrar com as coordenadas centrais em milímetros do multimarcador. Essas coordenadas foram anotadas quando o multimarcador foi posicionado no teto do estúdio.

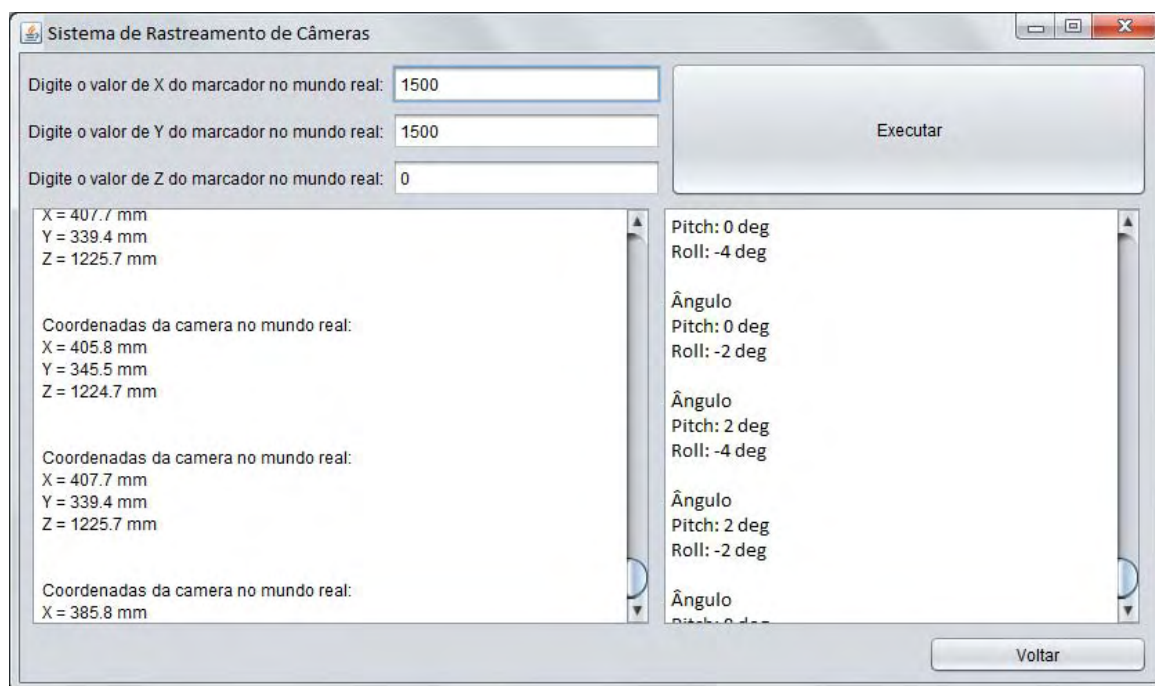


Figura 12.2 - Resultados obtidos através do rastreamento.

Na figura 12.2, são exibidos os resultados dos eixos X, Y e Z do rastreamento da câmera e os resultados do Wii Motion Plus, Pitch e Roll. Esses resultados são gravados em um arquivo txt para o posterior posicionamento da imagem virtual no cenário.