

UNIVERSIDADE ESTADUAL PAULISTA
“JÚLIO DE MESQUITA FILHO”
FACULDADE DE ARQUITETURA, ARTES E COMUNICAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM MÍDIA E TECNOLOGIA

IVAN ABDO AGUILAR

**ARSTUDIO 2.0: um sistema de estúdio virtual para geração de conteúdo midiático
baseado no motor de jogos Unity3D**

BAURU/SP

2017

Ivan Abdo Aguilar

**ARSTUDIO 2.0: um sistema de estúdio virtual para geração de conteúdo midiático
baseado no motor de jogos Unity3D**

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

BAURU/SP

2017

Aguilar, Ivan Abdo.

ARSTUDIO 2.0 : um sistema de estúdio virtual para
geração de conteúdo midiático baseado no motor de
jogos Unity3D / Ivan Abdo Aguilar, 2017
113 f.

Orientador: Antonio Carlos Sementille

Dissertação (Mestrado)- Universidade Estadual
Paulista. Faculdade de Arquitetura, Artes e
Comunicação, Bauru, 2017

1. ARSTUDIO 2. Estúdios virtuais 3. Sistemas de
pré-visualização 4. Realidade aumentada 5. Produção
audiovisual. I. Universidade Estadual Paulista.
Faculdade de Arquitetura, Artes e Comunicação. II.
Título.

IVAN ABDO AGUILAR

**ARSTUDIO 2.0: um sistema de estúdio virtual para geração de conteúdo midiático
baseado no motor de jogos Unity3D**

Área de Concentração: Ambientes Midiáticos e Tecnológicos

Linha de Pesquisa: Tecnologias Midiáticas

Banca Examinadora:

Prof. Adj. Antonio Carlos Sementille

Presidente / Orientador / Faculdade de Ciências – UNESP Bauru

Prof. Dr. Silvio Ricardo Rodrigues Sanches

Docente / Universidade Tecnológica Federal do Paraná – UTFPR

Prof^a. Dr^a. Leticia Passos Affini

Docente / Faculdade de Arquitetura, Artes e Comunicação – UNESP Bauru

Resultado: **APROVADO**

Bauru, 31 de maio de 2017.

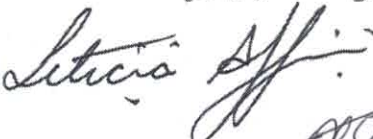
ATA DA DEFESA PÚBLICA DA DISSERTAÇÃO DE MESTRADO DE IVAN ABDO AGUILAR, DISCENTE DO PROGRAMA DE PÓS-GRADUAÇÃO EM MÍDIA E TECNOLOGIA, DA FACULDADE DE ARQUITETURA, ARTES E COMUNICAÇÃO - CÂMPUS DE BAURU.

Aos 31 dias do mês de maio do ano de 2017, às 09:00 horas, no(a) Auditório da Seção Técnica de Pós-graduação da Faculdade de Arquitetura, Artes e Comunicação da Unesp - câmpus de Bauru, reuniu-se a Comissão Examinadora da Defesa Pública, composta pelos seguintes membros: Prof. Adj. Dr. ANTONIO CARLOS SEMENTILLE - Orientador(a) do(a) Departamento de Computação / UNESP- Câmpus de Bauru, Profa. Dra. LETICIA PASSOS AFFINI do(a) Departamento de Comunicação Social / Faculdade de Arquitetura, Artes e Comunicação de Bauru, Prof. Dr. SILVIO RICARDO RODRIGUES SANCHES do(a) Departamento Acadêmico de Computação / Universidade Tecnológica Federal do Paraná, sob a presidência do primeiro, a fim de proceder a arguição pública da DISSERTAÇÃO DE MESTRADO de IVAN ABDO AGUILAR, intitulada **ARSTUDIO 2.0: um sistema de estúdio virtual para geração de conteúdo midiático baseado no motor de jogos Unity3D**. Após a exposição, o discente foi arguido oralmente pelos membros da Comissão Examinadora, tendo recebido o conceito final: APROVADO. Nada mais havendo, foi lavrada a presente ata, que após lida e aprovada, foi assinada pelos membros da Comissão Examinadora.

Prof. Adj. Dr. ANTONIO CARLOS SEMENTILLE



Profa. Dra. LETICIA PASSOS AFFINI



Prof. Dr. SILVIO RICARDO RODRIGUES SANCHES



AGRADECIMENTO

Agradeço aos meus pais, Glaucy e Francisco, por todo apoio desde pequeno e ao meu irmão, Igor, por todo incentivo de pesquisa.

À minha noiva, Jessika, por todo incentivo, ajuda, paciência e companhia não só nesse trabalho, mas em todas as horas.

À UNESP pela oportunidade de executar a pesquisa, pela estrutura e conhecimentos preciosos que me foram oferecidos.

Ao meu orientador Antonio Carlos Sementille por toda ajuda, apoio, ensinamentos e amizade durante todos esses anos.

À todos do laboratório, Everton, Renata, Mayckel, Gustavo Avellar, Gustavo Dewes, Paola, Juliana, Marcelo, Cubas, Victor e Mariana, muito obrigado por toda ajuda.

AGUILAR, I. A. ARSTUDIO 2.0: um sistema de estúdio virtual para geração de conteúdo midiático baseado no motor de jogos Unity3D, **2017, 113 f. Trabalho de Conclusão (Mestrado em Mídia e Tecnologia) - FAAC - UNESP, sob a orientação do Prof. Dr. Antonio Carlos Sementille, Bauru, 2017.**

RESUMO

Tanto as produções de cinema quanto as de televisão e internet podem se beneficiar das técnicas de combinação de elementos virtuais (2D e 3D) e elementos reais fornecidos pela Realidade Aumentada e, portanto, a aplicação destas técnicas em sistemas de estúdio virtual tem se revelado uma abordagem bastante flexível e inovadora para a geração de conteúdo midiático. Na cadeia de produção tradicional, elementos virtuais geralmente são inseridos somente na fase de pós-produção, e, conseqüentemente, o conteúdo virtual só é visível depois que todo processo de edição é finalizado. No estúdio real a equipe de gravação e os atores são guiados por sinais visuais simples onde as personagens ou objetos virtuais devem aparecer. Com a utilização de técnicas de Realidade Aumentada é possível, em tempo real, que os estúdios virtuais permitam a inserção, interação e visualização de objetos virtuais durante todas as etapas da cadeia de produção, minimizando os erros de gravação e, conseqüentemente, reduzindo custos. Um grande problema na implementação de estúdios virtuais diz respeito à geração e interação com elementos virtuais 3D de alta definição e em tempo real. Tradicionalmente este problema é tratado utilizando-se bibliotecas gráficas. No entanto, outra possibilidade é a utilização dos recursos de motores de jogos. Considerando o contexto exposto, o objetivo principal do presente trabalho consistiu em elaborar, implementar e testar um estúdio virtual, de baixo custo, capaz de combinar as funções de realidade aumentada da biblioteca Vuforia e os recursos de computação gráfica fornecidos pelo motor de jogos Unity 3D, bem como as capacidades de captura de movimentos do Kinect v2 da Microsoft. Este estúdio virtual foi denominado ARSTUDIO versão 2.

Palavras-chave: ARSTUDIO, Estúdios Virtuais, Sistemas de Pré-visualização, Realidade Aumentada, Produção Audiovisual.

ABSTRACT

Film, television and internet productions can benefit from the techniques of combining virtual elements (2D and 3D) with real elements provided by the use of Augmented Reality and, therefore, the application of these techniques in virtual studio systems have proved to be a quite flexible and innovative approach to the generation of media content. In traditional production pipelines, virtual elements are usually inserted only during the post-production phase, and consequently, the virtual content is only visible after the editing process is finished. While filming in the real studio, film crew and actors are guided by simple visual signals that represent where the virtual characters and objects should appear. With the use of Augmented Reality techniques it is possible, in real-time, that virtual studios allow for the insertion, interaction and visualization of virtual objects during all stages of the production pipeline, minimizing the recording errors and, consequently, reducing costs. A major problem in the implementation of virtual studios is the generation and interaction with high-definition 3D virtual elements in real-time. Traditionally this problem is handled using graphics libraries. However, another possibility is the use of game engines. Considering the above context, the main objective of this work was to design, implement and test a, low cost, virtual studio capable of combining functions from Vuforia's augmented reality library and the computer graphics capabilities provided by the Unity 3D game engine, as well as the motion capture capabilities from the Kinect v2 from Microsoft. This virtual studio was named ARSTUDIO version 2.

Keywords: ARSTUDIO, Virtual Studios, Previsualization Systems, Augmented Reality, Audiovisual Production.

LISTA DE ILUSTRAÇÕES

Figura 1 - Cadeia de produção tradicional.....	14
Figura 2 – Marcadores fiduciais ARToolKit.....	18
Figura 3 – Reconhecimento de marcadores no ARToolKit	19
Figura 4 – Multi-marcadores fiduciais ARToolKit.....	19
Figura 5 - Cadeia de produção otimizada.....	21
Figura 6 - Etapas da composição de um sinal misturado: a) plano de fundo; b) primeiro plano; c) plano de fundo mais o primeiro plano; d) máscara do primeiro plano; e) máscara do plano de fundo; f) máscara do primeiro plano mais a máscara do plano de fundo	33
Figura 7 - Sistema de captura de câmera utilizado pelo Synthevision	33
Figura 8 - Arquitetura de sistema utilizado pelo Synthevision.....	34
Figura 9 - Esquema de funcionamento da técnica <i>camera ganging</i>	35
Figura 10 - Ambiente de pré-visualização utilizando o RTFX: captura de movimento em tempo real (esquerda), manipulação da iluminação (centro) e a composição da cena final no motor de jogos (direita)	36
Figura 11 - Suportes para a câmera RGB com o Kinect v2.....	37
Figura 12 - Resultado do sistema de pré-visualização no Unity	38
Figura 13 - Ajuste automático de iluminação e cor	39
Figura 14 - Captura do esqueleto do ator e sua interação com uma parede virtual	40
Figura 15 - Visão do usuário, através do Oculus Rift, ao manipular um menu usando o Leap	41
Figura 16 – Módulos do ARSTUDIO	45
Figura 17 – Hierarquia de bibliotecas do ARSTUDIO 1.0	47
Figura 18 – Hierarquia de bibliotecas do ARSTUDIO 2.0	47
Figura 19 – Arquitetura modular do ARSTUDIO 2.0	49
Figura 20 – O ARSTUDIO 2.0 nas fases de Pré-Produção e Produção	52
Figura 21– O ARSTUDIO 2.0 na fase de Pós-Produção.....	53
Figura 22 – Grafo geral do ARSTUDIO 2.0	54
Figura 23 – Grafo de RA do ARSTUDIO 2.0	54
Figura 24 – Grafo de Áudio e Exportação do ARSTUDIO 2.0.....	57

Figura 25 – Grafo de captura de movimento do ARSTUDIO 2.0	58
Figura 26 - Mesa do operador	60
Figura 27 - Área de gravação	60
Figura 28 – Equipamentos Utilizados	63
Figura 29 - Aplicação da técnica de <i>chroma-key</i> e a inserção de um fundo novo.....	64
Figura 30 - Aplicação da técnica de <i>chroma-key</i> : a) cena não processada; b) primeira aplicação do <i>chroma-key</i> ; c) segunda aplicação do <i>chroma-key</i>	65
Figura 31 – Experimento de vídeo proveniente da câmera real, inserção de objetos virtuais e a aplicação da técnica de <i>chroma-key</i> : a) editor do Unity sem o <i>chroma-key</i> ; b) vídeo aumentado sem o <i>chroma-key</i> ; c) editor do Unity com o <i>chroma-key</i> ; d) vídeo aumentado com o <i>chroma-key</i>	66
Figura 32 - Translação da câmera virtual com a técnica de <i>chroma-key</i>	67
Figura 33 – Experimento da mixagem de áudios da cena virtual com o áudio do microfone no estúdio.....	68
Figura 34 - Experimento da inserção de um objeto virtual com referência a marcadores fiduciais e de textura e a comparação de robustez entre eles	69
Figura 35 – Comparação da quantidade de pontos de referência entre dois marcadores	70
Figura 36 – Erro de registro de profundidade entre elementos reais e virtuais	70
Figura 37 - Experimento da realidade aumentada com a técnica de renderização por camadas	72
Figura 38 – Configuração do controle como entrada no Unity	73
Figura 39 - Exemplo de incoerência na sobreposição de camadas devido à interferência do usuário	74
Figura 40 – Técnica de oclusão mútua	75
Figura 41 - Experimento da realidade aumentada com a oclusão mútua: a) ator em frente aos objetos virtuais; b) ator entre os objetos virtuais; c) ator atrás dos objetos virtuais.....	76
Figura 42 – Interface de usuário para fixar o marcador	77
Figura 43 – Experimento de fixação de marcador.....	77
Figura 44 – Fixação de marcador com a oclusão mútua	78
Figura 45 – Retirada do marcador ao substituir o plano de fundo	79
Figura 46 - Multi-marcador cúbico utilizado nos experimentos	80

Figura 47 - Experimento com um multi-marcador cúbico, translação de um objeto virtual em cena.....	81
Figura 48 - Experimento com um multi-marcador cúbico, rotação de um objeto virtual em cena.....	81
Figura 49 – Marcador cilíndrico utilizado nos experimentos.....	82
Figura 50 - Experimento com um marcador cilíndrico, translação e rotação de um objeto virtual.....	83
Figura 51 - Experimento de um marcador cilíndrico com a técnica de renderização por camadas e o objeto virtual atrás do corpo do ator	84
Figura 52 - Experimento de um marcador cilíndrico com a técnica de renderização por camadas e o objeto virtual atrás da cabeça do ator.....	85
Figura 53 – Movimentação e rotação de um objeto virtual com origem alterada, em relação um marcador cilíndrico.....	86
Figura 54 – Elementos virtuais à frente e atrás do ator, criando o efeito de que o mesmo está no interior do objeto virtual.....	86
Figura 55 - Utilização de botões virtuais na interação do ator, trocando entre objetos virtuais	87
Figura 56 – Marcador com delimitações de botões virtuais.....	87
Figura 57 - Experimento de botões virtuais transparentes na interação do ator com a troca entre objetos virtuais.....	88
Figura 58 – Kinect v2	88
Figura 59 – Juntas capturadas pelo Kinect v2	89
Figura 60 – Experimento de captura de movimentos e a transposição desses movimentos em um humanoide virtual.....	89
Figura 61 – Captura de movimentos e o ator contracenando com um humanoide virtual	90
Figura 62 – Imagem HDR gerada a partir da câmera omnidirecional.....	91
Figura 63 – Experimento de iluminação HDRI: a) cena sem luz virtual e sem iluminação HDRI; b) cena com luz virtual e sem iluminação HDRI; c) cena sem luz virtual mas com iluminação HDRI; d) cena com luz virtual e iluminação HDRI	92
Figura 64 – Visão do operador do sistema	93
Figura 65 – Visão do modo operador de câmera	94

Figura 66 – Experimento do modo diretor	94
Figura 67 – Visão do ator com o uso do retorno espalhado ao ator.....	95
Figura 68 – Transmissão ao Vivo pelo ARSTUDIO 2.0	96
Figura 69 – Resultados dos experimentos do ARSTUDIO 2.0 com uma webcam a 1080p ..	100
Figura 70 – Resultados dos experimentos do ARSTUDIO 2.0 com uma webcam a 720p	100
Figura 71 – Resultados dos experimentos do ARSTUDIO 2.0 com uma câmera profissional a 720p	101

SUMÁRIO

CAPÍTULO 1 – INTRODUÇÃO	12
1.1 JUSTIFICATIVA	13
1.2 OBJETIVOS	15
1.2.1 OBJETIVO GERAL	15
1.2.2 OBJETIVOS ESPECÍFICOS	15
1.3 ORGANIZAÇÃO DA DISSERTAÇÃO	15
CAPÍTULO 2 - REALIDADE AUMENTADA, ESTÚDIOS VIRTUAIS E MOTORES DE JOGOS	17
2.1 REALIDADE AUMENTADA	17
2.2 CADEIA DE PRODUÇÃO	20
2.3 ESTÚDIOS VIRTUAIS	23
2.3.1 O USO DA TÉCNICA DE <i>CHROMA-KEY</i> EM ESTÚDIOS VIRTUAIS	26
2.3.2 MOTOR DE JOGOS E ESTÚDIOS VIRTUAIS	28
CAPÍTULO 3 - ESTÚDIOS VIRTUAIS	32
3.1 CONSIDERAÇÕES INICIAIS	32
3.2 SYNTHVISION	33
3.3 AGRUPAMENTO DE CÂMERAS (<i>CAMERA GANGING</i>)	34
3.4 RTFX	35
3.5 PRÉ-VISUALIZAÇÃO EM TEMPO REAL UTILIZANDO UM MOTOR DE JOGOS	37
3.6 DREAMSPACE	40
3.7 COMPARAÇÃO ENTRE OS ESTÚDIOS VIRTUAIS LEVANTADOS	42
CAPÍTULO 4 – DESENVOLVIMENTO	44
4.1 SISTEMA ARSTUDIO VERSÃO 1	44
4.2 ESTRUTURAÇÃO DO ARSTUDIO VERSÃO 2	47
4.2.1 A ENGINE GRÁFICA UNITY E A BIBLIOTECA VUFORIA	48
4.2.2 ARQUITETURA DO ARSTUDIO VERSÃO 2	49
4.2.3 O ARSTUDIO 2.0 E A CADEIA DE PRODUÇÃO OTIMIZADA	51
4.2.4 GRAFOS DE CENA DO ARSTUDIO VERSÃO 2	53
CAPÍTULO 5 – EXPERIMENTOS E ANÁLISE DOS RESULTADOS	59

5.1 MATERIAIS UTILIZADOS	59
5.1.1 AMBIENTE FÍSICO	59
5.1.2 COMPONENTES DE HARDWARE E SOFTWARE	61
5.2 EXPERIMENTOS REALIZADOS	63
5.2.1 EXPERIMENTOS COM INSPEÇÃO VISUAL	63
5.2.2 EXPERIMENTOS OBJETIVOS	97
5.3 COMPARAÇÃO ENTRE ESTÚDIOS VIRTUAIS	102
CAPÍTULO 6 – CONCLUSÃO	105
REFERÊNCIAS	107

CAPÍTULO 1 – INTRODUÇÃO

Na cadeia de produção (*production pipeline*) tradicional utilizada na geração de conteúdo midiático, existem normalmente três fases principais para a maioria de produções audiovisuais cinematográficas e de televisão: pré-produção, produção e pós-produção. Cada uma dessas fases é responsável por aspectos distintos no processo de produção, exigindo equipes de profissionais especializados (HUGHES, 2012; MILLERSON; OWENS, 2009).

É crescente, nos últimos anos, a geração de conteúdos audiovisuais onde se tem uma mistura de atores reais (*live-action*¹), gravados na etapa de produção, com elementos virtuais (CGI²), e efeitos especiais (*Visual Effects* - VFX), normalmente inseridos na fase de pós-produção.

Na fase de produção desses conteúdos, o enquadramento de uma cena quando o objeto virtual é um ponto de interesse ou quando é necessária uma interação significativa ou contato visual com um ator virtual, são difíceis de serem realizados a não ser que exista um meio de visualizar os elementos virtuais em tempo real durante a gravação. Como uma das formas de redução de erros, causadas por essa falta de referência visual ou formas de interação com objetos virtuais durante a gravação, foram elaborados os chamados sistemas de estúdios virtuais, que podem utilizar-se de tecnologias como a Realidade Aumentada³ (RA) para a inserção, interação e visualização de objetos virtuais durante a gravação (GIBBS et al., 1998; THOMAS, 2006).

A finalidade do estúdio virtual é proporcionar a composição de imagens sintéticas (combinação de duas ou mais camadas de imagens) com o vídeo capturado da cena real em tempo real (OWEN et al., 2003; WRIGHT, 2010). Conhecido também como realidade virtual em terceira pessoa, a composição permite que um "sinal misturado" seja visualizado de modo que exista uma combinação entre cenários e objetos físicos com ambientes e elementos virtuais (GIBBS et al., 1998). O estúdio virtual se diferencia de software tradicional de composição e edição devido ao fato de que a perspectiva do cenário virtual é atualizada de forma sincronizada com os movimentos das câmeras do mundo real, onde é então realizada a composição da cena aumentada com ângulos de perspectivas e posições espaciais corretos do mundo real (OWEN et al., 2003).

¹ Conteúdo com atores reais, humanos ou animais.

² *Computer-generated imagery* ou imagem gerada por computador.

³ Uma técnica que suplementa a visão do usuário sobrepondo objetos virtuais, gerados por computador, sobre o mundo real, de forma a parecer que coexistem no mesmo espaço (AZUMA, 1997).

O projeto ARSTUDIO, desenvolvido no laboratório SACI (Sistemas Adaptativos e Computação Inteligente), campus da UNESP de Bauru, é um estúdio virtual que utiliza técnicas de RA onde o mundo real é acrescido com objetos virtuais em tempo real formando uma cena combinada. Este estúdio proporciona um ambiente de criação de conteúdo audiovisual de forma que haja a inserção e modificação de elementos virtuais e efeitos especiais, bem como a interação dos mesmos com atores reais em, tempo real, durante a gravação (CAMPOS et al., 2010; GASPARI et al., 2014; SEMENTILLE et al., 2014).

Considerando o contexto exposto, o presente trabalho focou-se no desenvolvimento e validação de um estúdio virtual, de baixo custo, visando o aperfeiçoamento do sistema ARSTUDIO com uma nova hierarquia de software, novos equipamentos, com uma nova base gráfica proporcionada pelo uso do motor de jogos⁴ (*game engine*) Unity3D⁵, e recursos de RA.

1.1 JUSTIFICATIVA

Na cadeia de produção tradicional, a fase de pré-produção é onde ocorre a elaboração do roteiro, planejamento do cenário a ser construído, planejamento das locações de gravação, escolha dos equipamentos a serem utilizados, recrutamento de atores e técnicos e a preparação e ensaio de todo processo de produção. A produção é a fase onde a gravação do conteúdo ocorre, o que geralmente exige: o transporte, montagem e desmontagem de cenários físicos; transporte das equipes e equipamentos de produção para gravações em locais específicos, a direção dos atores, operadores de câmera e técnicos de iluminação; a gravação de todas as cenas previstas no roteiro e programadas na fase anterior de pré-produção; bem como a regravação de cenas, quando necessário. A fase final, de pós-produção, é onde todos os materiais do programa (vídeo, áudio e gráficos) são compilados de forma que os trabalhos das fases anteriores se juntem para criar um conteúdo acabado (BARNWELL, 2008; BLONDÉ et al., 1996; HUGHES, 2012; MILLERSON; OWENS, 2009; SCHENK; LONG, 2011).

Um alto custo é exigido em virtude dessa montagem, desmontagem e armazenamento dos cenários e adereços de produção nas fases de pré-produção e produção,

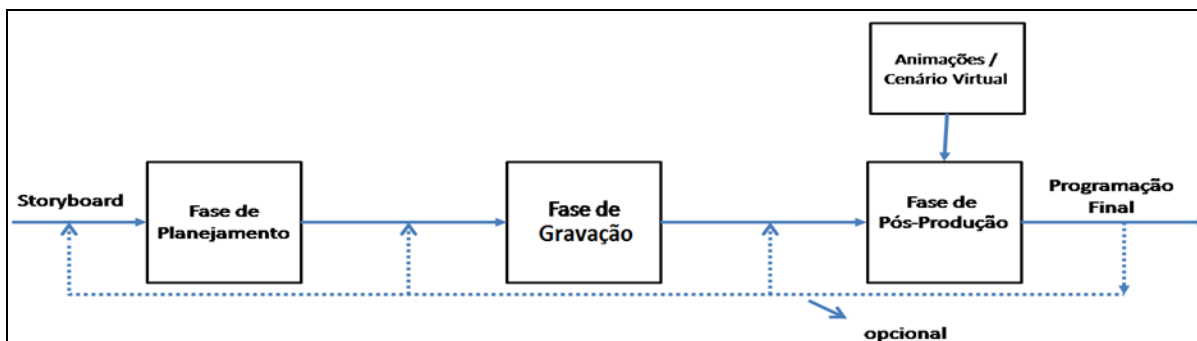
⁴ Um software, normalmente utilizado para a geração de jogos digitais, que contém, pelo menos, componentes como um sistema de renderização de gráficos 3D, um sistema de física e um sistema de áudio (GREGORY, 2009).

⁵ Um motor de jogos, desenvolvido pela empresa Unity Technologies, que tem suporte para uma gama de plataformas (GREGORY, 2009).

assim como a locação dos cenários e a logística do transporte da equipe e dos equipamentos necessários para a gravação no local (BLONDÉ et al., 1996).

A Figura 1 ilustra a cadeia de produção tradicional, onde os elementos tridimensionais e os efeitos especiais são utilizados somente na fase de pós-produção e, portanto, o conteúdo virtual misturado com o real é visível apenas após o término da gravação das cenas e a edição das mesmas. Em decorrência disso, o diretor, os atores e operadores de câmera são guiados por sinais visuais simples, como marcações ou objetos físicos no chão que correspondem aos locais onde as personagens, objetos virtuais ou efeitos especiais deverão aparecer. Devido a essa característica, diversos problemas podem ocorrer durante a fase de produção que impactarão a fase de pós-produção ou, até mesmo, toda a cadeia. São exemplos destes problemas: erro de enquadramento de câmera, erros no direcionamento dos olhares dos atores durante a atuação com elementos virtuais, e erro na sincronização espacial e/ou temporal durante interação bidirecional entre os atores e elementos virtuais. No pior dos casos, em decorrência desses ou outros problemas, a gravação da cena terá de ser refeita, como indicado nas linhas pontilhadas da Figura 1 (GRAU, 2005).

Figura 1 - Cadeia de produção tradicional



Fonte: GRAU, 2005

O sistema de estúdio virtual é uma ferramenta importante para as produtoras de conteúdo midiático, possibilitando oportunidades criativas bem como a redução de custos pela flexibilidade e diminuição de tempo de criação e troca entre cenários virtuais sem a necessidade de construir, montar, desmontar, transportar ou armazenar fisicamente o cenário real durante e entre as gravações. Ademais, o sistema propicia o compartilhamento do espaço e equipamentos do estúdio entre as gravações de outros programas (BLONDÉ et al., 1996).

A aplicação da RA nos sistemas de estúdios virtuais, durante a etapa de gravação, tem se revelado uma abordagem bastante flexível e inovadora para a geração de conteúdo,

podendo ser utilizada para a eliminação ou redução de erros durante a gravação e, consequentemente, a necessidade da regravação de cenas, otimizando a produção de conteúdo audiovisual e reduzindo os custos de produção (THOMAS, 2006). Considere-se, ainda, que o desenvolvimento de um estúdio virtual com recursos de RA que possa utilizar equipamentos convencionais (computador desktop, webcam, câmeras profissionais, etc.), sendo, portanto, de baixo custo, provavelmente será de interesse de pequenas produtoras, estações de televisão universitária, criadores domésticos de conteúdos online como os YouTubers e Vloggers, entre outros.

1.2 OBJETIVOS

1.2.1 OBJETIVO GERAL

O objetivo deste trabalho foi desenvolver um sistema de baixo custo, denominado ARSTUDIO 2.0, utilizando como base gráfica um motor de jogos, que permita a geração de cenas aumentadas compostas de elementos reais e virtuais, em tempo real. Esse sistema também permite a visualização e a interação com elementos virtuais durante a gravação. Para isso, foram utilizadas técnicas de RA, realidade virtual e computação gráfica.

1.2.2 OBJETIVOS ESPECÍFICOS

- Verificar a viabilidade do emprego de um motor de jogos (UNITY 3D) e equipamentos convencionais de baixo custo (tais como webcams e sensores de movimentos para jogos - Kinect v2 da Microsoft) como base de um sistema de estúdio virtual; e
- Verificar a viabilidade da utilização da Realidade Aumentada como forma de interação e visualização em um sistema de estúdio virtual.

1.3 ORGANIZAÇÃO DA DISSERTAÇÃO

Este documento está organizado nos seguintes capítulos:

No capítulo 2 são apresentadas as principais definições da técnica de Realidade Aumentada, a cadeia de produção, da técnica de *chroma-key*, de estúdios virtuais e de motores de jogos.

O capítulo 3 tem-se o levantamento e descrição dos principais estúdios virtuais encontrados na literatura e as características desses ambientes.

O capítulo 4 tem-se a descrição da arquitetura do ARSTUDIO versão 1, bem como a descrição detalhada de todo o desenvolvimento do ARSTUDIO versão 2.

No capítulo 5 são apresentados os detalhes referentes ao: ambiente experimental; resultados, dificuldades encontradas e soluções empregadas durante a implementação do protótipo ARSTUDIO 2.0 e uma breve comparação do mesmo com os estúdios virtuais encontrados na literatura.

Finalmente, no capítulo 6, têm-se as conclusões obtidas neste trabalho, além de propostas de trabalhos futuros.

CAPÍTULO 2 - REALIDADE AUMENTADA, ESTÚDIOS VIRTUAIS E MOTORES DE JOGOS

O presente capítulo apresenta os principais conceitos envolvidos neste trabalho, a saber: a tecnologia da Realidade Aumentada; a cadeia de produção tradicional e a otimizada por do meio da utilização de estúdios virtuais; os estúdios virtuais, a técnica de *chroma-key* e os motores de jogos.

2.1 REALIDADE AUMENTADA

A Realidade Virtual é uma tecnologia de imersão completa onde o usuário se encontra em um ambiente completamente virtual. Enquanto imerso nesse ambiente, o usuário não consegue visualizar o mundo real ao seu redor. Já a RA busca produzir um ambiente que integra perfeitamente o mundo real e o mundo virtual, onde os objetos aumentados são sobrepostos (compostos) as imagens provenientes do mundo real. A RA aumenta a realidade ao invés de completamente substituir a mesma por elementos virtuais. Essa adição de elementos virtuais ao mundo real também pode obstruir os elementos reais, retirando ou escondendo-os do usuário (AZUMA, 1997).

Segundo Azuma (1997), a RA é definida como um sistema que combina elementos reais com os virtuais, é sujeito a interações em tempo real e tem um alinhamento e sincronismo preciso entre os elementos virtuais tridimensionais (3D) com o ambiente real (registro).

A sobreposição do virtual com o real pode causar o um erro de registro, o deslocamento observado na imagem entre as posições pretendida e real de objetos virtuais (BAJURA; NEUMANN, 1995). Esse erro de registro pode ser causado pelo atraso entre a entrada do *frame* (quadro) do mundo real e o processamento desse *frame*, taxa de atualização do *frame*, iluminação do marcador, oclusão parcial ou total do marcador, escala do marcador, entre outros (AZUMA, 1997; BAILLOT; DAVIS; ROLLAND, 2001; ZHOU; DUH; BILLINGHURST, 2008).

Para minimizar os erros de registro, sistemas de marcador fiducial são utilizados, como o do ARToolKit (Figura 2). Esses sistemas consistem em padrões que são posicionados no ambiente real, são automaticamente detectados em imagens de câmeras digitais usando um algoritmo de detecção e possuem como finalidade oferecer suporte ao rastreamento, alinhamento e identificação. Em sistemas de RA, os marcadores fiduciais são geralmente

usados para rastrear elementos no ambiente onde a translação e rotação (pose) relativa entre uma câmera e um objeto são necessárias. Eles podem ser colocados de forma fixa no ambiente real para que a localização de uma câmera em movimento possa ser identificada ou eles possam ser colocados em objetos ou pessoas em movimento de modo que um local relativo a uma câmera fixa ou em movimento possa ser computado. Parâmetros importantes para tais sistemas de marcadores são a sua taxa de falsa detecção (taxa de falsos positivos), a sua taxa de confusão inter-marcador, tamanho de detecção mínimo (em pixels) e imunidade à variação de iluminação (KATO; BILLINGHURST; POUPYREV, 2000; OWEN; XIAO; MIDDLELIN, 2002; FIALA, 2005).

Figura 2 – Marcadores fiduciais ARToolKit



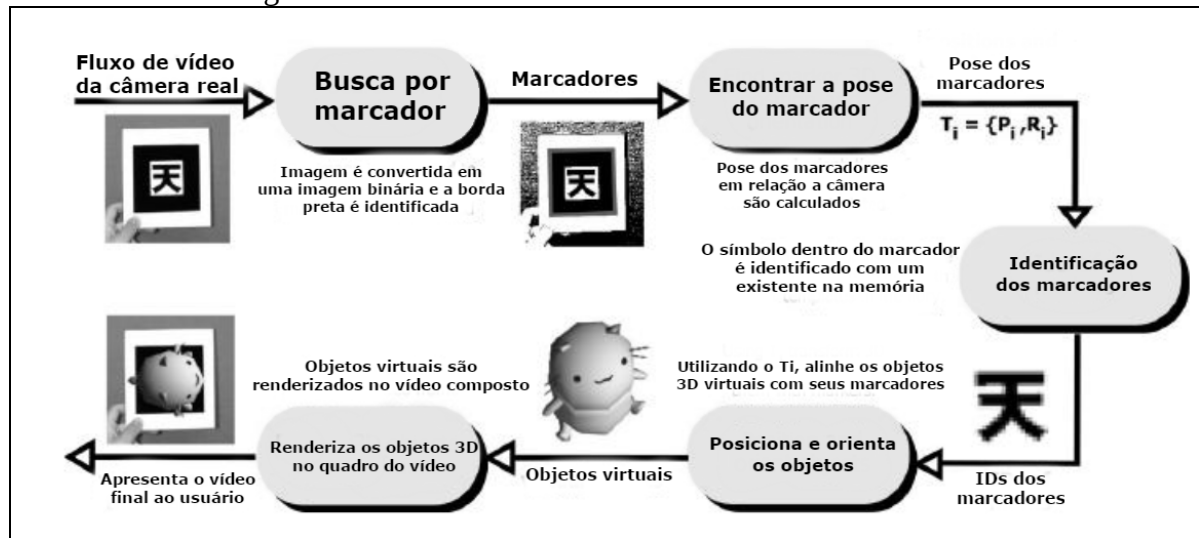
Fonte: KATO; BILLINGHURST; POUPYREV, 2000

O ARToolKit usa técnicas de visão computacional para calcular o ponto de vista da câmera real em relação a um marcador de mundo real (Figura 3). Primeiro a imagem de vídeo é transformada em uma imagem binária (preta ou branca). Nesta imagem, regiões quadradas são procuradas. Para cada quadrado, o padrão dentro do quadrado é capturado e comparado com alguns modelos de padrão previamente inseridos. Se houver uma correspondência, o ARToolKit encontrou um dos marcadores de rastreamento. O ARToolKit usa então o tamanho e a orientação de padrões conhecidos para calcular a posição da câmara de vídeo real em relação ao marcador físico. Feito isso o objeto virtual correspondente aquele marcador é renderizado^{6,7} sobre o mesmo (KATO; BILLINGHURST; POUPYREV, 2000).

⁶ Processo de conversão de conteúdos tridimensionais em imagens bidimensionais, com efeitos de fotorrealismo (DOBBINS, 2012).

⁷ Processo final de exportação de frames, em forma de vídeo ou uma sequência de imagens, de um filme a partir de uma composição. O frame gerado é uma imagem 2D composta por todas as camadas, configurações e outras informações pertinentes a imagem. A qualidade da composição e do formato a ser exportado afeta o tempo de exportação dos frames (tempo de renderização) (ADOBE AFTER EFFECTS, 2017).

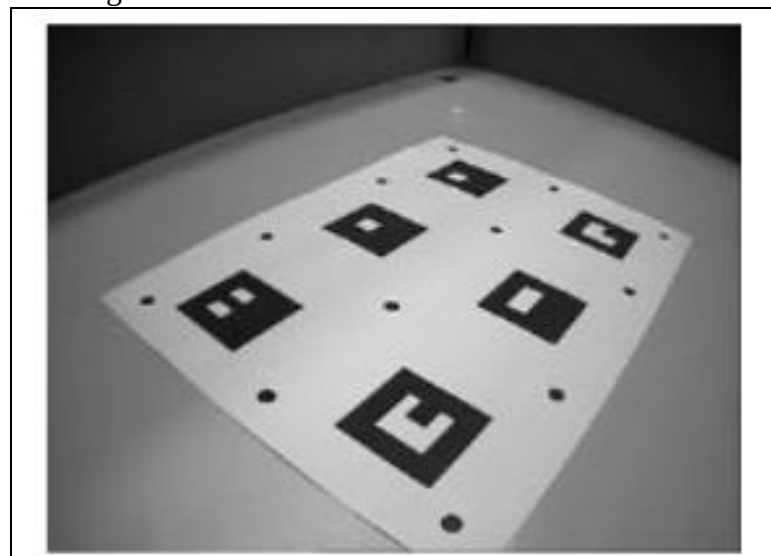
Figura 3 – Reconhecimento de marcadores no ARToolKit



Fonte: Adaptada de KATO; BILLINGHURST; POUPYREV, 2000

Para deixar a detecção de marcadores mais robustos foram criados os multi-marcadores, os quais proporcionam um rastreamento mais preciso, com relação aos marcadores planos, da estimativa da pose do marcador onde vários marcadores podem ser visíveis (Figura 4). O rastreamento de um multi-marcaador funciona pelo uso de diversos marcadores para formar um único marcaador maior. A estimativa da pose do multi-marcaador consiste em estimar as poses entre todos os marcadores que fazem parte do mesmo, o que gera um resultado mais estável e confiável. O multi-marcaador permite que sua detecção seja feita, mesmo que tenha partes obstruídas e que uma porção de seu conjunto de marcadores esteja fora do alcance da câmera (BARATOFF; NEUBECK; REGENBREHCT, 2002; SCHMALSTIEG; WAGNER, 2007).

Figura 4 – Multi-marcadores fiduciais ARToolKit



Fonte: KATO, et al. 2000

A realidade aumentada pode ser aplicada a diversas áreas como a Visualização de dutos em fábricas (NAVAB et al., 1999), Ambientes colaborativos (BUTZ et al., 1999), Arte (BILLINGHURST; GRASSET; LOOSER, 2005), Estúdios Virtuais (GASPARI et al., 2014), entre outros.

2.2 CADEIA DE PRODUÇÃO

Uma cadeia de produção permite organizar, dividir e processar as atividades necessárias para a geração de um projeto midiático em uma sequência de fases: cada fase realiza alguma tarefa e transfere o resultado para a próxima fase; a fase original pode, em seguida, começar a realizar sua tarefa nos próximos dados. Quando tal cadeia de produção é concebida adequadamente isto pode resultar na melhoria do rendimento do projeto (VAUGHAN, 2012; HUGHES et al., 2013).

Existem três fases principais que precisam ser executadas no processo de gravação para a maioria de produções audiovisuais convencionais, as fases de pré-produção, produção e pós-produção. Cada uma dessas etapas é responsável por aspectos distintos no processo de produção como um todo, tendo equipes específicas para cada etapa. Essas etapas são (HUGHES, 2012):

- *Pré-produção*: Fase dinâmica onde ocorre o planejamento, preparação e ensaio de todo processo de produção. Esta fase é onde ocorre: o financiamento do projeto; estipulação do orçamento necessário para cada fase; seleção das equipes necessárias; cronograma do projeto como um todo; a escrita do roteiro, especificando a lista de gravações necessárias e a criação de um cronograma para o mesmo; criação do *storyboard*; a escolha dos locais onde serão feitas as gravações; estudo com relação a iluminação necessária e ao som do ambiente, principalmente em locais abertos; quais equipamentos e equipes serão necessários durante cada gravação; e o planejamento das questões burocráticas como as documentações legais, acomodações de viagens, seguro de equipamentos, etc. Realizando uma pré-produção bem feita e planejada corretamente acarreta em uma minimização de tempo e custo durante as seguintes fases, principalmente na fase de produção;
- *Produção*: Onde a gravação do projeto em si ocorre. Nesta etapa é necessário haver a direção dos atores sobre suas atuações, realizar todas as gravações descritas e programadas na fase de pré-produção, bem como regravar as cenas quando for necessário. Esta fase pode causar diversas alterações no planejamento previamente feito, por exemplo: mudanças nos filtros das câmeras e aspectos de iluminação a serem usados nas gravações; alteração no posicionamento dos elementos do cenário e dos atores; modificação no roteiro, levando em consideração o que já foi gravado e

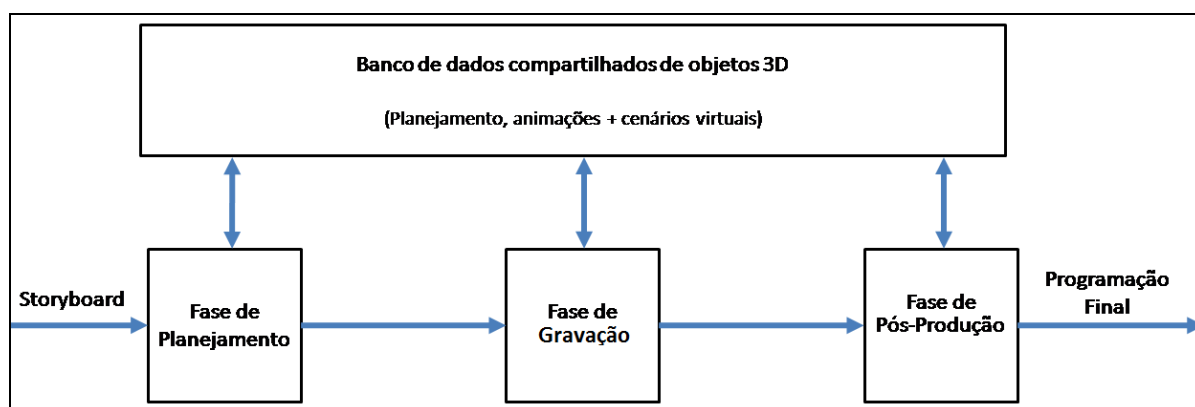
como esses novos aspectos no roteiro pode afetar outras cenas que serão ainda gravadas; alterações no cronograma devido ao clima; entre outras alterações;

- **Pós-produção:** Estágio final, depois que todos os materiais do programa (vídeo, áudio e gráficos) foram compilados, onde todo os trabalhos das fases anteriores se juntam. As funções normalmente realizadas nessa fase são: organização e documentação de todas as gravações de modo que elas terão uma coerência narrativa com o programa final; transferência das gravações para a equipe de pós-produção e efeitos especiais; aquisição e adição das narrações, músicas e efeitos sonoros a serem utilizados; substituição das falas dos atores, caso estejam com erros, qualidade ruim, com ruído, etc.; captura de cenários necessários, por exemplo, aqueles usados para o fundo de cenas; edição, combinação e composição de áudios e vídeos; correção de cores; adição de animações e efeitos de transição entre as gravações; adição de objetos virtuais; criação de montagens; tratamentos adicionais ao filme; prévias e revisões com o diretor e o cliente; e a criação das cópias finais para distribuição para os diversos meios, internet, cinema, DVD, etc.

O conteúdo 3D, nas produções convencionais, era inserido somente na fase de pós-produção e o conteúdo virtual era visível somente ao término das edições das gravações. No estúdio, os atores, os operadores de câmera e o diretor só tinham indicações visuais simples, como marcações no chão de onde as personagens ou objetos virtuais deveriam aparecer (GRAU, 2005).

De acordo com Grau (2005), as produções audiovisuais cinematográficas e de televisão que utilizam efeitos especiais com computação gráfica de alta qualidade normalmente seguem uma cadeia de produção 3D (cadeia de produção otimizada) com três fases que podem usar objetos 3D de um banco de dados compartilhado (Figura 5):

Figura 5 - Cadeia de produção otimizada



Fonte: Adaptada de GRAU, 2005.

- *Fase de Planejamento (Pré-produção)*: consiste de elementos 3D estáticos na cena. O diretor define a ação e o enquadramento de câmera para cada cena nessa fase. O posicionamento de atores podem ser demarcados por humanoides⁸ virtuais genéricos. Devido ao fato dos elementos 3D já estarem presente nesta fase, os animadores podem criar animações simples das animações propostas utilizando personagens virtuais. As animações podem ser rústicas, não precisam ser de qualidade final, uma vez que as mesmas podem ser alteradas e melhoradas durante as fases de gravação e pós-produção.
- *Fase de Gravação (On-Set)*: onde o banco de dados 3D (incluindo animações) da fase de planejamento são utilizados e alterados durante a gravação. Os modelos 3D, as animações e as imagens de fundo são utilizadas na pré-visualização. A pré-visualização consiste em uma visão prévia do vídeo aumentado renderizado em um monitor. Essa visão possibilita que o diretor decida nos posicionamentos e animações dos atores e dos adereços (reais e virtuais) em cena. Portanto, no percorrer desta fase os dados da fase de planejamento serão refinados, editados e armazenados para serem utilizados na fase de pós-produção.
- *Fase de Pós-Produção*: fase onde o conteúdo midiático é renderizado na qualidade final. Para isso, são realizados os ajustes finais dos modelos 3D e de suas animações, utilizando as cenas de gravação selecionadas. Dado que os animadores e modeladores podem utilizar o banco de dados compartilhado, que foi utilizado durante todas as fases da cadeia de produção, o trabalho será mais eficiente do que começar do zero, como no caso da cadeia de produção convencional.

A produção de um filme, geralmente, inicia-se com a definição de um roteiro. O diálogo das personagens e toda a ação são descritos como se estivessem acontecendo no presente momento. O roteiro apenas descreve o que o espectador irá ver e ouvir. A partir dele o diretor realiza a decupagem (*decouper*) do roteiro, que consiste na concepção de um roteiro técnico que contém a sequência das cenas a serem gravadas, especificação do uso e posicionamento de câmeras, iluminação, efeitos audiovisuais, etc. e serve como base para calcular a viabilidade e os custos da produção do conteúdo. A partir desse roteiro técnico o planejamento visual pode começar, com a elaboração do *storyboard*. *Storyboarding* não é apenas uma tradução do roteiro, o plano verbal, em uma série de imagens visuais, é a reescrita da história agora usando o fluxo de imagens para mostrar, ao invés de palavras, o que será apresentado visualmente. Hitchcock sabia disso e há rumores dele ter dito uma vez que

⁸ Uma máquina ou criatura com a aparência ou qualidades de um humano (CAMBRIDGE DICTIONARY, 2017).

quando o *storyboard* estivesse concluído, 95% do filme estava pronto e o resto era, simplesmente a execução (COMPARATO, 2000; GLEBAS, 2009).

A localização da câmera é tão importante para a cena quanto à posição de personagens. No filme, este arranjo físico dos atores em relação à câmera também é conhecido como *blocking* (planejamento da coreografia). O *blocking*, um termo de gravação utilizado cinematograficamente, é o movimento coreografado dos atores e/ou câmeras. A diferença é que o *blocking* usa esses elementos para adicionar interesse e dividir a cena gravada. Feito a encenação das personagens, técnicas de *blocking* podem ser utilizadas para criar uma cena mais criativa, como uma personagem obscura andando na frente da câmera. Um bom *blocking* e encenação (*staging*) consiste de alguns elementos como os pontos de interesse, pontos de vista do espectador ou das personagens e linhas de ação, tais como ângulos e posicionamento da câmera. Com o advento da computação gráfica, há a possibilidade de utilizar esses métodos nas imagens geradas por computador (GLEBAS, 2009).

2.3 ESTÚDIOS VIRTUAIS

Os avanços no hardware e software de computadores, em especial nos processadores e nas placas gráficas para a renderização, têm afetado o modo de produção de conteúdo audiovisual para a TV e o Cinema. As tecnologias utilizadas pelas produtoras têm avançado grandemente, principalmente nos aspectos de gravação e pós-produção, diminuindo os tamanhos dos equipamentos e assim tornando-os mais flexíveis e portáteis (BLONDÉ et al., 1996).

Na cadeia de produção tradicional, a inserção de elementos virtuais ocorre após as gravações com os atores, na fase de pós-produção. Em consequência dessa inserção após as gravações, erros podem ocorrer (interação bidirecional entre o real e o virtual, enquadramento, entre outros), impactando a produção do conteúdo midiático (GRAU, 2005).

A fase de pós-produção é a fase da cadeia onde se tem altos custos, em algumas produções de cinema os custos da pós-produção pode se igualar aos demais custos de produção. Por causa dos orçamentos baixos nas produções de TV, a pós-produção é restringida ao mínimo e a tendência ainda é a produção de conteúdo ao vivo (HELZLE et al., 2015).

Para auxiliar na produção desses conteúdos com elementos virtuais, surgem os sistemas de estúdios virtuais como alternativas mais flexíveis para a produção de conteúdo

mediático do que os estúdios convencionais, e têm sido recentemente utilizados na produção de vídeos para a televisão e o cinema. Esses sistemas possibilitam não só a substituição completa do cenário real pelo cenário virtual, mas também uma abordagem mais prática de inserção de elementos virtuais ao ambiente real. Proporcionando assim uma forma de gerar e visualizar, em tempo real, vídeos aumentados de imagens reais com elementos virtuais 3D, sincronizados com o movimento da câmera. Isso permite que elementos não facilmente reproduzidos no mundo real sejam gerados virtualmente e incorporados a gravação do cenário real (GÜNSEL; TEKALP; VAN BEEK, 1997; RAJA; MCKENNA; GONG, 1998; THOMAS, 2006).

Sistemas de estúdios virtuais contêm três componentes principais (RAHBAR; POURREZA, 2008):

- *Rastreamento de câmera*: gera um fluxo de dados descrevendo a posição e rotação da câmera, geralmente chamado de processo de estimativa da pose da câmera;
- *Renderização em tempo real*: utiliza dados de rastreamento de câmera e gera uma imagem virtual de um estúdio de gravação (ou set de filmagem);
- *Mixagem de vídeo*: combina o fluxo de vídeo proveniente da câmera do estúdio (câmera real) com o vídeo do software de renderização em tempo real, a fim de produzir o vídeo final combinado (vídeo aumentado) de pré-visualização.

Por causa das técnicas utilizadas no desenvolvimento de estúdios virtuais serem voltadas para o uso em tempo real, esses estúdios foram utilizados também na produção virtual de filmes. Esses estúdios eram utilizados como sistemas de pré-visualização durante a fase de produção onde atores reais precisavam interagir com atores e cenários virtuais (HELZLE et al., 2015).

Nos estúdio virtual, as câmeras reais e a virtuais estão permanentemente interligadas. Para alcançar isto, os parâmetros da câmara real necessitam ser determinados. Estes parâmetros são: as coordenadas (x, y, z) de posicionamento da câmera no mundo real; dados sobre inclinações, rotações no eixo fixo sobre o plano horizontal, e, possivelmente, movimento efetuado com a câmera em torno do eixo de suas lentes; e a distância focal e ajuste focal da lente da câmera (RATTHALER, 1996).

Segundo Grau, Pullen e Thomas (2004) o controle em um sistema de estúdio virtual consiste de quatro componentes: o controle da câmera, o qual permite a configuração remota dos parâmetros da câmera, como foco, zoom e abertura. Isto é importante, pois uma ou

mais câmeras podem ser penduradas no teto e, portanto, não serão acessíveis; o controle de gravação, usado para iniciar e parar a gravação da cena; o controle de animação, capaz de começar animações pré-definidas em 3D ou controlá-las ao vivo; e o controle da pré-visualização 3D.

A pré-visualização (*previsualization* ou *previz*) é um processo colaborativo que gera versões preliminares de gravações ou sequências, predominantemente usando ferramentas de animação 3D e um ambiente virtual. Ela permite que cineastas possam explorar visualmente idéias criativas, planejar soluções técnicas, e comunicar uma visão compartilhada para uma produção de forma eficiente (WONG, 2012).

Segundo Wong (2012) Existem diversos subtipos de pré-visualizações como o Pitchvis, D-Vis, Technical Previs, On-Set Previs e Postvis.

- *Pitchvis*: ilustra o potencial de um projeto antes que ele foi totalmente financiado ou aprovado. Como parte do desenvolvimento, estas sequências são conceituais, para serem refinadas ou substituídas durante a pré-produção;
- *D-Vis (Visualização do Design)*: utiliza uma estrutura virtual na pré-produção que permite a colaboração inicial do projeto em profundidade entre os cineastas. Antes das gravações serem desenvolvidas, D-Vis proporciona um espaço preliminar, uma estrutura virtual precisa de design no âmbito do qual as necessidades de produção podem ser testadas, e os locais de gravação podem ser observados e selecionados. Artifícios de design que são aprovados são criados e disponibilizados para os outros processos de pré-visualização;
- *A Pré-visualização Técnica (Technical Previs)*: incorpora e gera informações precisas sobre a câmera, iluminação, o design e o layout de cena para ajudar a definir os requisitos de produção. Isso muitas vezes toma a forma de diagramas dimensionais que ilustram como gravações particulares podem ser realizadas, utilizando termos e medições do mundo real;
- *A Pré-visualização On-Set (On-Set Previs)*: cria em tempo real visualizações, no local de gravação, para ajudar o diretor, diretor de fotografia, supervisor de efeitos visuais, e a equipe a rapidamente avaliar as imagens capturadas. Isso inclui o uso de técnicas que podem sincronizar e compor a gravação ao vivo com elementos virtuais em 2D ou 3D para um retorno visual imediato de como seria aquela cena na fase de pós-produção;
- *Postvis*: combina elementos digitais e a gravação da produção para validar a seleção de metragem de filme, fornece marcadores de posição de enquadramento para o editorial e refina efeitos. Edições incorporando sequências de postvis são frequentemente mostradas para um público teste para obter opiniões, e aos produtores e fornecedores de efeitos visuais para o planejamento e orçamento.

O processo de geração de conteúdo utilizando um sistema de estúdio virtual começa com a utilização de uma câmera para gravar os atores em um ambiente controlado. Os atores são extraídos do vídeo pela utilização da técnica de *chroma-key*. Os atores isolados são então inseridos em um novo ambiente virtual. Para que haja interação dos atores com o ambiente virtual, elementos virtuais são integrados a cena utilizando técnicas de RA (GRAU; PRICE; THOMAS, 2002).

Segundo Millerson e Owens (2009), o custo inicial da criação de um sistema de estúdio virtual possa ser bastante significativo, mas as economias de não ter que alterar fisicamente entre muitos tipos de cenas diferentes pode compensar pelo preço, a longo prazo.

As produções com estúdios virtuais também apresentam uma independência maior aos aspectos físicos, por exemplo, a necessidade de esperar a iluminação correta para a gravação de cenas ao ar livre, pois o plano de fundo e a iluminação podem ser modificados em uma fase de pós-produção. Condições específicas de clima ou limitações espaciais, como a necessidade de bloquear ruas em uma cidade, podem ser contornadas já que o local da cena pode ser reconstruído digitalmente ou gravado de forma separada e posteriormente composto com a cena real. O uso dessas produções virtuais também podem ser utilizadas para a diminuição do tempo de produção visto que diversos conteúdos compostos em tempo real podem ser utilizados na edição final, sem a necessidade de passar por um tratamento de pós-produção (HELZLE et al., 2015).

2.3.1 O USO DA TÉCNICA DE *CHROMA-KEY* EM ESTÚDIOS VIRTUAIS

A técnica de *chroma-key* (*chroma-keying*) é uma das técnicas mais utilizadas nas abordagens de *matting* digital em produções televisivas e cinematográficas. O *matting* digital é o processo de composição de imagens digitais, podendo ser dividido em duas fases. Na primeira fase (segmentação binária) é quando cada pixel da imagem/*frame* de vídeo é avaliado e decidido se pertence totalmente ao primeiro plano ou ao plano de fundo. Nesse processo é necessário que um elemento do primeiro plano seja extraído de uma imagem de fundo, estimando uma opacidade (alfa) para cada pixel. A imagem da opacidade, tomada como um todo, é referida como a *matte alfa* ou *key*. Opacidades fracionárias (entre 0 e 1) são importantes para a transparência e borramentos causados pelo movimento do elemento no primeiro plano, bem como para a cobertura parcial de um pixel do plano de fundo em torno da borda do elemento de primeiro plano. A segunda fase é a combinação desses elementos em primeiro plano com um novo plano de fundo (uma nova cena). *Matting* de vídeo (*video*

matting) é uma operação importante na televisão e na produção de filmes, dando ao diretor o poder de inserir novos elementos em uma cena ou transportar um ator a um novo local. (VAN DEN BERGH; LALIOTI, 1999; CHUANG et al., 2001; CHUANG et al., 2002; WANG; COHEN, 2007).

A técnica de *chroma-key*, portanto, consiste na identificação de partes da imagem em que se encontra uma determinada cor (*keying color* ou cor chave) com a finalidade de separar o plano de fundo dos elementos de interesse. Para utilizar o *chroma-key*, a gravação é feita com um plano de fundo de cor uniforme e os atores à frente do mesmo, no primeiro plano. O ator não deve vestir roupas com a mesma cor do plano de fundo. Uma nova imagem é, então, gerada combinando-se os elementos de interesse com um novo plano de fundo. Um uso clássico da técnica de *chroma-key* são as previsões meteorológicas na televisão (VAN DEN BERGH; LALIOTI, 1999; THOMAS, 2006).

Normalmente escolhem-se as cores azul ou verde como a cor chave pelo fato de não serem predominantes na pigmentação da pele humana, causando assim menos interferência no processo de *chroma-key* com os atores em cena. A cor verde, porém, apresenta um fator de preferência sobre a cor azul: os fotosensores dos dispositivos de captura de imagens (câmeras e filmadoras) possui uma predominância de filtros da cor verde para simular a fisiologia do olho humano, que é mais sensível à luz verde (HAILEY, 2002).

Segundo Grau, Pullen e Thomas (2004), um estúdio virtual necessita que o estúdio real possua um fundo e o piso colorido, como usado na técnica de *chroma-key*, de modo que os elementos de interesse da cena real possam ser extraídos e transpostos para a cena virtual. A fim de gerar um sinal bom sem ruído, é necessário ter o fundo colorido uniformemente claro e iluminado. Isto dá origem a vários problemas, principalmente quando utilizados em grandes estúdios:

- Proporcionar uma iluminação uniforme e brilhante sobre uma grande área;
- Iluminar os atores para dar o efeito artístico desejado, uma vez que a iluminação é determinada principalmente pelas exigências técnicas. Cenas com baixos níveis de iluminação apresentam problemas na extração do plano de fundo;
- Áreas onde sombras escuras não podem ser evitadas, como debaixo das mesas, podem causar problemas;
- Luz colorida espalhada pelo fundo e piso iluminados podem incidir sobre atores e adereços de cena, mudando sua tonalidade e dando uma aparência anormal. Essa incidência da cor sobre os atores é conhecida como "derrame" ou "*spill*".

Na televisão e no cinema, a técnica de *chroma-key* é usada extensivamente na troca do plano de fundo e na inserção de efeitos especiais. Esta técnica se tornou popular pelas vantagens de ser simples de compreender e produzir imagens com qualidade. Ela pode alcançar a implantação de imagem para um nível de precisão de poucos pixels. (IDDAN; YAHAV, 2001; MILLERSON; OWENS, 2009).

2.3.2 MOTOR DE JOGOS E ESTÚDIOS VIRTUAIS

Jogos digitais podem ser considerados como simulações interativas em tempo real. Na maioria dos jogos digitais um mundo é modelado matematicamente para que ele possa ser manipulado pelo computador. Esse modelo é uma simplificação da realidade (mesmo que imaginária). Portanto é uma simulação de um mundo real ou virtual. Os jogos também são simulações temporais, portanto, o mundo virtual é dinâmico - mudando com o decorrer da história e das ações das personagens. Um jogo necessita responder as ações de entrada do jogador, portanto é uma simulação interativa e temporal (GREGORY, 2009).

O termo motores de jogos (*game engine*) surgiu na década de 90, em referência a jogos de tiro em primeira pessoa (*First Person Shooter*) como o *Doom*. A arquitetura desse jogo foi desenvolvida de modo que os componentes principais (sistemas de renderização tridimensional, colisão, áudio, entre outros) estavam separados dos conteúdos de arte, fases, modelos, e as regras que controlavam a experiência do usuário. Essa separação ficou clara quando essa arquitetura foi utilizada para criar novos jogos com apenas a criação da parte artística, modelos, personagens e regras do jogo. No final da década de 90 jogos como o *Quake III Arena* e o *Unreal* foram desenhados para serem modificados e reutilizados, onde era possível também escrever código utilizando linguagens de programação como o *Quake C* e as licenças desses motores de jogos começaram a ser vendidas (GREGORY, 2009).

Um motor de jogo é um software extensível e pode ser utilizado como a base para muitos jogos digitais diferentes sem grandes modificações. Muitos motores de jogos são criados com o foco de desenvolvimenento em um gênero de jogo específico e uma plataforma (hardware) específica onde será executado esse jogo. Quanto mais abrangente for o alcance do motor de jogos em termos de gêneros de jogos e plataformas, menos otimizado esse sistema será para o gênero específico em uma plataforma específica (GREGORY, 2009).

Existem motores de jogos voltados para a criação de jogos em 2D e 3D. Alguns motores de jogos, comerciais, voltados para jogos em 3D são: Quake; Unreal Engine; Source;

Frostbite; CryENGINE; PhyreEngine; XNA Game Studio; e Unity. Alguns motores de jogos, comerciais, voltados para jogos em 2D são: Multimedia Fusion 2, Game Salad Creator, e Scracth. Existem muitos motores de jogos proprietários com acesso limitado a empresa criadora, como os motores de jogos da empresa Electronic Arts (EA) e a empresa Naughty Dog. Existem também motores de jogos de código aberto (open source) como os: Panda3D, Yake, Crystal Space, Torque e Irrlicht (GREGORY, 2009).

Segundo Gregory (2009), um motor de jogos normalmente consiste em uma arquitetura extensa com diversas camadas e interligações entre as camadas. Algumas dessas camadas são:

- *Hardware*: o hardware específico (sistema de computador ou console de jogo) onde o jogo executará. Algumas plataforma típicas incluem computadores baseados no Microsoft Windows, Linux e MacOS; plataformas móveis como o iPhone e o iPad da Apple, *smartphones* e *tablets* do Android, PlayStation Vita da Sony, entre outros; e consoles de jogos como o Xbox da Microsoft, PlayStation da Sony e os consoles da Nintendo (GameCube, Wii, Wii U, entre outros);
- *Drivers*: drivers de dispositivos com componentes de software de baixo nível fornecidos pelo sistema operacional ou hardware. Esses drivers auxiliam na comunicação entre o sistema e os hardwares disponíveis;
- *Sistema Operacional*: nos computadores, o jogo será apenas um de diversos programas em execução, onde o processador, a memória e o tempo serão compartilhados entre os diversos programas. No console, o sistema operacional pode ser um camada fina compilada junto com o código do jogo. No console, normalmente o jogo utiliza todo o processador, memória e tempo para somente ele;
- *SDKs externos*: muitos motores de jogos podem utilizar em baixo nível e/ou serem estendidos pela utilização de SDKs de terceiros pelo uso de APIs (application programming interface);
- *Renderização*: a renderização normalmente é realizada com uso de bibliotecas gráficas como a Glide, OpenGL, DirectX, libgcm, Edge. Componentes renderizados podem ser materiais e shaders, iluminação virtual, câmeras, textos e fontes, modelos geométricos primitivos, etc.;
- *Física e colisões*: o uso das leis de física e da detecção de colisões é disponível pelo uso de bibliotecas que possibilitam a interação com o mundo virtual;
- *Animações de personagens*: onde a leitura e a manipulação de animações (exportadas de programas de modelagem como o Maya e o 3DS Max) são realizadas;

- *Sistema Principal*: diversas utilidades como o gerenciamento de memória, biblioteca de matemática, estrutura de dados e algoritmos específicos, tocador de vídeo, entre outros;
- *Gerenciador de recursos*: uma interface dentro do sistema para que o usuário gerencie a manipule os arquivos de entrada no jogo (modelos 3D, texturas, imagens, materiais, fontes, parâmetros de física, esqueleto de personagens virtuais, mundo virtual, etc.);
- *Grafos de cena/Otimização de Oclusão*: controlam a estrutura dos elementos e os componentes dos elementos a serem renderizados na cena, levando em consideração a oclusão, frustrum da câmera, camadas de renderização, entre outros;
- *Efeitos Visuais*: efeitos disponíveis pelo sistema como o mapa de iluminação e sombras dinâmicas, iluminação HDR, sistemas de partículas (fumaça, fogo, água, etc.), efeitos de pós-produção (correção de cores, efeito de saturação de imagem, etc.), entre outros;
- *Interface de Usuário (Graphical User Interface - GUI)*: interface como menus e consoles que o usuário do jogo poderá utilizar para interagir como o jogo;
- *Dispositivos de entrada*: sistema que gerencia a entrada de dispositivos como o mouse, teclado, controles de videogames, etc;
- *Audio*: sistema que gerencia e executa os arquivos de áudio;
- *Rede*: sistema que gerencia a transferencia de dados entre clientes e servidores;
- *Sistema de códigos (scripts)*: um sistema que gerencia o código do sistema, atualizando e vinculando o código com o jogo conforme as modificações e o recarregamento dos códigos forem realizados.

Uma das tecnologias mais difíceis, e consequentemente mais caras, das produções com estúdios virtuais é a questão do tempo real, porque o seu desenvolvimento demanda tempo e conhecimento, e o mercado ainda é pequeno. Para o rastreio de movimentos de câmera em tempo real, existem apenas algumas tecnologias diferentes que possibilitam o uso em ambientes internos e externos e que são práticos o suficiente para serem instalados e prontos para uso dentro de uma a três horas. Uma alternativa a essa dificuldade tem sido o uso de motores de jogos. Os motores de jogos apresentam ferramentas flexíveis de tempo real e suas qualidades de renderização, recursos e capacidades estão em constante evolução (HELZLE et al., 2015).

Motores de jogos como ferramentas de renderização em tempo real estão começando a ser utilizados. Eles possuem ferramentas que processam em tempo real, são

flexíveis e apresentam uma capacidade e qualidade de renderização que estão constantemente sendo aprimoradas. Uma solução atualmente em desenvolvimento é o sistema denominado FilmEngine da Crytek. Esse sistema é baseado no motor de jogos Cryengine e oferece diversas funcionalidades aos cineastas (HELZLE et al., 2015; FILM ENGINE, 2017).

CAPÍTULO 3 - ESTÚDIOS VIRTUAIS

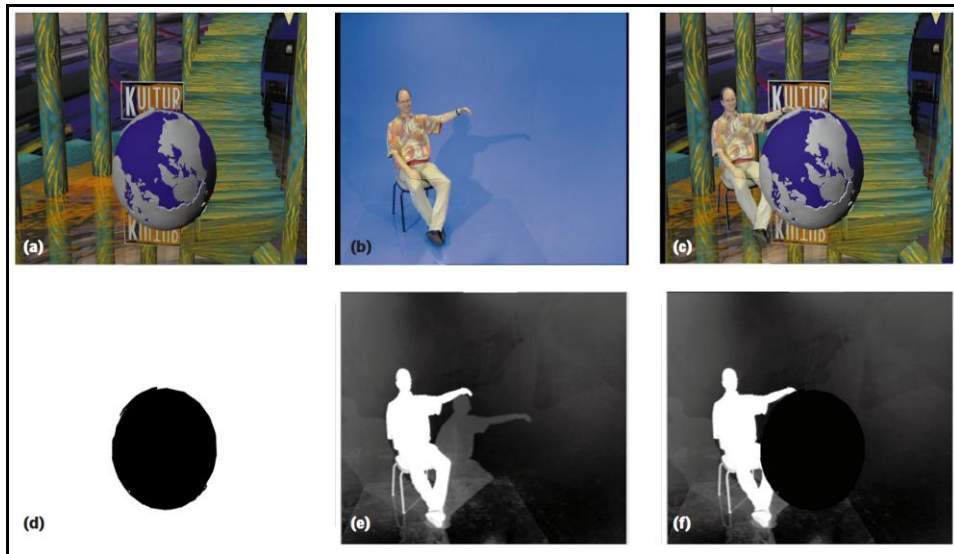
Neste capítulo serão apresentados os principais conceitos e técnicas associados aos estúdios virtuais e um levantamento dos principais estúdios virtuais encontrados na literatura.

3.1 CONSIDERAÇÕES INICIAIS

Historicamente os primeiros sistemas de estúdio virtual começaram como protótipos que estendiam a técnica tradicional de *chroma-key*. Um operador do *chroma-key* era responsável por selecionar a cor chave que seria utilizada. O hardware e software requeridos para realizar essa composição compreendem um sistema de estúdio virtual, enquanto a imagem combinada com o vídeo real constitui um cenário virtual (GIBBS et al., 1998).

A técnica que viabiliza a composição do real com o virtual, também conhecida por realidade virtual em terceira pessoa, possibilita a pré-visualização de um “sinal misturado” onde existe uma combinação de elementos reais e virtuais. Esses sistemas são divididos em três partes principais: rastreamento, renderização e composição. Para a produção de conteúdo, o set de gravação normalmente conta com: câmeras para capturar o mundo real e produzir um fluxo de vídeo de primeiro plano (*foreground*); um computador para gerar um plano de fundo (*background*) virtual correspondente e, opcionalmente, máscaras (filtros de camadas); e um software de estúdio virtual para realizar a composição destes planos (normalmente utilizando a técnica de *chroma-key*), em tempo real, onde o vídeo aumentado resultante é então enviado ao visor da câmera para auxiliar o operador de câmera no enquadramento, por exemplo. A Figura 6 ilustra estas etapas da composição de um sinal misturado (GIBBS et al., 1998).

Figura 6 - Etapas da composição de um sinal misturado: a) plano de fundo; b) primeiro plano; c) plano de fundo mais o primeiro plano; d) máscara do primeiro plano; e) máscara do plano de fundo; f) máscara do primeiro plano mais a máscara do plano de fundo

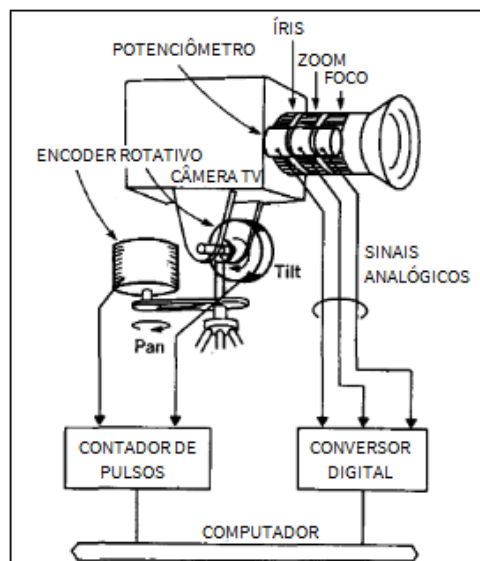


Fonte: GIBBS et al., 1998

3.2 SYNTHEVISION

Em 1989 a Synthevision, uma técnica de sintetização de imagem, foi apresentada por Shimoda, Hayashi e Kanatsugu (1989). Essa técnica requer que sensores analógicos fossem acoplados a câmera real, do primeiro plano, para produzir dados de *pan* (rotação na horizontal), *tilt* (rotação na vertical), *zoom* e foco, como apresentado na Figura 7 (SHIMODA; HAYASHI; KANATSUGU, 1989).

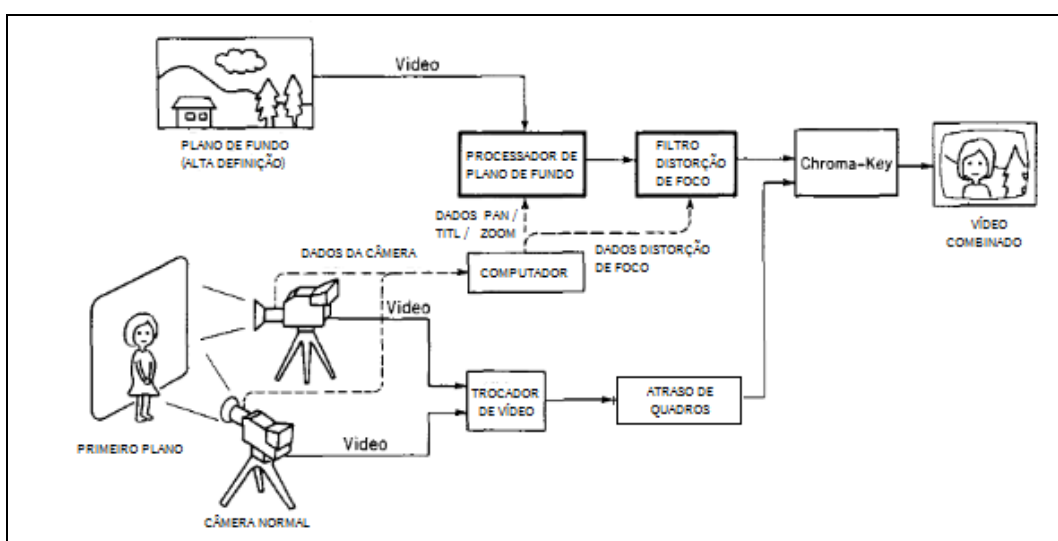
Figura 7 - Sistema de captura de câmera utilizado pelo Synthevision



Fonte: Adaptada de SHIMODA; HAYASHI; KANATSUGU, 1989

Com esses dados, as movimentações eram simuladas em uma câmera virtual, do plano de fundo, o que alterava a perspectiva em relação a esse plano de forma que existisse uma correspondência e sincronização entre as movimentações das câmeras dos dois planos (real e virtual). Era então realizada a composição do primeiro plano com o plano de fundo modificado, utilizando um *chroma-key* convencional. O plano de fundo consistia de um quadro de maior tamanho, o que possibilitava uma área maior para as rotações e o *zoom*. A arquitetura desse sistema é ilustrada pela Figura 8 (SHIMODA; HAYASHI; KANATSUGU, 1989).

Figura 8 - Arquitetura de sistema utilizado pelo Synthevision



Fonte: Adaptada de SHIMODA; HAYASHI; KANATSUGU, 1989

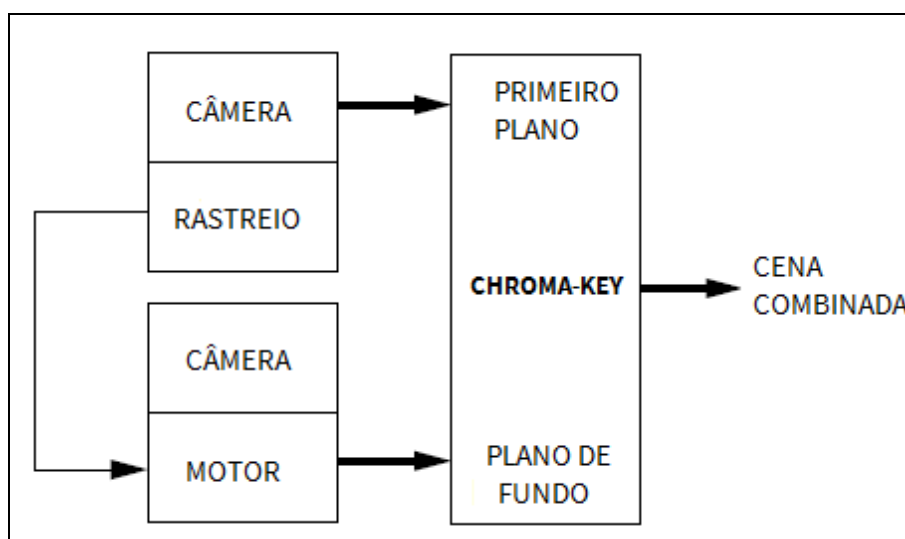
A criação do Synthevision trouxe algumas mudanças, não só no *chroma-key* convencional, mas também nos seguintes aspectos: uma alteração na cadeia de produção tradicional utilizada na criação de conteúdo, devido à possibilidade de composição da cena com elementos reais e virtuais em tempo real ou após a etapa de gravação; a utilização da computação gráfica para a geração de conteúdo a ser utilizado no plano de fundo; o armazenamento das imagens de plano de fundo em um banco de dados, viabilizando assim um esquema de troca efetivo do fundo durante a gravação (SHIMODA; HAYASHI; KANATSUGU, 1989).

3.3 AGRUPAMENTO DE CÂMERAS (CAMERA GANGING)

Uma alternativa a técnica utilizada pela Synthevision para que haja uma concordância entre as informações capturadas da câmera real com a virtual, é o uso da técnica

chamada de *camera ganging* (agrupamento de câmeras), onde duas câmeras reais são agrupadas mecânica e eletricamente no modo mestre e escravo. Normalmente a câmera do primeiro plano é o mestre e a do plano de fundo é o escravo, onde são duplicados os movimentos de câmera, ilustrado na Figura 9. Essa técnica é mais comum nas produções de efeitos especiais onde se tem mais de uma camada de vídeo ao vivo. Existem algumas formas de realizar a *camera ganging*: gravar os movimentos de uma câmera e utilizar esses dados para movimentar a segunda câmera; fisicamente interconectando duas câmeras de forma que a câmera escrava duplica os movimentos da câmera mestre, enquanto o operador de câmera está utilizando o mesmo, e; controlar câmeras por meio de um dispositivo externo (como um computador) que as direciona de forma simultânea ou em tempos diferentes através de uma sequência de movimentações (GIBBS et al., 1998).

Figura 9 - Esquema de funcionamento da técnica *camera ganging*



Fonte: Adaptada de GIBBS et al., 1998

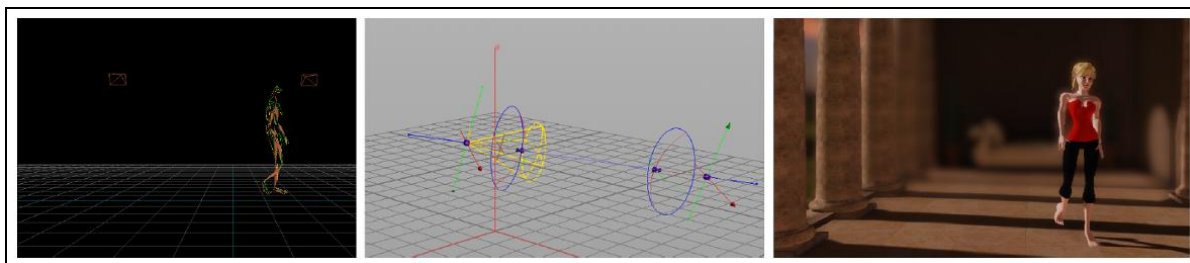
3.4 RTFX

A pré-visualização é uma facilidade que pode ser usada para diversas tarefas: planejamento e animação dos movimentos de personagens virtuais; criação e composição de efeitos especiais; e o design de espaços virtuais. Muitas ferramentas de pré-visualização tendem a se focar em poucas tarefas específicas que auxiliam em etapas essenciais da cadeia de produção. Na cadeia de produção, normalmente, são utilizadas diversos softwares para a: modelagem de elementos 3D, visualização e movimentação da câmera virtual, geração de efeitos especiais e sistemas de captura de movimento para capturar e manipular movimentos de personagens virtuais (NORTHAM; ISTEAD; KAPLAN, 2012).

O sistema RTFX (*Real-Time Special Effects* ou efeitos especiais em tempo real) apresenta os seguintes recursos: pré-visualização do ambiente e objetos virtuais, bem como efeitos especiais e captura de movimentos dos atores, com retorno para o diretor; captura de movimentos distribuídos, podendo ter capturas realizadas em no mesmo local e/ou em locais distintos e as informações das capturas integradas em tempo real ao cenário virtual, o que retira a limitação de quantos atores um sistema único de captura pode apresentar; visualização distribuída, dando ao diretor a possibilidade de alterar a visão sobre o cenário virtual com diversas câmeras virtuais, incluindo câmeras calibradas com equipamentos reais; e o controle de esqueletos virtuais de forma distribuída, o que possibilita que modelos virtuais podem ser animados por uma mistura de captura de movimentos em tempo real e/ou através de animações previamente programadas. Por exemplo, partes distintas de um objeto podem ser manipuladas por meio da captura de dois atores diferentes e uma terceira parte do objeto pode ser controlado por um programa em tempo real ou por movimentações pré-definidas (NORTHAM; ISTEAD; KAPLAN, 2011).

O ambiente RTFX utiliza uma comunicação de dados, através de *plugins*, entre um servidor central e clientes, cujos clientes são programas gráficos ou aplicações de captura, que trabalham juntos para produzir uma única cena de pré-visualização. Desse modo, diversos usuários podem trabalhar em partes distintas para aumentar uma visão final da cena. A Figura 10 apresenta um cenário de pré-visualização pelo RTFX (NORTHAM; ISTEAD; KAPLAN, 2012).

Figura 10 - Ambiente de pré-visualização utilizando o RTFX: captura de movimento em tempo real (esquerda), manipulação da iluminação (centro) e a composição da cena final no motor de jogos (direita)



Fonte: NORTHAM; ISTEAD; KAPLAN, 2012

Nesse caso o ambiente utilizou um sistema de captura de movimentos (com atores capturados em tempo real e com informações previamente capturadas e armazenadas), um

sistema para a animação, os motores de jogos Unreal Engine⁹ (UDK) e o Unity para a visualização em tempo real (NORTHAM; ISTEAD; KAPLAN, 2012).

3.5 PRÉ-VISUALIZAÇÃO EM TEMPO REAL UTILIZANDO UM MOTOR DE JOGOS

O sistema de pré-visualização no Unity, criado pelo Goussencourt e Bertolino (2015), é dividido em dois módulos principais. No primeiro módulo, off-line (etapa de preparo antes da utilização efetiva do sistema), ocorre a configuração da câmera RGB e do sensor que captura a profundidade, o Kinect v2¹⁰, para captura da profundidade da cena. A Figura 11 apresenta dois suportes para a câmera RGB com o Kinect v2 (GOUSSENCOURT; BERTOLINO, 2015).

Figura 11 - Suportes para a câmera RGB com o Kinect v2



Fonte: GOUSSENCOURT; BERTOLINO, 2015

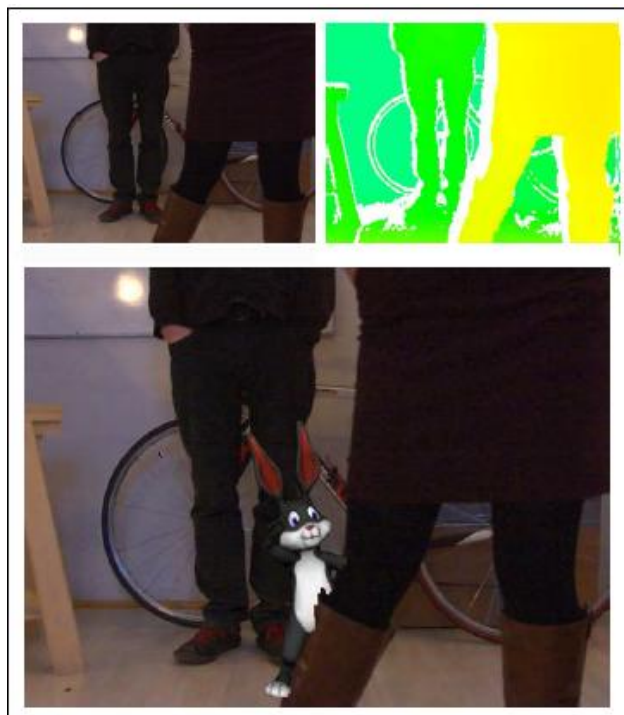
O segundo módulo, online (realizado em tempo real durante a execução do sistema), é onde são renderizadas, no Unity, as informações capturadas pelas câmeras e é onde a composição da cena, levando em consideração a posição e a profundidade dos objetos reais e virtuais, é então realizada. Nesse sistema, também é feito um rastreamento da câmera, o que permitiu com que seja possível realizar a segmentação do primeiro plano e plano de fundo sem o uso do *chroma-key* e a manipulação da câmera real com uma concordância na movimentação da câmera virtual. A Figura 12 mostra o resultado desse sistema, onde a

⁹ Um motor de jogos, desenvolvido pela empresa Epic Games (GREGORY, 2009).

¹⁰ Um sensor que captura a profundidade do ambiente real e também é utilizado para realizar a captura de movimentos (LACHAT et al., 2015).

primeira fileira apresenta o vídeo proveniente da câmera RGB e a profundidade vinda do Kinect v2, respectivamente, enquanto a segunda fileira apresenta a cena final composta por um objeto virtual que está entre os atores reais (GOUSSENCOURT; BERTOLINO, 2015).

Figura 12 - Resultado do sistema de pré-visualização no Unity



Fonte: GOUSSENCOURT; BERTOLINO, 2015

Nesse sistema, o Unity é responsável pela: estruturação e controle da cena e seus elementos virtuais; composição e edição da cena misturando imagens reais com os modelos virtuais; iluminação e correção de cor (*color correction*) dos adereços e cenários virtuais de forma automática; interações entre os atores do mundo real e os elementos do mundo virtual; criação de dados a serem utilizados na etapa de pós-produção (GOUSSENCOURT; DELLAC; BERTOLINO, 2015).

A edição e composição da cena no Unity são feitas utilizando a interface de usuário, nativa do motor de jogos. Essa interface permite que os elementos da cena, bem como seus parâmetros, podem ser modificados em tempo real, pelo usuário manipulando o sistema. Algumas das modificações possíveis através da interface são: a inserção e remoção de objetos bidimensional ou tridimensional, inserção e alteração de iluminação, alteração do posicionamento e escala dos objetos, ativação e controle de animações, entre outros. Além disso, o sistema dispõe da possibilidade de adicionar controladores novos a interface através de códigos que o usuário pode criar, tornando assim a interface ainda mais adequada e

adaptada para cada projeto. Existe também a opção de desenvolver um executável independente, retirando assim o uso da interface de desenvolvimento do Unity e os recursos que nem sempre são necessários e que podem ser custosos computacionalmente, mas para isso existe a necessidade da criação de uma nova interface para que o usuário possa interagir e utilizar esse programa (GOUSSENCOURT; DELLAC; BERTOLINO, 2015).

Esse sistema conta com um ajuste automático da iluminação e também das cores do mundo virtual para que os elementos do mundo virtual fiquem mais realistas e parecem estar no mesmo espaço que a cena captura do mundo real. A Figura 13 ilustra a cena combinada antes e depois desse ajuste de iluminação e de cor (GOUSSENCOURT; DELLAC; BERTOLINO, 2015).

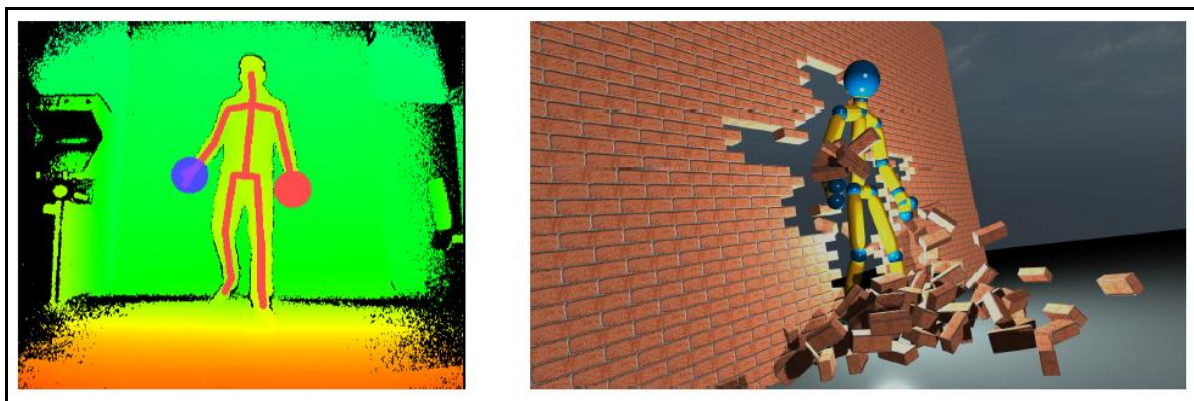
Figura 13 - Ajuste automático de iluminação e cor



Fonte: GOUSSENCOURT; DELLAC; BERTOLINO, 2015

O sistema também utiliza o sensor Kinect v2 para realizar a captura de movimento do ator em cena, dando assim a possibilidade de interação do ator real com o mundo virtual. Um esqueleto 3D virtual, o qual representa o corpo capturado do ator, é gerado na cena virtual. Quando esse esqueleto virtual colide com um objeto virtual um evento no Unity é acionado, como por exemplo o deslocamento de tijolos virtuais. Esse evento pode ser uma ação com base nas leis de física, já que o Unity possui bibliotecas de física nativamente. A Figura 14 mostra a captura do esqueleto do ator real e a sua interação com uma parede virtual (GOUSSENCOURT; DELLAC; BERTOLINO, 2015).

Figura 14 - Captura do esqueleto do ator e sua interação com uma parede virtual



Fonte: GOUSSENCOURT; DELLAC; BERTOLINO, 2015

A pré-visualização das informações não é a etapa final na cadeia de produção, então o sistema apresentado também armazena informações (profundidade, rastreo de câmera, entre outros dados) para que as mesmas possam ser utilizadas para auxiliar na etapa de pós-produção. Dessa forma as cenas podem ser reprocessadas a fim de gerar uma renderização final de maior qualidade e realismo. As informações são exportadas em um formato profissional utilizadas na indústria de pós-produção, para que possam ser utilizadas em programas dedicados a etapa de composição e edição (GOUSSENCOURT; DELLAC; BERTOLINO, 2015).

3.6 DREAMSPACE

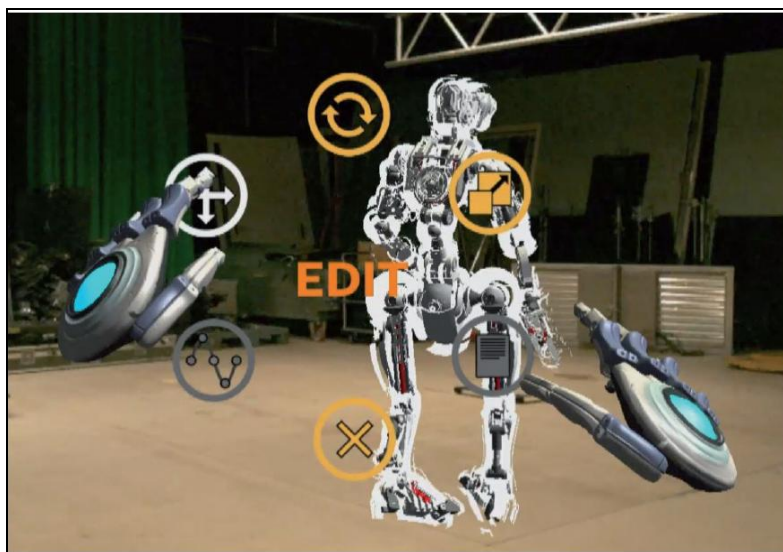
Foi desenvolvido um sistema (Dreamspace), por um grupo de universidades e empresas comerciais, de produção virtual, para a criação de conteúdo aumentado. Nesse sistema existe um aplicativo para *tablets*, tendo o Unity como base, para a: alteração da posição, rotação e escala de objetos; alteração da posição e rotação da câmera virtual e a edição dos parâmetros das luzes, como a cor, intensidade e posição, na cena virtual. Múltiplos *tablets* podem ser utilizados em paralelo para alterar a cena de forma simultânea e dinâmica. Um servidor de sincronização recebe essas alterações e distribui para os demais *tablets* (GRAU et al., 2016; TROTTNOW et al., 2015).

O servidor recebe o rastreamento da câmera real, em tempo real, através de um sistema de rastreo. Essas informações são enviadas para um sistema de visualização em tempo real. Esse sistema oferece a possibilidade de escolha do *rendering engines* (algoritmos de renderização) para a visualização, em alta qualidade, da cena editada em tempo real (TROTTNOW et al., 2015).

Também é utilizado um sistema de controle das luzes reais e virtuais, permitindo que caso ocorra uma alteração na iluminação do mundo virtual o sistema detectará essa alteração e alterará a iluminação no set de gravação de forma correspondente e vice-versa (TROTTNOW et al., 2015).

O Dreamspace conta com recursos de RA e reconhecimento de gestos com as mãos, através de: um Oculus Rift DK2, um HMD (*Head-Mounted Display* ou capacete de RA); com uma webcam acoplado (fornecendo uma visão de RA); e um Leap (Leap Motion Controller¹¹) para a interação com o sistema. Um sistema de captura foi utilizado para capturar o posicionamento do Oculus Rift enquanto o usuário utilizava o mesmo pelo set de gravação. Isso permitiu com que o usuário se movimentasse e observasse o set de forma aumentada. O Leap fica em frente ao corpo do usuário e serve para a manipulação (rotação, escala e posição) dos objetos virtuais na cena. O processamento é feito através de um computador executando o Unity. A Figura 15 apresenta a visão do usuário utilizando o Oculus Rift enquanto manipula um menu através do uso do Leap (TROTTNOW et al., 2015).

Figura 15 - Visão do usuário, através do Oculus Rift, ao manipular um menu usando o Leap



Fonte: TROTTNOW et al., 2015

Um sistema de captura de movimentos foi utilizado para capturar a atuação do ator em tempo real. A combinação do rastreamento do ator com o sistema de rastreamento da câmera possibilita a renderização de uma cena aumentada pelo ponto de vista da câmera principal e essa cena é enviada para os operadores de câmera em tempo real (TROTTNOW et al., 2015).

¹¹ Sensor que captura as mãos do usuário e possibilita a interação das mesmas com o mundo virtual (WEICHERT et al., 2013).

3.7 COMPARAÇÃO ENTRE OS ESTÚDIOS VIRTUAIS LEVANTADOS

Nas seções anteriores foram descritos quatro sistemas de estúdios virtuais com as suas principais características. O Quadro 1 sintetiza essas características.

Quadro 1 – Comparação dos principais estúdios virtuais levantados

	Synthevision	RTFX	Goussencourt & Bertolini (2015)	Dreamspace
Criadores	NHK (Japanese Broadcasting Corporation)	University of Waterloo	SolidAnim & Grenoble Alpes University	The Foundy, Ncam, Filmakademie Baden-Wurttemberg, Universitat Des Saarlandres, Intel, Crew, iMinds, Stargate Studios
Ano	1989	2011	2015	2015
Renderização	2D	3D	3D	3D
Uso da Realidade Aumentada	Não	Não	Não	Sim
Permite rastreamento de câmeras	Sim	Não	Sim	Sim
Retorno para o ator	Não	Sim	Sim	Sim
Mixagem de áudio dentro do sistema	Não	Não	Não	Não
Iluminação virtual HDRI	Não	Não	Sim	Não
Alteração da iluminação virtual de forma automática e correspondente com a real	Não	Não	Não	Sim
Alteração da iluminação real de forma automática e correspondente com a virtual	Não	Não	Não	Sim
Captura de movimentos para animação de personagens virtuais	Não	Sim	Sim	Sim
Pré-visualização da cena no estúdio virtual	Sim	Sim	Sim	Sim
Oclusão mútua por sensor de profundidade	Não	Não	Sim	Sim
Uso de motor de jogos	Não	Sim	Sim	Sim
Uso da rede para conectar com outros dispositivos	Não	Sim	Não	Sim
Interação do ator com objetos virtuais	Não	Não	Sim	Sim
Manipulação do sistema por meios além do computador principal	Não	Sim	Não	Sim
Sistema comercial	Sim	Não	Não	Não

Fonte: Elaborado pelo autor

Com base nas informações do Quadro 1 observou-se, principalmente, um crescimento no uso de: motores de jogos como base para o sistema; alteração de iluminação, tanto a virtual quanto a real, de forma automática; interação do ator com objetos virtuais; uso de sensores de profundidade e o uso da rede junto com o sistema.

CAPÍTULO 4 – DESENVOLVIMENTO

Neste capítulo são apresentados: o sistema do ARSTUDIO versão 1 bem como a estrutura e o desenvolvimento do ARSTUDIO versão 2.

Para o desenvolvimento deste trabalho, adotou-se a Prototipagem Evolutiva, a qual prevê um processo iterativo de geração de modelos de software (protótipos). À medida que o sistema é desenvolvido, diversas funcionalidades são adicionadas/alteradas no protótipo. Nesse sentido, a versão anterior do sistema, o ARSTUDIO versão 1.0, utilizava uma hierarquia de bibliotecas de software inteiramente diferente, a qual não se baseava em motores de jogos. O desenvolvimento da versão 2.0, portanto, exigiu uma completa reestruturação da arquitetura de software.

Para a realização desse trabalho, o seguinte planejamento foi seguido:

- Seleção, estruturação e implementação das características essenciais de um estúdio virtual em uma nova hierarquia, visando uma nova versão do ARSTUDIO;
- Execução de experimentos (inspeção visual e objetivos) realizados com o protótipo do ARSTUDIO 2.0;
- Análise dos resultados obtidos nos experimentos;
- Análise do sistema criado quanto a desempenho e viabilidade do uso de um motor de jogos como base de um sistema de estúdio virtual;
- Analisar o uso da Realidade Aumentada como forma de interação e visualização em um sistema de estúdio virtual.

4.1 SISTEMA ARSTUDIO VERSÃO 1

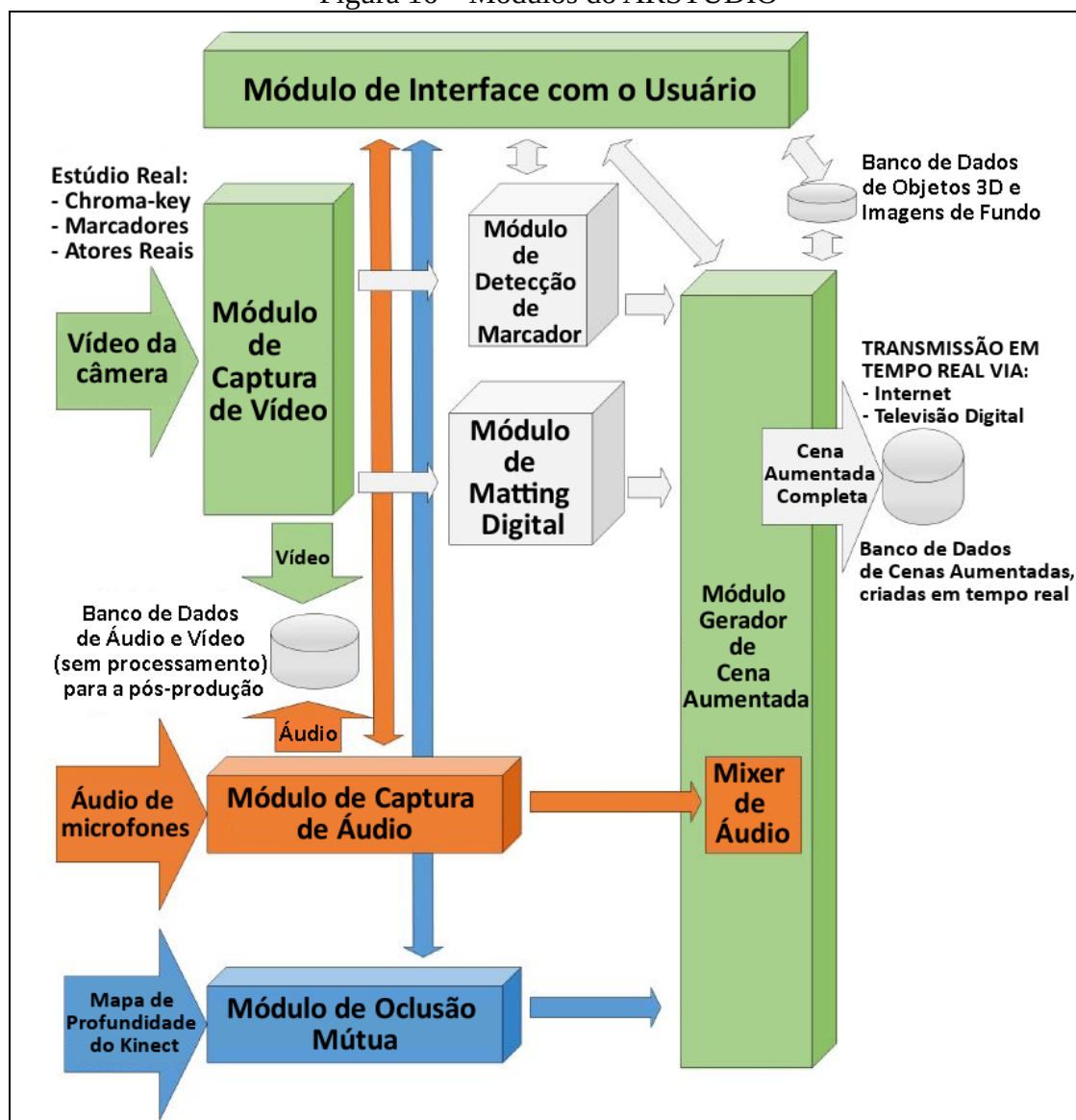
O intuito do projeto ARSTUDIO é o desenvolvimento de um sistema de estúdio virtual, de baixo custo, com RA, que utiliza técnicas de *chroma-key*, emprega a oclusão mútua através da profundidade obtida por meio do sensor Kinect v1, permite a renderização e a visualização da imagem composta de elementos reais e virtuais, permite a interação dos atores com os elementos virtuais e adiciona som ao vídeo aumentado, gerado pelo sistema. O projeto é desenvolvido na UNESP-Bauru e participam dele pesquisadores do Programa de Pós-graduação em Mídia e Tecnologia (PPGMiT/FAAC), do Programa de Pós-graduação em Ciência da Computação (PPGCC/FC) e do programa de Graduação em Ciência da Computação (FC) (GASPARI, et al., 2014; SEMENTILLE et al., 2014).

O ARSTUDIO é um sistema de estúdio virtual com RA que permite a inclusão de efeitos especiais em tempo real, diferentes dos sistemas atuais onde os efeitos especiais são

adicionados durante a etapa de pós-produção em uma cadeia de produção convencional. Esse sistema utiliza equipamentos convencionais (webcam, computador, televisão, iluminação, tela de *chroma-key*) e não convencionais (sensor Kinect v1, marcadores de RA) de baixo custo. O uso de marcadores fiduciais possibilita a inserção de elementos virtuais na cena, por meio de técnicas de RA (GASPARI, et al., 2014)

O sistema ARSTUDIO foi desenvolvido em uma arquitetura modular, para facilitar a manutenção e reutilização do código. Esta arquitetura foi composta pelos seguintes sete módulos interdependentes: captura de vídeo, interface com o usuário, detecção de marcadores, *chroma-key*, captura de áudio, oclusão mútua e gerador de cena combinada (Figura 16).

Figura 16 – Módulos do ARSTUDIO



Fonte: Adaptada de GASPARI et al., 2014

Cada módulo é responsável por uma executar funcionalidade como (GASPARI et al., 2014):

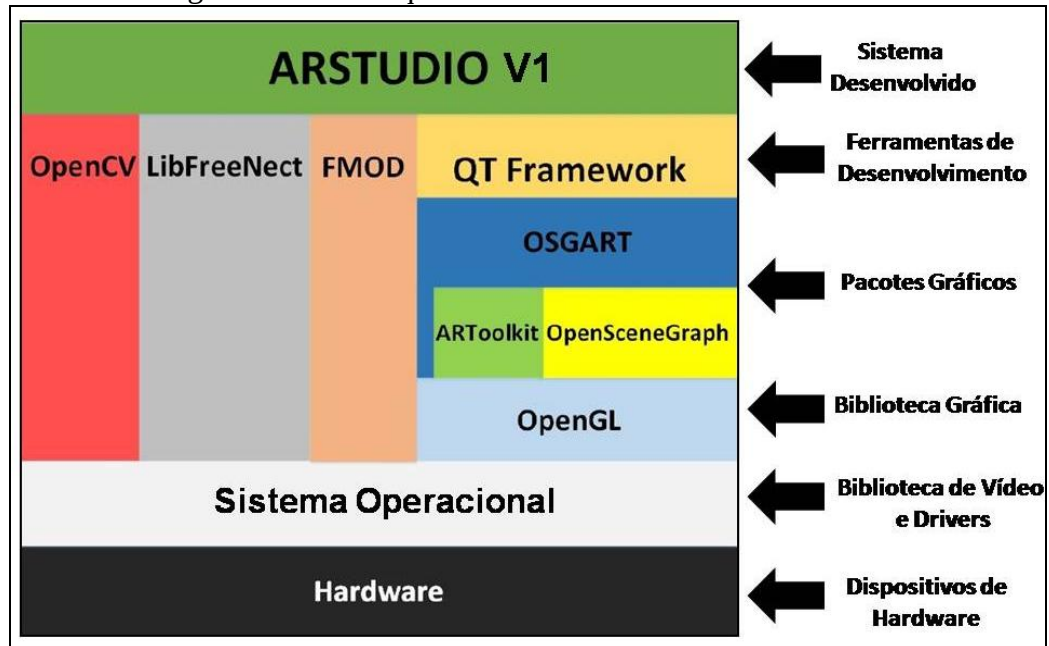
1. *Módulo de Captura de Vídeo*: responsável pela captura de vídeo proveniente da câmera real, pela transmissão dessas informações para que outros módulos possam utilizar e processar e pelo armazenamento local do mesmo;
2. *Módulo de Interface com o Usuário*: permite a configuração das cenas e dos objetos virtuais, bem como a interação do usuário com todo o sistema;
3. *Módulo de Detecção de Marcador*: usando técnicas de visão computacional, o vídeo é recebido pelo módulo de captura de vídeo, o vídeo é então processado, identificando marcadores encontrados na cena real e esse módulo também apresenta a possibilidade de controlar os marcadores presentes na cena;
4. *Módulo de Matting Digital*: realiza técnicas de *matting* digital, para substituir, pixel-a-pixel, a imagem de fundo da cena real, usando métodos baseados em *chroma-key* e *color difference-key*;
5. *Módulo de Captura de Áudio*: responsável por capturar o áudio de microfones no cenário real;
6. *Módulo de Oclusão Mútua*: executa, com informações de profundidade, a oclusão mútua entre objetos virtuais e os atores reais em cena;
7. *Módulo Gerador de Cena Aumentada*: a partir das informações geradas por todos os outros módulos é gerada a cena aumentada final, em tempo real, para transmissão pela internet ou TV digital, e/ou é armazenado o fluxo de vídeo localmente.

A cena aumentada gerada pelo sistema é estruturada como um grafo de cena, o que permite criar cenas virtuais complexas de forma incremental, onde vários objetos podem ser correlacionados, e alterá-las em tempo real, através do módulo de interface do usuário (GASPARI et al., 2014).

No desenvolvimento do ARSTUDIO, utilizou-se de bibliotecas multiplataforma e software livre. Nesta hierarquia (Figura 17), a camada de aplicação é o nível mais alto, com base no arcabouço de desenvolvimento Qt. Abaixo, bibliotecas de realidade virtual (OpenSceneGraph) e realidade aumentada (ARToolKit) estão localizadas, e a biblioteca osgART apresenta um nível que englobam as duas anteriores. Em direção ao hardware está a biblioteca gráfica OpenGL, que suporta as outras bibliotecas. As biblioteca OpenCV, LibFreeNect e a API FMOD se encontram em um nível paralelo ao Qt, osgART e ao OpenGL e se comunicam entre as camadas do sistema ARSTUDIO e o sistema operacional. Por fim

estão as camadas do sistema operacional e os dispositivos de hardware (GASPARI et al., 2014).

Figura 17 – Hierarquia de bibliotecas do ARSTUDIO 1.0

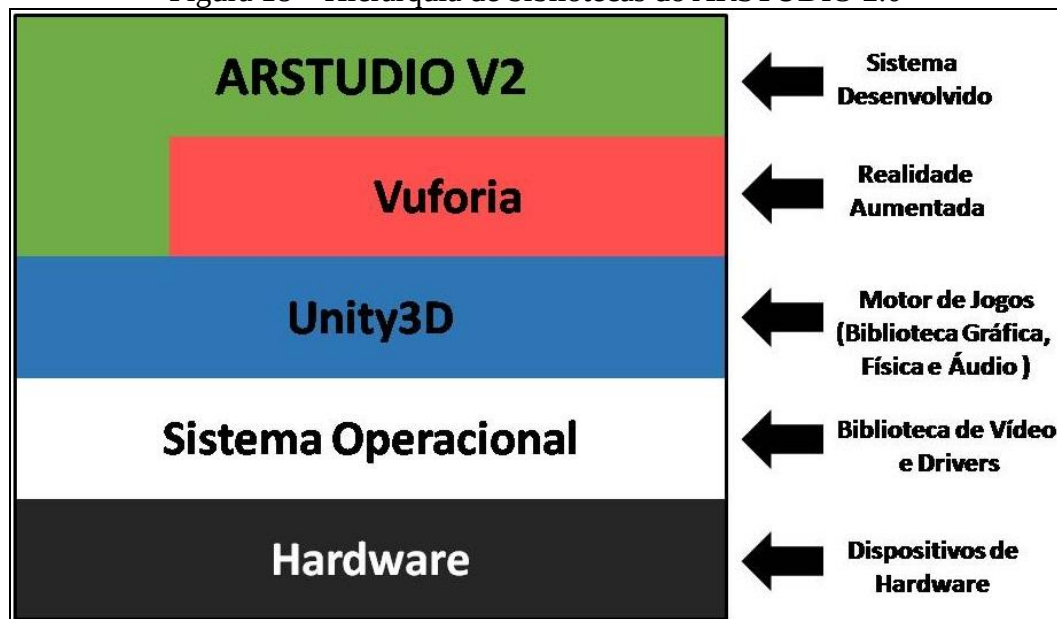


Fonte: Adaptada de GASPARI et al., 2014

4.2 ESTRUTURAÇÃO DO ARSTUDIO VERSÃO 2

Por se tratar de um software com uma arquitetura nova, foi necessária a construção de uma nova hierarquia (Figura 18) e ambiente de desenvolvimento. Para isso foram selecionados dois softwares principais, o Unity3D e o Vuforia.

Figura 18 – Hierarquia de bibliotecas do ARSTUDIO 2.0



Fonte: Elaborada pelo autor

4.2.1 A ENGINE GRÁFICA UNITY E A BIBLIOTECA VUFORIA

Para a base do sistema foi utilizado o motor de jogos Unity3D, visto que o mesmo consta com recursos como: a possibilidade de importação de diversos formatos de modelos tridimensionais (fbx, obj, dae, max, blend), Figuras (psd, jpg, png, gif, etc.), áudios (mp3, ogg, wav, etc.), vídeos (mov, avi, mp4, mpeg, etc.), animações, entre outros; a compatibilidade com plataformas móveis (Apple iOS, Google Android, Windows phone e BlackBerry 10), consoles (Microsoft Xbox 360 e Xbox One, Sony PlayStation 3 e PlayStation 4 e Nintendo Wii e Wii U) e computadores de mesa (Microsoft Windows, Apple Macintosh e Linux). Ele ainda suporta um Webplayer (possibilidade de executar jogos nos navegadores) para implantação em todos os principais navegadores da web, 27 plataformas no total; a possibilidade de estender o ambiente, interface de usuário, do Unity; a inserção/geração de efeitos especiais; a compatibilidades com diversos dispositivos de entrada (controles, câmeras, sensores, etc.); e diversos recursos gráficos de iluminação e renderização em tempo real (GREGORY, 2009; UNITY3D, 2017a).

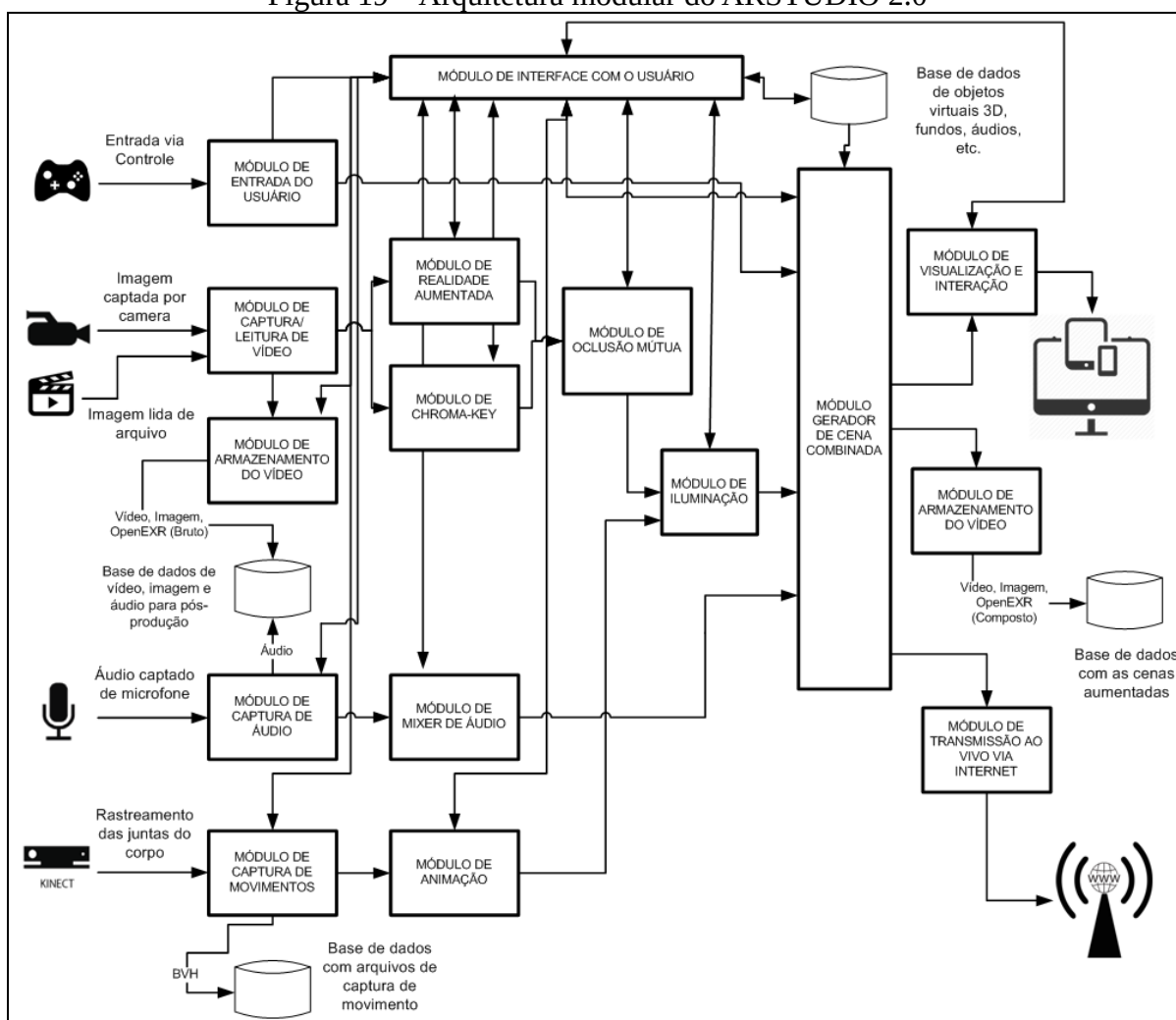
Os principais objetivos de design do Unity são a facilidade de desenvolvimento e a implantação de jogos multiplataforma. Como tal, o Unity fornece um ambiente de editor integrado fácil de usar, no qual oferece a possibilidade de criar e manipular os recursos que compõem o mundo de jogo e visualizar rapidamente o jogo em ação no editor ou diretamente no hardware destino. O Unity também fornece um conjunto de ferramentas para analisar e otimizar o jogo em cada plataforma de destino. O Unity suporta a programação (*scripts*) em JavaScript e C#; um sistema de animação que suporta o *retargeting* da animação (a capacidade de reproduzir uma animação criada para uma personagem em uma personagem diferente); e oferece suporte para jogos multiplayer em rede (GREGORY, 2009).

Para a implementação do recurso de realidade aumentada foi utilizado o software Vuforia, o qual é possível ser incorporado ao ambiente do Unity expandindo assim as suas funcionalidades. O Vuforia possibilita a obtenção de uma visão do mundo real através de uma câmera conectada ao computador, o que então permite a detecção de marcadores, predefinidos, e o uso da realidade aumentada ao encontrar os mesmos na cena (VUFORIA, 2017).

4.2.2 ARQUITETURA DO ARSTUDIO VERSÃO 2

O sistema ARSTUDIO 2.0 foi desenvolvido em uma arquitetura modular, para facilitar a manutenção e reutilização do código. Esta arquitetura foi composta pelos seguintes quinze módulos interdependentes: captura/leitura de vídeo, armazenamento do vídeo, Realidade Aumentada, captura de áudio, captura de movimentos, animação, entrada do usuário, interface com o usuário, *chroma-key*, iluminação, oclusão mútua, mixer de áudio, gerador de cena combinada, visualização e interação (Figura 19).

Figura 19 – Arquitetura modular do ARSTUDIO 2.0



Fonte: Elaborada pelo autor

Cada módulo é responsável por um exercer uma funcionalidade como:

1. *Módulo de captura/leitura de vídeo*: responsável pela captura do fluxo de vídeo proveniente de uma câmera real ou pela leitura de um arquivo de vídeo e pela transmissão desse vídeo para que outros módulos possam utilizar e processar;

2. *Módulo de armazenamento do vídeo*: responsável pelo armazenamento local do vídeo (bruto ou aumentado) nos formatos de vídeo, imagens (*frame-a-frame* ou a cada *n frames*) ou em OpenEXR;
3. *Módulo de Realidade Aumentada*: usando técnicas de visão computacional, o vídeo é recebido pelo módulo de captura de vídeo, o vídeo é então processado, identificando marcadores encontrados na cena real e esse módulo também apresenta a possibilidade de controlar os marcadores presentes na cena e interagir por meio de botões virtuais;
4. *Módulo de captura de áudio*: responsável pela captura de áudio proveniente do microfone, pela transmissão desse áudio para que outros módulos possam utilizar e processar e pelo armazenamento local do mesmo;
5. *Módulo de captura de movimentos*: responsável pela aquisição das informações de posicionamento e orientação das juntas do ator, bem como o armazenamento local do mesmo;
6. *Módulo de animação*: a partir das informações obtidas pelo módulo de captura de movimentos, permite a transposição dos movimentos (animação) em uma personagem do banco de dados;
7. *Módulo de entrada do usuário*: permite a entrada de ações via as entradas tradicionais (mouse e teclado) e o controle de videogame;
8. *Módulo de interface com o usuário*: possibilita a configuração da cena e dos objetos virtuais, bem como a interação do usuário com todo o sistema;
9. *Módulo de chroma-key*: realiza técnicas de *chroma-key* (segmentação binária) para substituir a imagem de fundo da cena real;
10. *Módulo de iluminação*: permite a configuração da iluminação por meio de fontes virtuais de luz e HDRI na cena virtual;
11. *Módulo de oclusão mútua*: executa, com informações de profundidade dos marcadores do módulo de Realidade Aumentada, a oclusão *mútua entre objetos virtuais e os atores reais em cena*;
12. *Módulo de mixer de áudio*: permite à mixagem de diversas fontes de áudio, o áudio é recebido pelo módulo de captura de áudio e pelos áudios do banco de dados;
13. *Módulo gerador de cena combinada*: a partir das informações geradas por todos os outros módulos, gera a cena aumentada final, em tempo real, para a transmissão ao módulo de visualização e interação, transmissão via internet ou TV digital, ou armazena o vídeo localmente.
14. *Módulo de visualização e interação*: responsável pela exibição, em tempo real, da cena aumentada para retorno aos atores, operador do sistema do ARSTUDIO 2.0,

diretor e ao operador de câmera. Esse módulo também possibilita a interação do diretor e do operador do sistema/câmera.

15. *Módulo de transmissão ao vivo via internet*: permite a transmissão ao vivo da cena aumentada e do áudio do microfone pela internet, através de utilização de uma câmera virtual.

4.2.3 O ARSTUDIO 2.0 E A CADEIA DE PRODUÇÃO OTIMIZADA

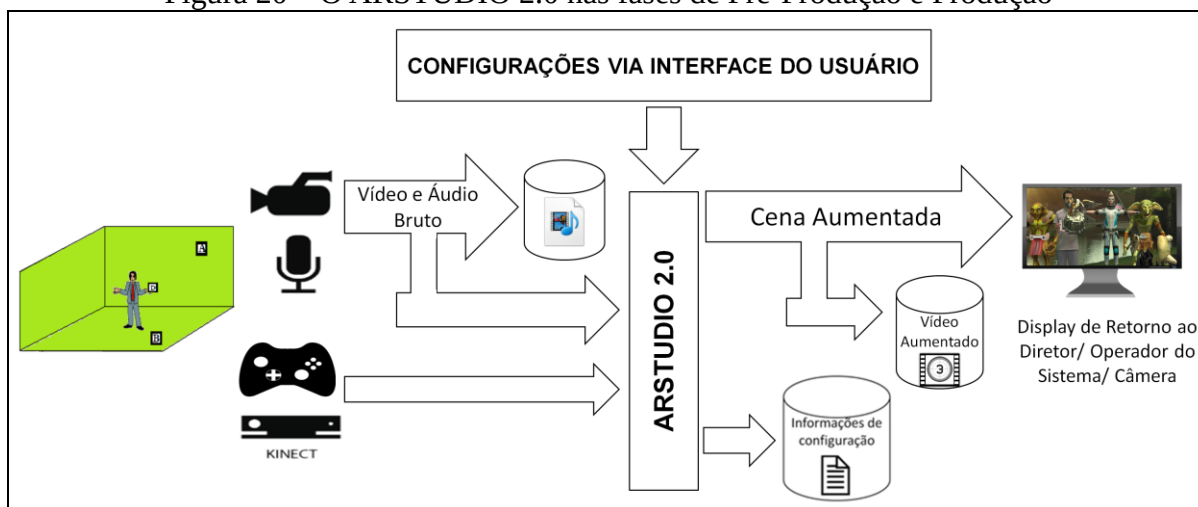
O ARSTUDIO 2.0, por meio da criação de um protótipo em software, foi projetado de forma que contempla uma grande parte das atividades da cadeia de produção de conteúdo midiático que exija a inserção de elementos virtuais e efeitos especiais. O sistema foi estruturado de forma a atuar nas três etapas da cadeia de produção: pré-produção, produção e pós-produção. O sistema também faz uso de um banco de dados com modelos virtuais 3D e 2D, áudios e vídeos que podem ser escolhidos e utilizados durante as etapas da cadeia.

Na etapa de pré-produção, o estúdio virtual pode ser utilizado no planejamento do conteúdo (elaboração do roteiro, roteiro técnico e *storyboard*), permitindo a visualização e interação com objetos 3D (de baixa resolução).

Na etapa de produção, o conteúdo é gravado seguindo-se o roteiro. O ARSTUDIO realiza, então, a combinação dos elementos 3D e cenários virtuais (com média resolução) com os elementos reais (atores e adereços), em tempo real. Os atores também podem interagir com os elementos virtuais em tempo real. A cena aumentada pode ser visualizada em tempo real, tanto pelos atores, quanto pelo diretor e operadores de câmera.

Conforme mostrado na Figura 20, a captura da cena é feita por meio de câmeras (webcams ou profissionais). A entrada do usuário é realizada mediante dispositivos convencionais (mouse e teclado), na interface de usuário, e por controle de videogame. As informações de captura de movimentos são obtidas através do sensor de profundidade, Kinect v2. O sistema também conta com o rastreamento de marcadores (fiduciais e de textura), permitindo a geração de uma cena aumentada com coerência de profundidade.

Figura 20 – O ARSTUDIO 2.0 nas fases de Pré-Produção e Produção



Fonte: Elaborada pelo autor

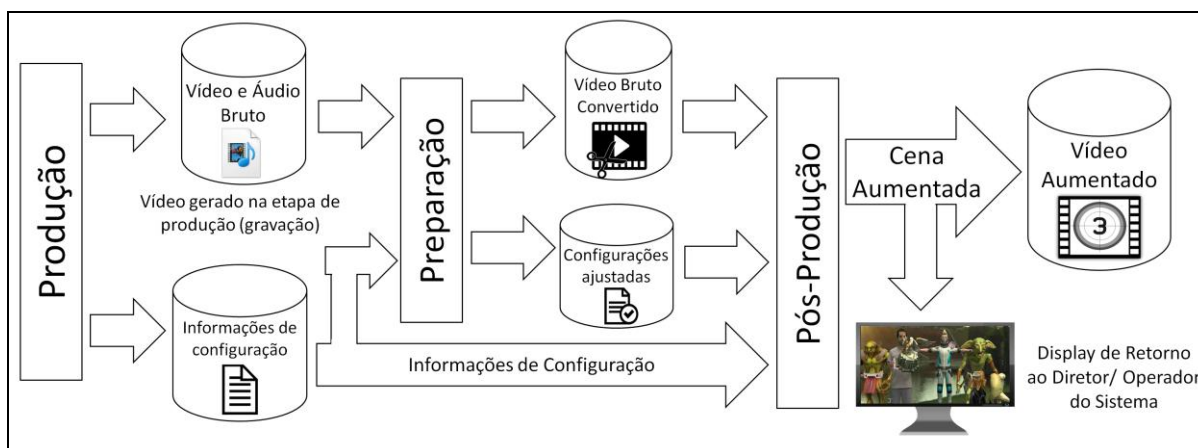
Os fluxos de vídeo e áudio podem ser armazenados na forma bruta (sem adição de elementos virtuais), em alta definição para processamento na etapa de pós-produção. As configurações (escolha da imagem/vídeo de fundo, áudios, marcadores, modelos, luzes virtuais, *chroma-key*, etc.) realizadas via a interação do usuário com o sistema também são armazenadas, bem como a cena aumentada gerada em tempo real.

Para que o modo de pós-produção fosse possível, foi necessária a execução das três seguintes etapas, durante a cadeia de produção:

- a) *Produção*: onde, durante a gravação, é realizado o armazenamento do fluxo de vídeo bruto, com áudio, e das informações de configuração;
- b) *Preparação*: etapa onde a edição e conversão do formato do vídeo são realizadas, as informações de configuração ajustadas (por exemplo, uma alteração na escala de um modelo, momento de disparo de áudio/animação), caso necessário, e a imagem/vídeo de fundo e os modelos são substituídos por equivalentes em alta resolução;
- c) *Pós-Produção*: onde o vídeo bruto e as informações de configuração servem de entrada, gerando-se, como resultado final, um vídeo aumentado de maior qualidade.

A Figura 21 ilustra a sequência de atividades necessárias para a execução do modo de pós-produção.

Figura 21– O ARSTUDIO 2.0 na fase de Pós-Produção



Fonte: Elaborada pelo autor

Com esse modo de pós-produção, o conteúdo obtido na fase de produção pode ser reprocessado pelo ARSTUDIO 2.0, substituindo-se os algoritmos de execução em tempo real por algoritmos mais eficientes e os elementos 3D, utilizados na fase de produção, por equivalentes em alta resolução, gerando-se, então, como resultado final, vídeos com maior qualidade.

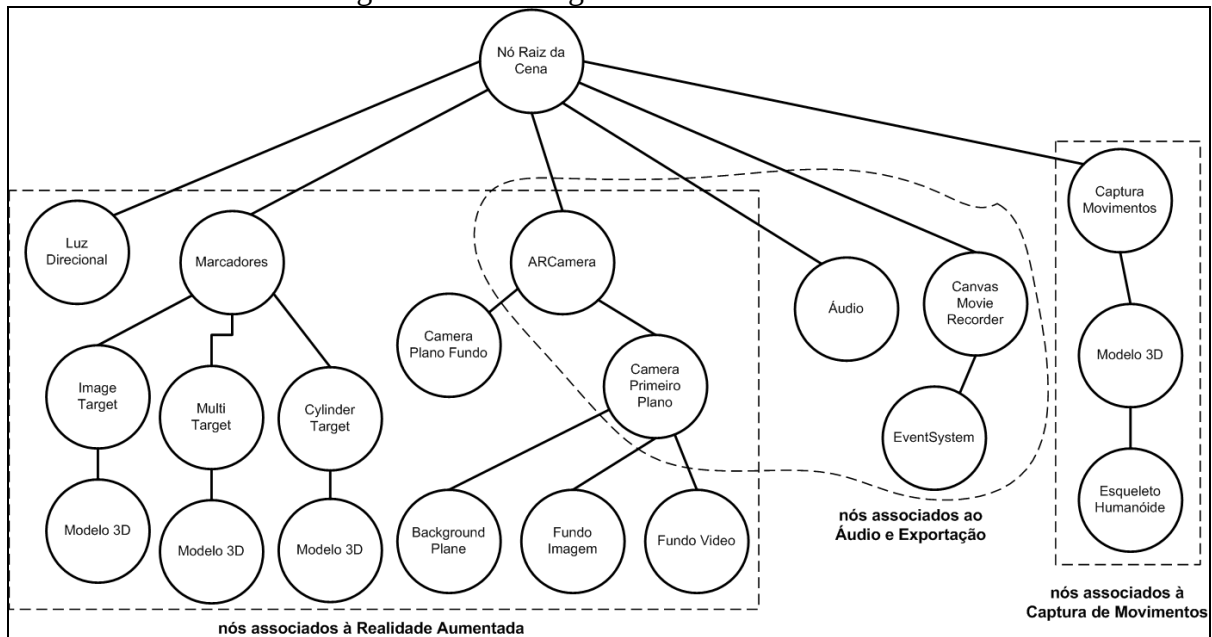
4.2.4 GRAFOS DE CENA DO ARSTUDIO VERSÃO 2

A unidade básica do Unity é o *GameObject*. A cena virtual elaborada no Unity possui uma hierarquia elaborada por uma combinação de *GameObjects* que formam um grafo de cena¹². Os *GameObjects* possuem informações/características chamadas de componentes (*component*). Esses componentes podem conter informações específicas como: transformação (posição, escala e rotação); modelo 3D associado, câmera virtual, áudio, iluminação, animação e principalmente os scripts que agregam funcionalidades específicas aos *GameObject* associado (UNITY3D, 2017c).

O grafo geral de cena do ARSTUDIO 2.0 (Figura 22) abrange os módulos de seu sistema. Esse grafo geral de cena é o grafo que possui o maior número de recursos desenvolvido neste trabalho. Esse grafo maior pode ser dividido em três grafos principais, o grafo de RA, o grafo de captura de movimento e o grafo de áudio e exportação. Cada grafo possui uma hierarquia única, compõe uma cena diferente que será renderizada e as funções que serão executadas naquela cena específica.

¹² Estrutura hierárquica de nós que representa a cena virtual. Os nós possuem geometrias ou outras estruturas da cena. O grafo de cena determina como a cena será renderizada.

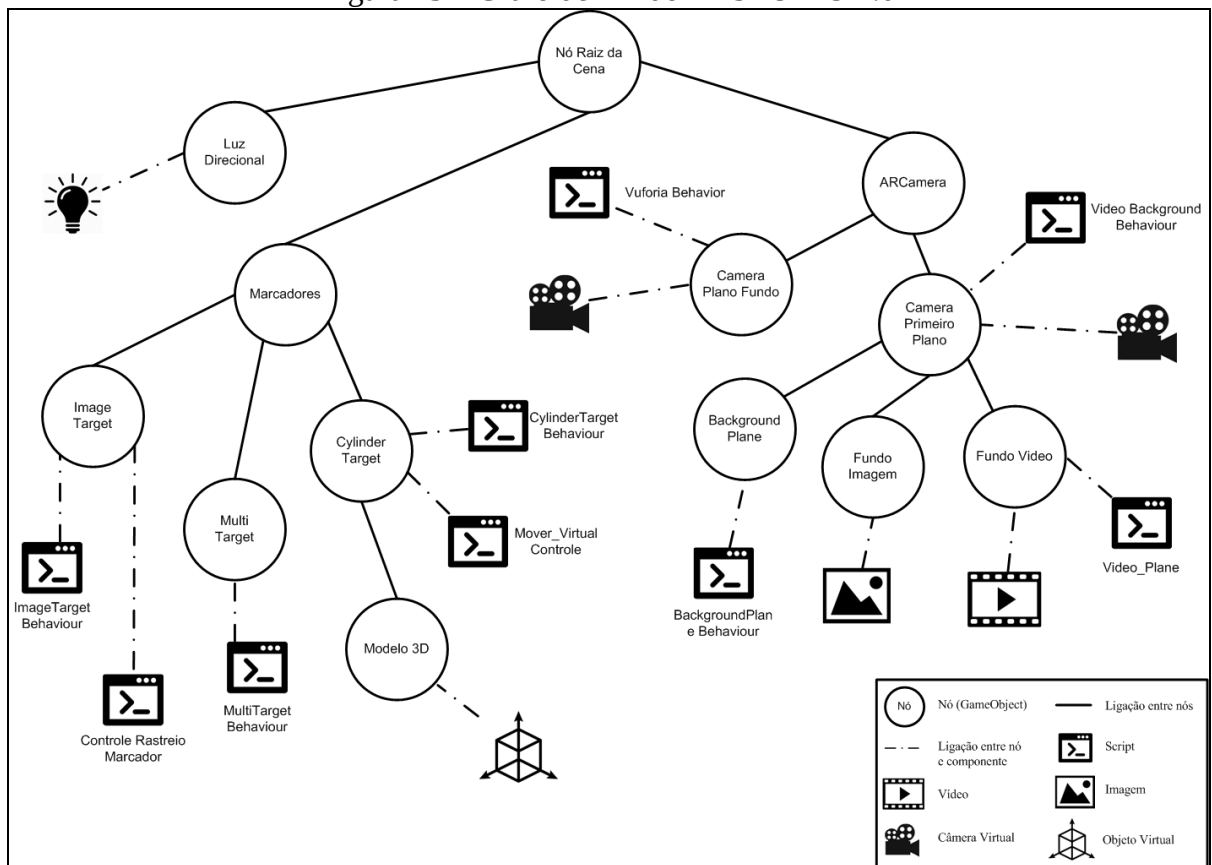
Figura 22 – Grafo geral do ARSTUDIO 2.0



Fonte: Elaborada pelo autor

O grafo de cena de RA (Figura 23) abrange os seguintes módulos do sistema: captura/leitura de vídeo, entrada do usuário, Realidade Aumentada, *chroma-key* e iluminação.

Figura 23 – Grafo de RA do ARSTUDIO 2.0



Fonte: Elaborada pelo autor

Os principais *GameObjects* e seus componentes são:

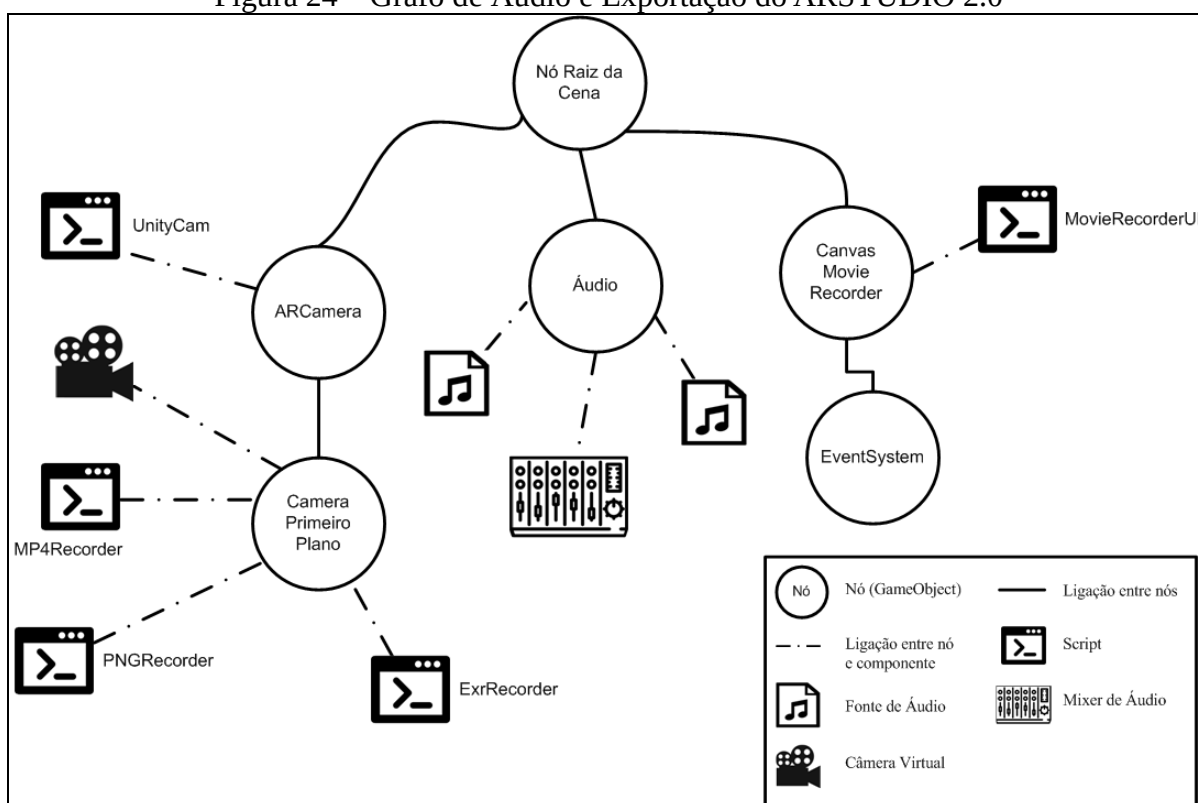
1. *ARCamera*: possui o componente “VuforiaBehaviourScript”. Esse *script* é responsável pela escolha e a inserção do banco de dados de marcadores a serem utilizados e procurado na cena, a quantidade de marcadores e serem rastreados e renderizados ao mesmo tempo, pela escolha de qual câmera servirá como entrada ao sistema (quando mais de uma câmera está conectada), pela escolha da qualidade (resolução) da câmera de entrada e pela seleção do centro do mundo virtual (a câmera virtual, o primeiro marcador encontrado pela câmera ou um marcador específico).
2. *Câmera_Primeiro_Plano*: possui os componentes “camera” e “VideoBackgroundBehaviour”. O primeiro componente é responsável pela câmera virtual que irá renderizar os elementos no primeiro plano (por exemplo, os objetos virtuais associados aos marcadores). O segundo componente é um *script* quem obtém o fluxo de vídeo que entra no sistema através da câmera.
3. *BackgroundPlane*: um plano que contém o *script* “BackgroundPlaneBehaviour” onde o fluxo de vídeo, do “VideoBackgroundBehaviour”, será aumentado pelos objetos virtuais e mostrado ao usuário. É nesse plano também que o *chroma-key* é aplicado através de *shaders* (código de baixo nível a ser executado na placa gráfica do sistema) ao material de renderização do plano.
4. *Fundo_Imagem*: contém a imagem que será apresentada ao executar o *chroma-key* e tornar os plano de fundo transparente. Essa imagem está atrás do plano do fluxo de vídeo de câmera (*BackgroundPlane*).
5. *Fundo_Video*: contém a vídeo que será apresentado ao executar o *chroma-key* e tornar os plano de fundo transparente. Para que a execução do vídeo pudesse ser controlada, foi gerado um *script* “Video_Plane” que começa a execução do vídeo e o seu áudio ao começar a capturar o mundo real. Esse código também possibilita a entrada por teclado para parar e continuar a execução do vídeo.
6. *Marcadores*: contém os marcadores a serem utilizados na cena. Mesmo se esse *GameObject* estiver presente na cena, se os marcadores não forem configurados no *ARCamera*, eles não serão procurados e renderizados pelo sistema.
7. *ImageTarget*: possui um *script* denominado “ImageTargetBehaviour”, responsável pelos marcadores planos que serão procurados e utilizados. Esse *script* contém o estado do rastreo do marcador, que pode ser: detectado (o primeiro *frame* quando o marcador foi encontrado), rastreado (o marcador já foi encontrado e continua sendo) e perdido (marcador deixa de ser encontrado pelo sistema). Quando o marcador for “detectado”, o objeto 3D associado a ele será renderizado na cena e quando for “perdido” o objeto deixará de aparecer. Foi criado um *script* “ControleRastreoMarcador” para subscrever os comandos do *script* nativo “ImageTargetBehaviour”. Esse *script* criado é responsável por encerrar as colisões e a

física quando o marcador não está em cena, pois o script nativo deixava os objetos colidindo mesmo quando o marcador não estava sendo mostrado ao sistema.

8. *MultiTarget*: possui o script “MultiTargetBehaviour”, responsável por identificar cada marcador que faz parte do multi-marcador, bem como calcular a diferença de translação e rotação entre esses marcadores (por exemplo, se o multi-marcador estiver no formato de um cubo). Ao encontrar o multi-marcador, o objeto a ele associado será renderizado. No caso de um multi-marcador cúbico, cada face do cubo (um marcador diferente), representa um lado do modelo 3D, ao rotacionar o cubo e trocar qual (ais) face é apresentado ao sistema, o modelo também irá rotacionar.
9. *CylinderTarget*: possui o script “CylinderTargetBehaviour”, responsável por identificar cada marcador que faz parte do marcador cilíndrico e renderizar o objeto 3D a ele associado. Ao rotacionar o cilindro, o modelo 3D também irá rotacionar. Esse GameObject também possui o script “Mover_Virtual_Control”, responsável por capturar as ações do controle de videogame e alterar a posição e a rotação (nos três eixos), do modelo 3D associado ao marcador.
10. *Luz_Direcional*: luz virtual que incide sobre o mundo virtual. Essa luz é responsável não só pelo brilho, mas também pelas sombras entre os elementos virtuais.

O grafo de áudio e exportação (Figura 24) abrange os seguintes módulos do sistema: captura de áudio, mixer de áudio, armazenamento do vídeo, transmissão ao vivo via internet, e o módulo de interface com o usuário.

Figura 24 – Grafo de Áudio e Exportação do ARSTUDIO 2.0



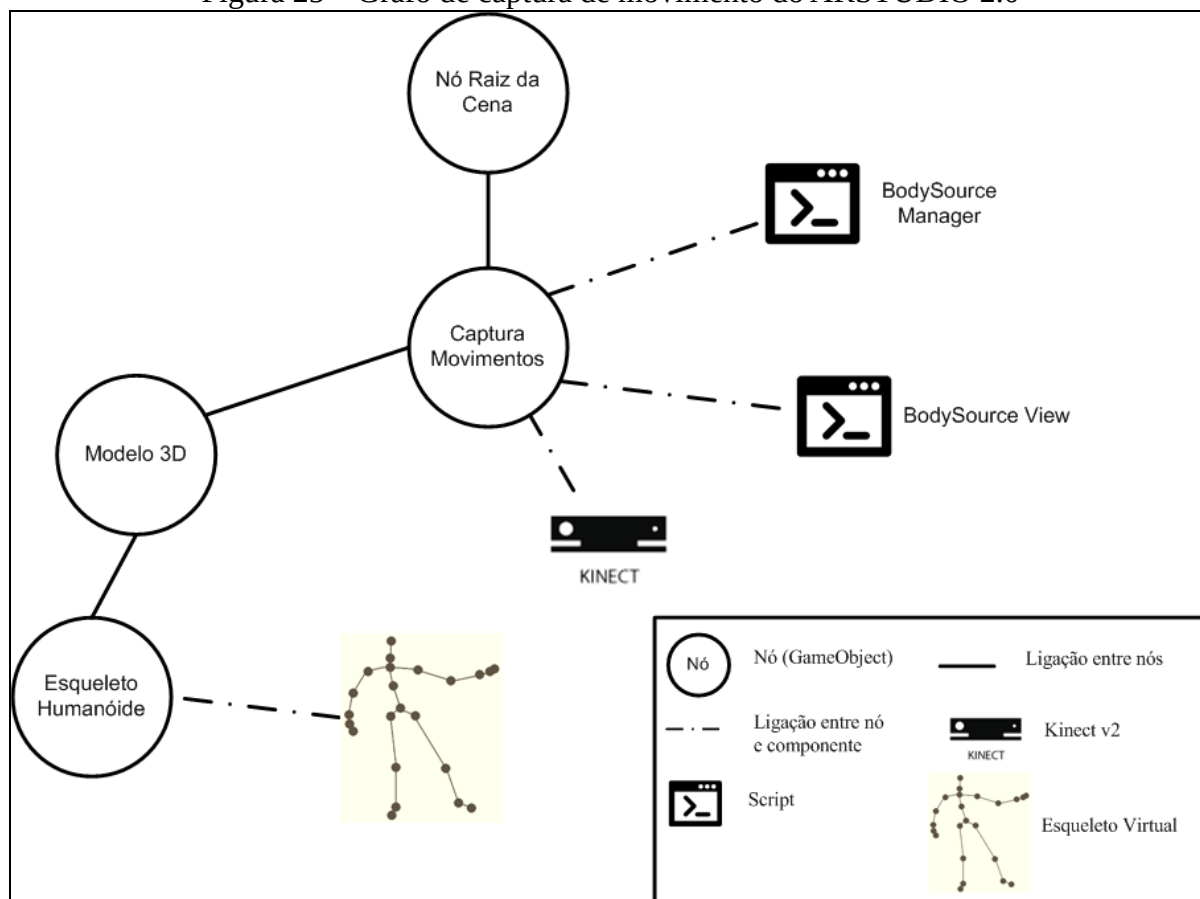
Fonte: Elaborada pelo autor

Os principais *GameObjects* e seus componentes são:

1. *ARCamera*: possui o componente “UnityCam”, responsável por criar uma câmera virtual no Unity, a qual captura a cena gerada pelo sistema e a transmite via internet.
2. *Áudio*: possui os áudios (provenientes de arquivos mp3, wav, etc.) que serão utilizados e executados na cena virtual. Também fornece o uso do *AudioMixer*, onde diversas fontes de áudios podem ser mixadas.
3. *Canvas-MovieRecorder*: possui o componente “MovieRecorderUI”, responsável pela captura de áudio do microfone, bem como a configuração de quanto tempo máximo terá de duração o áudio do microfone, qual será o nome do arquivo de som exportado, e em qual pasta será exportado. Esse *GameObject*, também é responsável por desenhar uma interface de usuário na tela onde o usuário poderá iniciar ou encerrar a gravação (exportação) da cena aumentada com o áudio.
4. *Câmera_Primeiro_Plano*: possui os scripts “MP4Recorder”, “PNGRecorder” e “ExrRecorder”, que junto com o “MovieRecorderUI”, possibilitam que o vídeo aumentado seja exportado nos formatos de vídeo (MP4), imagem (PNG) e OpenEXR.
5. *Event_System*: possibilita a interação com a interface de usuário e também a entrada de controles de videogame.

O grafo de captura de movimento (Figura 25) abrange os seguintes módulos de captura de movimentos e animação.

Figura 25 – Grafo de captura de movimento do ARSTUDIO 2.0



Fonte: Elaborada pelo autor

Os principais *GameObjects* e seus componentes são:

1. *BodySourceManager*: responsável por iniciar o Kinect v2 ao executar o sistema e por atualizar a sua captura a cada *frame*.
2. *BodySourceView*: script responsável por capturar o esqueleto e as juntas do ator com o Kinect v2 e por armazenar as posições e rotações das juntas em um formato BVH exportado pelo sistema.
3. *Esqueleto Humanoide*: o modelo 3D associado a captura de movimento terá de ter um esqueleto compatível com os pontos capturados no ator. Portanto, o esqueleto terá de ser parecido com o de um humano, para que a transposição dos animação do ator ao modelo 3D possa ocorrer.

CAPÍTULO 5 – EXPERIMENTOS E ANÁLISE DOS RESULTADOS

Neste capítulo são apresentados: a composição do ambiente experimental utilizado no projeto; os experimentos (inspeção visual e objetivos) realizados com o protótipo do ARSTUDIO 2.0 e uma análise dos resultados obtidos; e uma breve comparação do protótipo com os principais estúdios virtuais encontrados na literatura, inclusive com o ARSTUDIO 1.0.

5.1 MATERIAIS UTILIZADOS

Diversos experimentos foram realizados durante o desenvolvimento do sistema de estúdio virtual ARSTUDIO 2.0, com a finalidade de analisar o desempenho e a usabilidade do sistema na geração de conteúdo midiático. Para a realização dos mesmos, foi utilizado um ambiente experimental composto por componentes de hardware e software e um ambiente físico controlado.

5.1.1 AMBIENTE FÍSICO

O ambiente físico utilizado foi o laboratório SACI (Sistemas Adaptativos e Computação Inteligente) no Central de Laboratórios de Pesquisa II, campus da UNESP de Bauru.

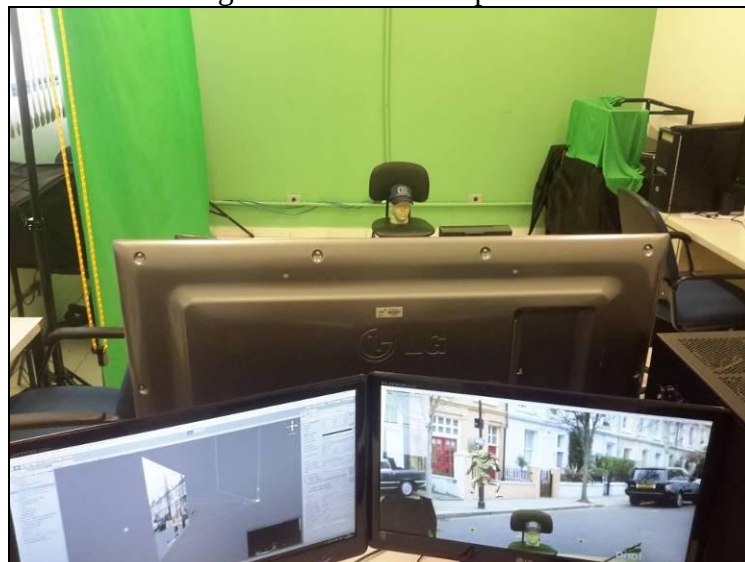
Dispositivos que auxiliaram nos experimentos de gravação foram:

- Weifeng Tripod (modelo WT3730);
- Velbon Tripod (modelo DV-7000);
- Dois Tripés para Iluminação (modelo Greika W-806);
- Tela verde (de *chroma-key*);
- Dois Greika Softbox;
- Dois Photo Bulb E27 125W;
- Plataforma giratória.

Este ambiente de desenvolvimento e testes (Figuras 26 e 27) é composto de duas áreas:

- a) *Mesa do Operador*: local com equipamentos como o computador que executa o sistema, a câmera para gravação, a televisão para retorno ao ator e o sensor de profundidade;
- b) *Área de Gravação*: local de atuação do ator e onde estão os equipamentos utilizados durante a gravação. Por exemplo: luzes, tripés, objetos reais, telas de *chroma-key*, marcadores a serem utilizados como referência dos elementos virtuais, etc.

Figura 26 - Mesa do operador



Fonte: Elaborada pelo autor

Figura 27 - Área de gravação



Fonte: Elaborada pelo autor

O layout adotado para o espaço físico experimental levou em consideração os seguintes aspectos principais: campo de visão da câmera de gravação, posicionamento da iluminação visando à correta aplicação da técnica de *chroma-key* e visualização do retorno para o ator.

5.1.2 COMPONENTES DE HARDWARE E SOFTWARE

Para os experimentos, foram utilizados os seguintes componentes de *hardware*: um computador, para o desenvolvimento e a execução do sistema durante a gravação; uma webcam, uma câmera de ação e uma câmera profissional, todas com resolução 1080p (Full HD); uma placa de captura para obtenção do fluxo de vídeo da câmera de ação e da câmera profissional; um dispositivo Kinect v2, para a captura de movimentos do ator; um *tablet* e um *smartphone*, para retorno ao operador de câmera, do vídeo misturado, e controle de gravação; uma televisão com resolução 1080p, para retornar (de forma espelhada) o vídeo misturado ao ator; um controle de videogame, para auxiliar na manipulação dos objetos virtuais em cena; uma câmera omnidirecional para a geração de um mapa de iluminação do mundo real (HDRI); marcadores planos, multi-marcadores e marcadores cilíndricos para o uso de RA (Figura 28).

As especificações do computador utilizado são:

- Sistema Operacional: Microsoft Windows 10 Pro 64-bit;
- Processador: Intel Core i7-4770 Haswell Quad-Core CPU 3.40GHz;
- Memória RAM: 12.0GB DDR3;
- Placa de vídeo: NVIDIA GeForce GTX 760, 2048MB GDDR5;
- Disco Rígido: 2TB (7200RPM);
- Placa Mãe: ASUS H81M-A/BR;
- BIOS: American Megatrends Inc., v 20.01;
- Dois Monitores LG 23" Flatron E2350V;
- Placa de Captura: AVerMedia Live Gamer HD Lite C985.

Os outros dispositivos utilizados foram:

- Logitech HD Pro Webcam C910;
- Microsoft Kinect for Windows v2 (modelo 1656);
- *Tablet*: Samsung Galaxy Note 10.1 2014 Edition (modelo SM-P601), Android 5.1.1;
- *Smartphone*: Xiaomi Redmi 3S, Android 6.0.1;
- LG HD 3D Smart TV 42" (modelo 42LB6500);
- Câmera profissional: Panasonic AVCCAM HD (modelo AG-AC8);
- Câmera de ação: GoPro HERO3+ Black Edition;
- Controle de videogame: MadCatz (modelo 4716);
- Câmera Omnidirecional: RICOH THETA S;
- Marcadores planos de Realidade Aumentada;
- Multi-marcadores de Realidade Aumentada;
- Marcadores Cilíndricos de Realidade Aumentada.

Os softwares utilizados foram:

- Motor de Jogos: Unity3D 5.5.2f1 (64-bit);
- Sistema de Realidade Aumentada: Vuforia for Unity 6.2.10.

Figura 28 – Equipamentos Utilizados



Fonte: Elaborada pelo autor

5.2 EXPERIMENTOS REALIZADOS

A fim de avaliarem-se as funcionalidades do ARSTUDIO 2.0 foram realizados dois tipos de experimentos:

- Experimentos com inspeção visual, ou seja, que envolvem uma inspeção visual do resultado obtido após a utilização de cada funcionalidade implementada no sistema. Neste caso, foram realizados 18 experimentos diferentes, um para cada funcionalidade do ARSTUDIO.
- Experimentos objetivos (quantitativos), cuja finalidade foi determinar-se o desempenho do sistema quanto à taxa de *frames* de saída, ou seja, após a renderização da cena aumentada. Foram realizados 24 experimentos em 4 cenários diferentes.

5.2.1 EXPERIMENTOS COM INSPEÇÃO VISUAL

Experimento 1 – Teste da técnica de chroma-key

Este experimento teve como finalidade: testar a obtenção de um fluxo de vídeo, proveniente de uma câmera conectada ao computador que captura o mundo real (câmera real);

aplicar a esse fluxo uma técnica de *chroma-key*, em tempo real; trocar o plano de fundo por um plano novo (uma imagem ou vídeo virtual); exibir a cena final (vídeo aumentado), o qual mistura o vídeo proveniente da câmera real com o plano de fundo novo. A técnica de *chroma-key* utilizada foi uma modificação baseada na técnica de segmentação binária desenvolvida por Dawson (2015).

Nessa técnica de *chroma-key*, a cada *frame*, a imagem é processada e é calculada a diferença entre cada pixel da imagem e a cor do fundo que deseja tornar transparente (cor chave). Quando essa diferença for igual à zero (é encontrado um pixel com a mesma cor da cor chave), esse pixel é tornado transparente. Existe uma variável de tolerância que controla o intervalo aceito (intervalo da diferença entre o pixel da imagem e a cor chave). Quanto maior for o valor dessa tolerância, mais cores e tonalidades, com base na cor chave, serão aceitas. Se a cor chave é o verde, ao aumentar o valor da tolerância, outros tons de verdes e cores serão aceitos e tornados transparentes até que se chegue ao valor máximo de tolerância onde todos os pixels da imagem serão tornados transparentes. A configuração da cor chave e do valor da tolerância é feito antes ou durante a gravação. A aplicação dessa técnica de *chroma-key* é ilustrada na Figura 29.

Figura 29 - Aplicação da técnica de *chroma-key* e a inserção de um fundo novo



Fonte: Elaborada pelo autor

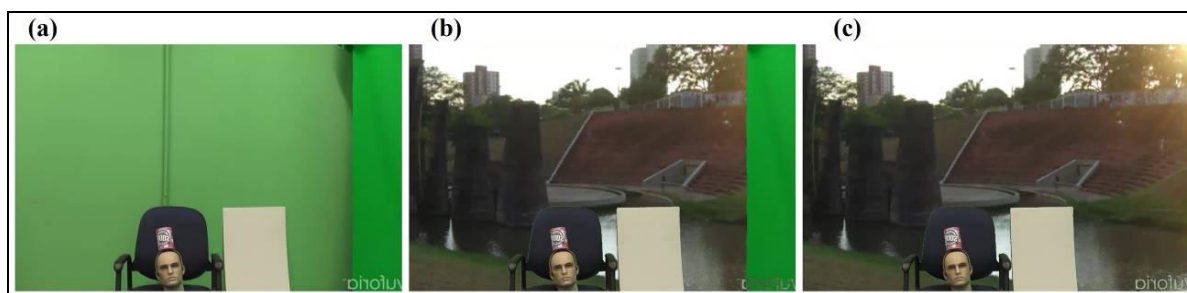
Note-se que foi aplicada a técnica de *chroma-key* com a cor chave verde, a cor da parede (mas que poderia ser de qualquer outra cor). Isso tornou o fundo transparente (primeira coluna), podendo então visualizar o que estava atrás no ambiente virtual (segunda coluna).

Experimento 2 - Teste do aprimoramento da técnica de *chroma-key*

A eficiência da técnica de *chroma-key* é influenciada por diversos fatores como a iluminação do ambiente, a cor chave do material a ser retirada, a transparência de materiais em cena, a resolução da câmera real, sombra, etc. (FOSTER, 2010). Devido a esses fatores, nem sempre o resultado do *chroma-key* é adequado ao uso. Considerando esse contexto, o intuito deste experimento foi de aprimorar a técnica de *chroma-key* no Unity.

Foi desenvolvida uma técnica de *chroma-key*, onde a imagem, proveniente da câmera real, é processada duas vezes (a técnica de *chroma-key* é aplicada duas vezes a imagem da câmera). Com essa técnica, é possível processar a imagem com uma configuração (cor chave e valor de tolerância) e o resultado obtido passa novamente por um processamento de imagem com outra configuração, gerando um resultado mais adequado. O resultado dessa técnica de *chroma-key* é apresentado na Figura 30.

Figura 30 - Aplicação da técnica de *chroma-key*: a) cena não processada; b) primeira aplicação do *chroma-key*; c) segunda aplicação do *chroma-key*



Fonte: Elaborada pelo autor

Observando-se o cenário da Figura 30 nota-se a existência de um fundo com duas tonalidades diferentes de cor chave (Figura 30a). Com apenas uma aplicação da técnica de *chroma-key*, nem todo o fundo é substituído (Figura 30b). No entanto, com a segunda da técnica, todo o fundo é substituído (Figura 30c).

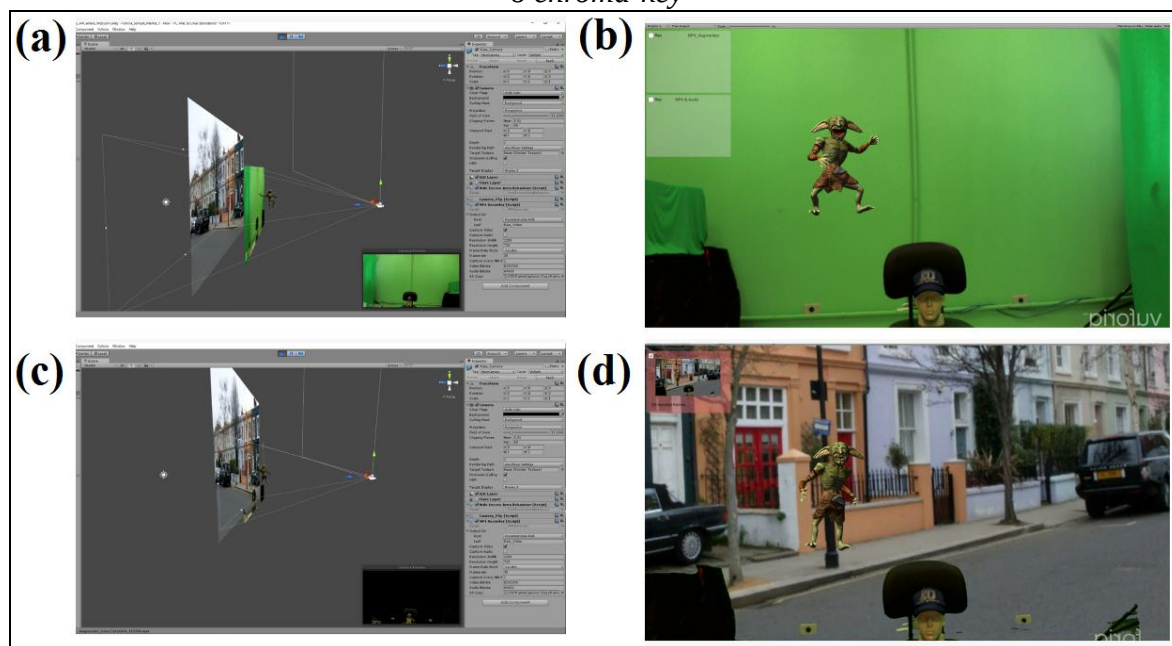
Experimento 3 - Cena aumentada com elementos virtuais e uso da técnica de *chroma-key*

Nesse experimento buscou-se: obter um fluxo de vídeo da câmera real; inserir elementos virtuais, sem referência no mundo real, com animação; gerar e manipular uma cena

composta destes elementos; e exportar a cena composta em forma de um arquivo de vídeo (MP4), em forma de imagens (PNGs) ou no formato OpenEXR¹³ (EXR).

A partir do fluxo de vídeo proveniente da câmera, realizou-se a inserção, através do Editor do Unity (ambiente de desenvolvimento do Unity), de modelos tridimensionais que podiam ser manipulados (inseridos, removidos, transladados, rotacionados, alterados a escala e realizado trocas com outros modelos) durante a gravação. A inserção, devido à ausência de uma referência de posicionamento no mundo real, foi feita manualmente no ambiente virtual, sendo posicionado dentro do espaço de captura da câmera virtual e em frente ao plano que contém o vídeo do mundo real (Figuras 31a e 31c). Nas Figuras 31c e 31d, a técnica de *chroma-key* foi aplicada. Um algoritmo de exportação de vídeo, desenvolvida pelo Unity Technologies Japan (2016), foi modificado e utilizado para exportar a cena composta em forma de vídeo, imagens e OpenEXR.

Figura 31 – Experimento de vídeo proveniente da câmera real, inserção de objetos virtuais e a aplicação da técnica de *chroma-key*: a) editor do Unity sem o *chroma-key*; b) vídeo aumentado sem o *chroma-key*; c) editor do Unity com o *chroma-key*; d) vídeo aumentado com o *chroma-key*



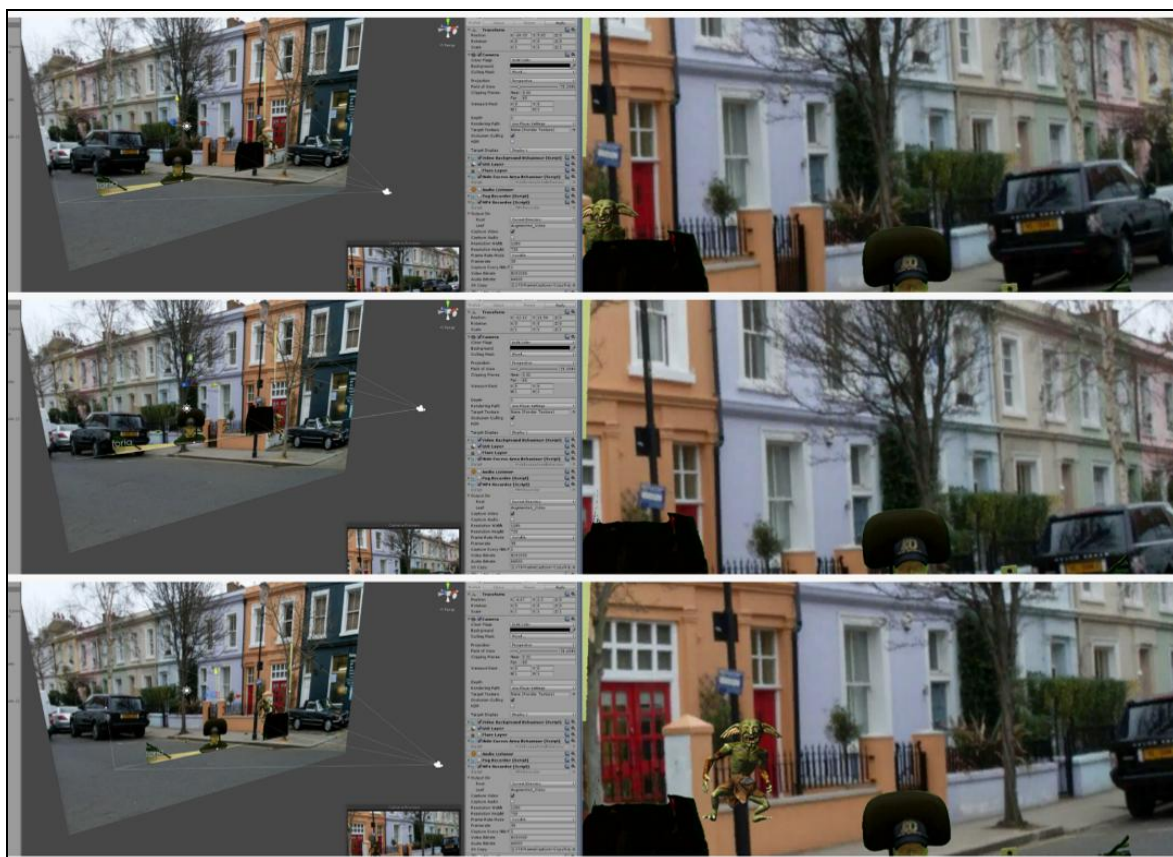
Fonte: Elaborada pelo autor

Observa-se que nas Figuras 31a e 31c, Editor do Unity, existem dois planos no espaço virtual. O primeiro plano, da direita para a esquerda, contém o fluxo de vídeo proveniente da

¹³ Um arquivo de imagem de ampla faixa dinâmica (HDR) desenvolvido pela empresa ILM (Industrial Light & Magic) para uso na indústria de computação gráfica, tanto de efeitos visuais como a de animação (OPENEXR, 2017).

câmera real. No segundo plano, atrás do primeiro, está uma imagem que é apresentada quando o *chroma-key* é configurado e ativado. O segundo plano é maior do que o primeiro devido ao fato de que se a câmera virtual, e o primeiro plano, sofrem uma rotação ou translação, o segundo plano (com a nova imagem de fundo) poderá aparecer, dando assim uma impressão de um espaço maior do que o espaço físico do estúdio real (Figura 32). Existe uma opção de exportar o vídeo, permitindo a exportação do vídeo aumentado no momento da gravação.

Figura 32 - Translação da câmera virtual com a técnica de *chroma-key*



Fonte: Elaborada pelo autor

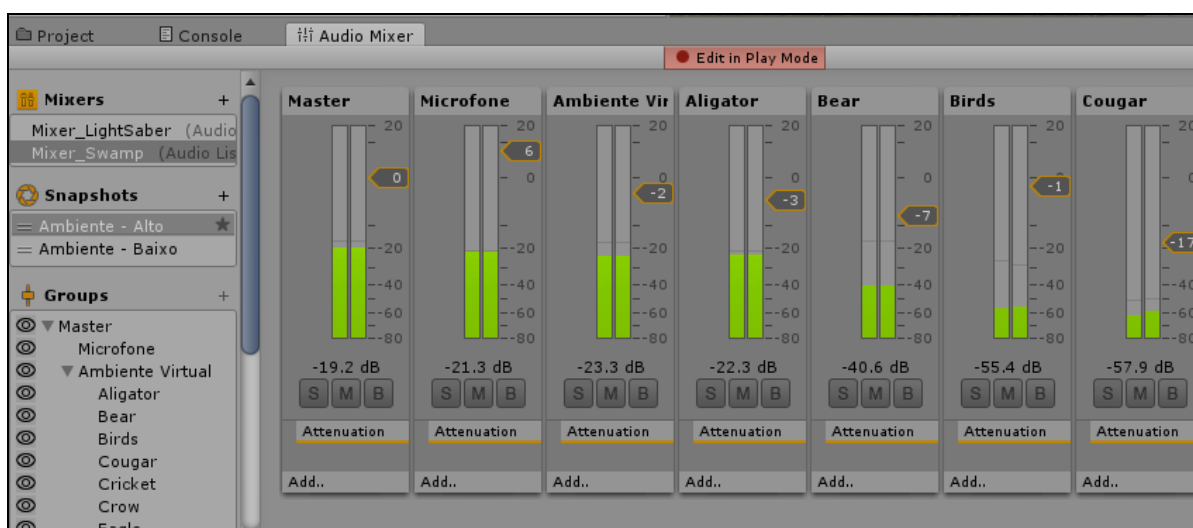
Experimento 4 - Integração de diversas fontes de áudio ao conteúdo gerado

Os objetivos desse experimento foram: obter os áudios da cena virtual (arquivos de áudio importados dentro do sistema); obter o áudio proveniente de um microfone no estúdio (capturar a voz do ator e os sons do ambiente real); capturar/exportar o som do estúdio em arquivo de áudio (wav); mixar os áudios da cena virtual com os do estúdio; e exportar esse áudio mixado, com o vídeo aumentado, em um arquivo mp4.

Nesse experimento, arquivos de áudio (previamente obtidos) foram utilizados, dentro do Editor do Unity, para que houvesse som enquanto fosse capturado o fluxo de vídeo

do mundo real. Esses áudios podiam ser controlados (momento de início e fim) durante a gravação. O som do estúdio foi capturado por meio do microfone da câmera utilizada para capturar o vídeo do mundo real. A mixagem dos áudios da cena virtual com os do estúdio foi feita através de um recurso dentro do Unity chamado de mixer de áudio (*Audio Mixer*), o qual permite a mixagem de diversas fontes de áudio, em tempo real (a interface do *Audio Mixer* é ilustrada pela Figura 33). Após a mixagem, o vídeo final pode ser exportado.

Figura 33 – Experimento da mixagem de áudios da cena virtual com o áudio do microfone no estúdio



Fonte: Elaborada pelo autor

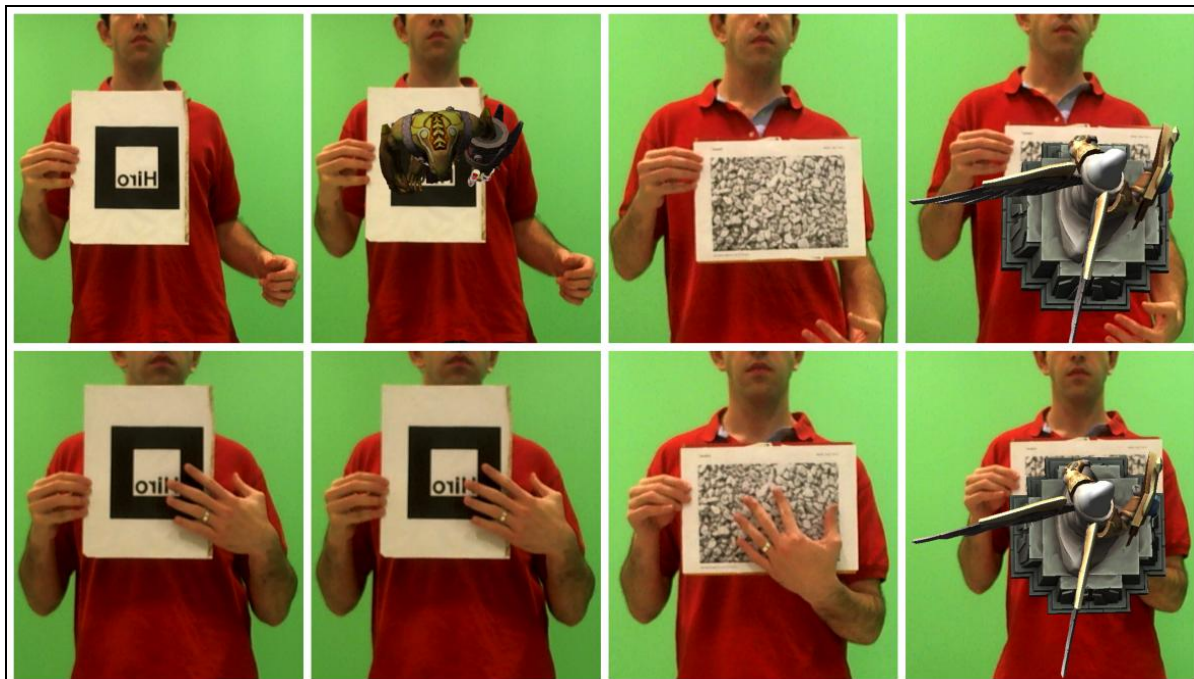
Experimento 5 - Oclusão dos marcadores

A RA busca produzir um ambiente que integre perfeitamente o mundo real e o mundo virtual, onde os objetos aumentados são sobrepostos as imagens provenientes do mundo real (AZUMA, 1997).

Neste experimento buscou-se verificar a qualidade do registro obtido, por meio da inserção de um objeto virtual com referência no mundo real através de marcadores planos do tipo fiduciais e de textura, bem como a comparação da robustez desses dois tipos de marcadores. A Figura 34 apresenta quatro colunas, as primeiras duas sendo a utilização de um marcador fiducial e as demais a utilização de um marcador de textura.

O fluxo de vídeo da câmera real é analisado *frame-a-frame* pelo Vuforia, a procura dos marcadores previamente cadastrados. Assim que o marcador é encontrado pelo sistema, o objeto virtual associado é renderizado sobre o mesmo.

Figura 34 - Experimento da inserção de um objeto virtual com referência a marcadores fiduciais e de textura e a comparação de robustez entre eles



Fonte: Elaborada pelo autor

Pela comparação dos resultados obtidos na segunda coluna e na quarta coluna, nota-se que a utilização do marcador com textura é mais robusto, com relação à oclusão. Isso ocorre devido ao fato de que o marcador com textura possui mais pontos de referência, enquanto que, com relação ao marcador fiducial, é utilizado primeiro o reconhecimento completo da borda para a detecção, ficando mais sujeito a erros de registro. Para ilustrar melhor essa comparação, os dois marcadores passaram pelo algoritmo, do Vuforia, de detecção de pontos de referência e o resultado é apresentado na Figura 35. Pode-se notar que a quantidade de pontos de referência no marcador de textura é superior à quantidade de pontos no marcador fiducial.

Figura 35 – Comparação da quantidade de pontos de referência entre dois marcadores



Fonte: Elaborada pelo autor

O uso do Vuforia, no entanto, apresenta erro de registro com relação à profundidade dos elementos virtuais e reais, onde apesar do ator estar mais próximo da câmera do que o marcador, o objeto virtual está à frente do ator. Esse erro é ilustrado na Figura 36.

Figura 36 – Erro de registro de profundidade entre elementos reais e virtuais



Fonte: Elaborada pelo autor

Experimento 6 - Oclusão mútua entre elementos reais e virtuais

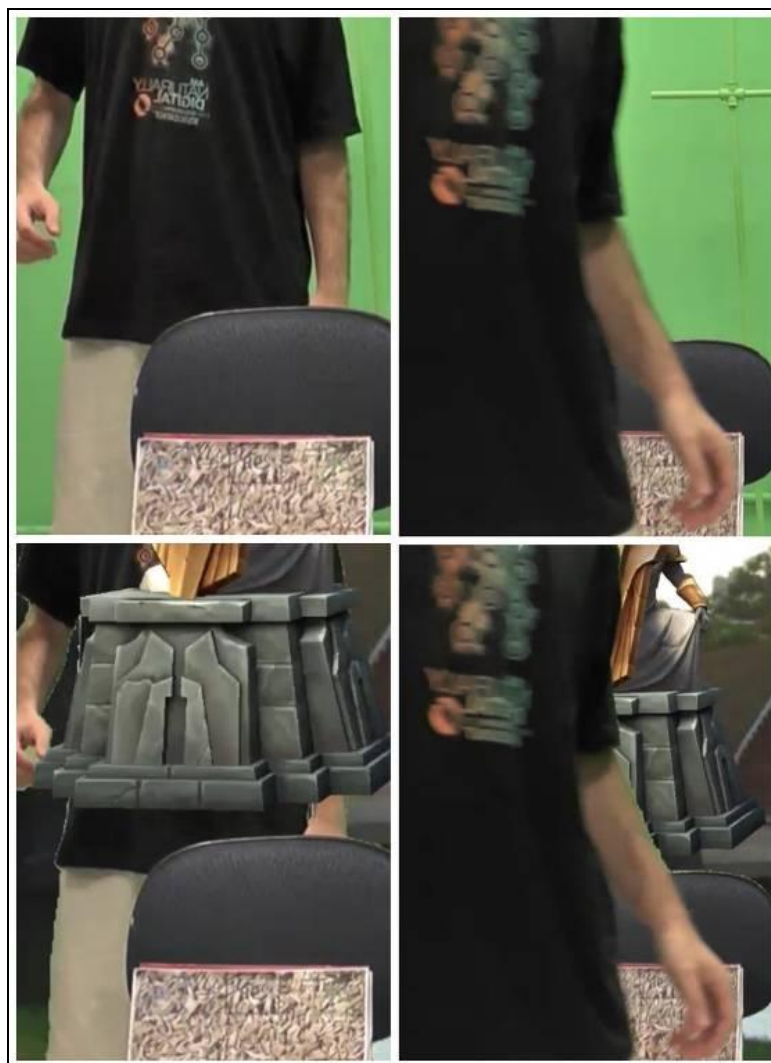
A oclusão mútua¹⁴ é desejável, para que o ambiente misturado fique mais convincente, por ser uma indicação de profundidade no espaço aumentado. Em contrapartida, uma oclusão mútua errada pode confundir o usuário (BREEN, et al., 1996; KIYOKAWA; KURATA; OHNO, 2001).

Em grande parte dos sistemas de RA, os objetos virtuais ficam sempre à frente do plano dos objetos reais, mesmo que o objeto real esteja mais próximo à câmera real do que o marcador referente ao posicionamento do objeto virtual, já que não há informação de profundidade dos elementos reais na cena (SANCHES, et al., 2007).

Considerando esse contexto, esse experimento teve como objetivo verificar a qualidade da inserção de elementos virtuais atrás de objetos reais, por meio da utilização da renderização por camadas (planos), onde a cena final é renderizada em uma ordem pré-definida. Foram criadas duas camadas: a primeira (camada de fundo) contendo os elementos virtuais (plano de fundo novo e objetos virtuais) que devem ficar atrás dos objetos reais e é renderizada primeiro; e a segunda (camada da frente) contendo o fluxo da câmera real e os elementos virtuais que devem ficar à frente dos objetos reais e será renderizada por último. Para que os objetos virtuais renderizados atrás dos objetos reais pudessem ser visualizados, foi necessário o uso da técnica de *chroma-key* para deixar transparente o plano de fundo e então poder visualizar os elementos virtuais que estavam atrás dos objetos reais. O posicionamento do objeto virtual à frente ou atrás de objetos reais é feito de forma manual pelo usuário, configurado previamente ou durante a execução do sistema. A Figura 37 apresenta o uso dessa técnica de renderização por camadas.

¹⁴ Quando um objeto mais próximo ao usuário obscurece a visão de objetos mais distante naquele mesmo campo de visão (BREEN, et al., 1996).

Figura 37 - Experimento da realidade aumentada com a técnica de renderização por camadas



Fonte: Elaborada pelo autor

Com a utilização dessa técnica de renderização por camadas é possível inserir um elemento virtual, com relação a um marcador na cena, de forma que se possa estabelecer, previamente ou durante a execução, se o elemento virtual está à frente ou atrás dos objetos reais.

Experimento 7 - Entradas por meio de controle de videogame

Devido ao fato do Unity ser um motor de jogos, ele conta com diversos recursos voltados para jogos digitais. Um desses recursos é a facilidade de se obter, como entrada de dados, ações realizadas em um controle de videogame (UNITY3D, 2017a).

Nesse experimento foram testadas algumas das funções implementadas por meio da integração de um controle para o computador (MadCatz). Para que as ações do controle, MadCatz, pudessem servir como entrada no Unity, cada botão precisou ser configurado antes de ser utilizado. A configuração do controle foi feita com base na configuração do controle do Xbox360 (UNIFYWIKI, 2015). A Figura 38 apresenta a configuração de cada botão do controle.

Figura 38 – Configuração do controle como entrada no Unity



Fonte: UNIFYWIKI, 2015

Algumas das ações realizadas com o uso da entrada pelo controle foram: alteração da translação ou rotação de elementos virtuais nos eixos x, y ou z; retornar os elementos virtuais aos seus valores iniciais de translação ou rotação; alteração do elemento virtual entre as camadas de frente ou de fundo; e iniciar ou encerrar a gravação da cena.

Essa técnica, porém, possui um problema devido à camada em que o objeto virtual está sendo renderizada ser controlada pelo usuário, e não referente à profundidade real entre os objetos virtuais e reais, estando então sujeito a erro pelo usuário. Esse problema pode causar incoerências onde o objeto virtual estará em um momento à frente e em outro atrás do objeto real devido à troca de camadas pelo usuário operando o sistema, como mostrado na Figura 39.

Figura 39 - Exemplo de incoerência na sobreposição de camadas devido à interferência do usuário



Fonte: Elaborada pelo autor

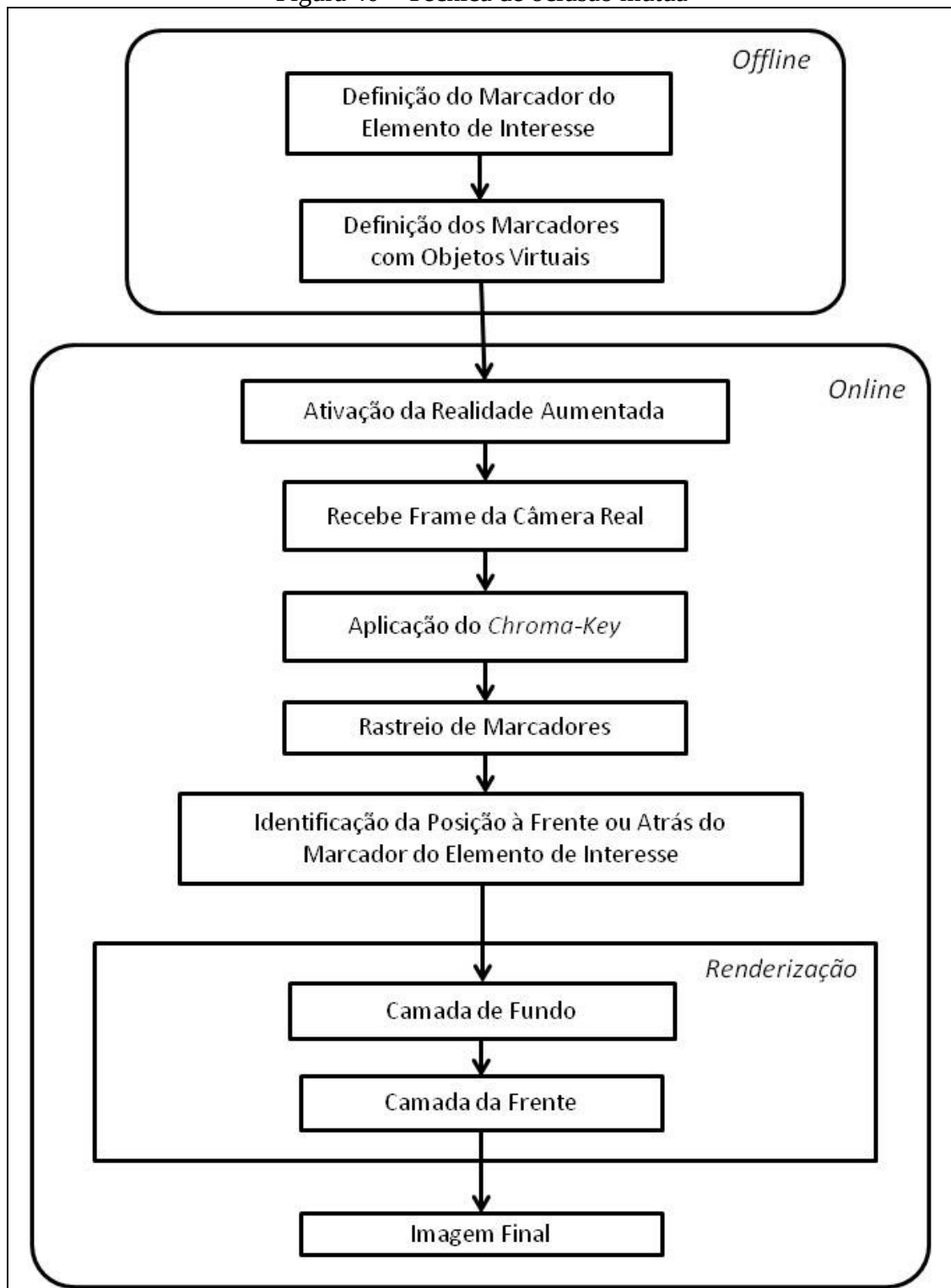
Experimento 8 – Oclusão mútua melhorada

Neste experimento buscou-se verificar a eficácia da resolução encontrada para o problema que surgiu devido ao uso de técnica de renderização por camadas, onde somente o operador do sistema controla se um elemento virtual estará à frente ou atrás do elemento de interesse (elemento real na cena real que deseja posicionar objetos virtuais à frente ou atrás). Para isso, foi criada uma técnica de oclusão mútua baseada naquela desenvolvida por Sanches, et al. (2007).

Para esse experimento, foram utilizados três marcadores de textura, dois deles referentes a objetos virtuais e o terceiro marcador referente ao elemento de interesse. Pelo uso das informações do rastreamento fornecido pelo Vuforia, foi possível obter a distância de cada marcador em relação à câmera real. Definindo qual marcador seria a referência do elemento de interesse e sabendo essas distâncias, um algoritmo foi criado para que quando um marcador, referente a um objeto virtual, estivesse a uma distância maior (em relação à câmera real) do que o marcador referente ao elemento de interesse, o objeto virtual seria colocado na camada de fundo, para ser renderizado atrás do elemento real, e quando o marcador, referente a um objeto virtual, estivesse a uma distância menor da câmera real, o objeto virtual seria

colocado na camada de frente para ser renderizado à frente do elemento de interesse. A Figura 40 ilustra o *pipeline* adotado para a implementação da técnica de exclusão mútua, onde as etapas *offline* e *online* são realizadas previamente e durante a execução do sistema, respectivamente, e a Figura 41 apresenta o resultado de sua utilização.

Figura 40 – Técnica de oclusão mútua



Fonte: Elaborada pelo autor

Figura 41 - Experimento da realidade aumentada com a oclusão mútua: a) ator em frente aos objetos virtuais; b) ator entre os objetos virtuais; c) ator atrás dos objetos virtuais



Fonte: Elaborada pelo autor

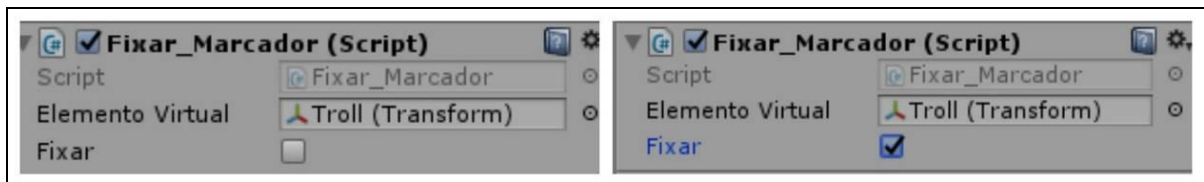
Como pode ser observado, com a técnica de oclusão mútua desenvolvida, é possível que haja uma coerência entre a profundidade e a renderização de objetos virtuais e reais em cena, podendo agora se ter objetos virtuais à frente ou atrás do elemento de interesse de forma automática e não apenas por controle do usuário. Observa-se que na Figura 41a o ator (com o marcador do elemento de interesse em mãos) está à frente aos marcadores dos objetos virtuais, na Figura 41b está entre os marcadores e na Figura 41c está atrás dos marcadores.

Experimento 9 – Fixação de marcadores

A finalidade deste experimento foi verificar a efetividade da solução encontrada para o aperfeiçoamento do registro de objetos virtuais quando os marcadores e a câmera real permanecessem parados durante a gravação e também na retirada de marcadores da cena. Com a retirada do marcador da cena, é possível eliminar os erros de registro, já que o marcador não será procurado a cada *frame*, portanto, não sofrerá possíveis erros de iluminação, ângulo do marcador com relação à câmera, entre outros fatores, e também é possível eliminar o erro na situação em que o elemento virtual não obstrui por completo o marcador a ele vinculado.

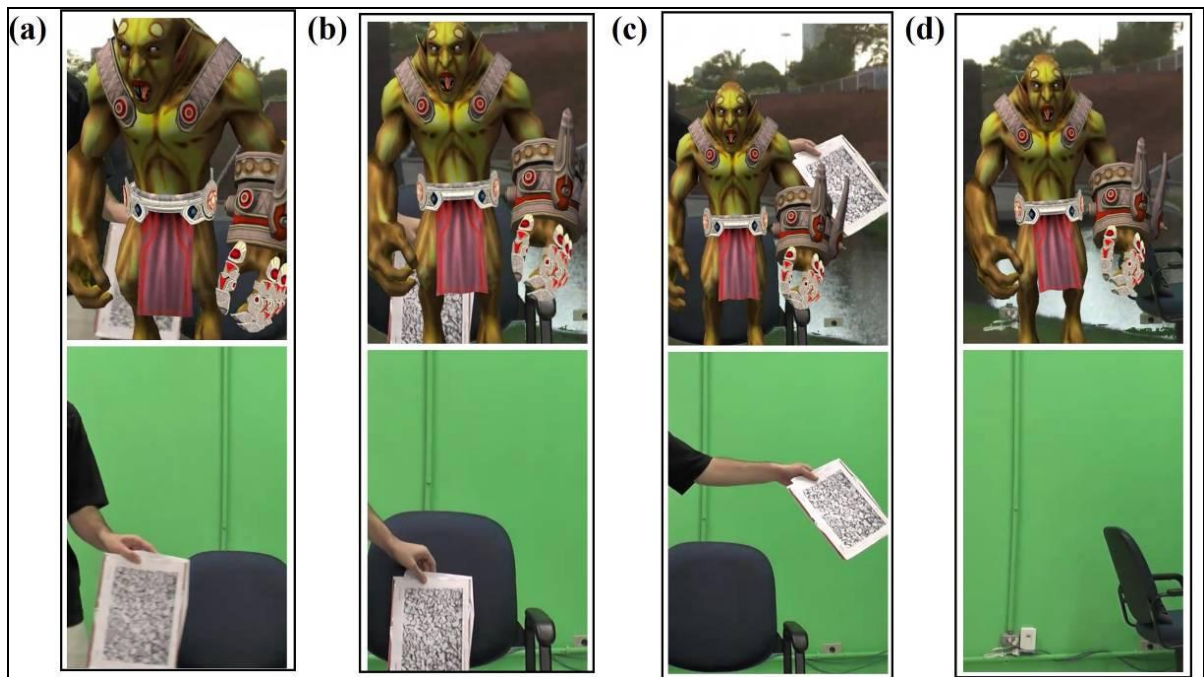
Para resolver esse problema, foi utilizada uma técnica de fixação de marcador, baseada na técnica desenvolvida por Gaspari et al. (2014). O marcador, com um elemento virtual associado, é apresentado à câmera, fazendo com que o elemento virtual apareça na cena. No editor do Unity, foi criado um botão que permite a interrupção ou a continuação na detecção do marcador a partir do *frame* corrente. Esta função pode ser observada no detalhe da interface ilustrada na Figura 42 (botão "Fixar"). Quando esse botão é pressionado o elemento virtual é fixado ao mundo virtual, no posicionamento e rotação calculados para aquele *frame*, e não utiliza mais o marcador como referência para atualizar o seu posicionamento ou a sua rotação e, com isso, o marcador poderá ser retirado da cena. O uso dessa técnica de fixação está apresentado na Figura 43.

Figura 42 – Interface de usuário para fixar o marcador



Fonte: Elaborada pelo autor

Figura 43 – Experimento de fixação de marcador



Fonte: Elaborada pelo autor

Pode ser observado que na Figura 43a o objeto virtual estava associado ao marcador; na Figura 43b o marcador foi fixado; na Figura 43c, apesar do marcador continuar na cena, o objeto virtual não utiliza mais o marcador como referência de posicionamento ou rotação; na Figura 43d o objeto virtual continua em cena, mesmo com o marcador fora da cena.

É possível fixar todos ou somente alguns dos marcadores em cena. A Figura 44 apresenta a fixação de um marcador, com um objeto virtual associado, e um segundo marcador é utilizado com a técnica de oclusão mútua. Com isso é possível que o ator esteja á frente ou atrás do objeto virtual, pela oclusão mútua, sem que seja necessário que o marcador do objeto virtual continue em cena.

Figura 44 – Fixação de marcador com a oclusão mútua

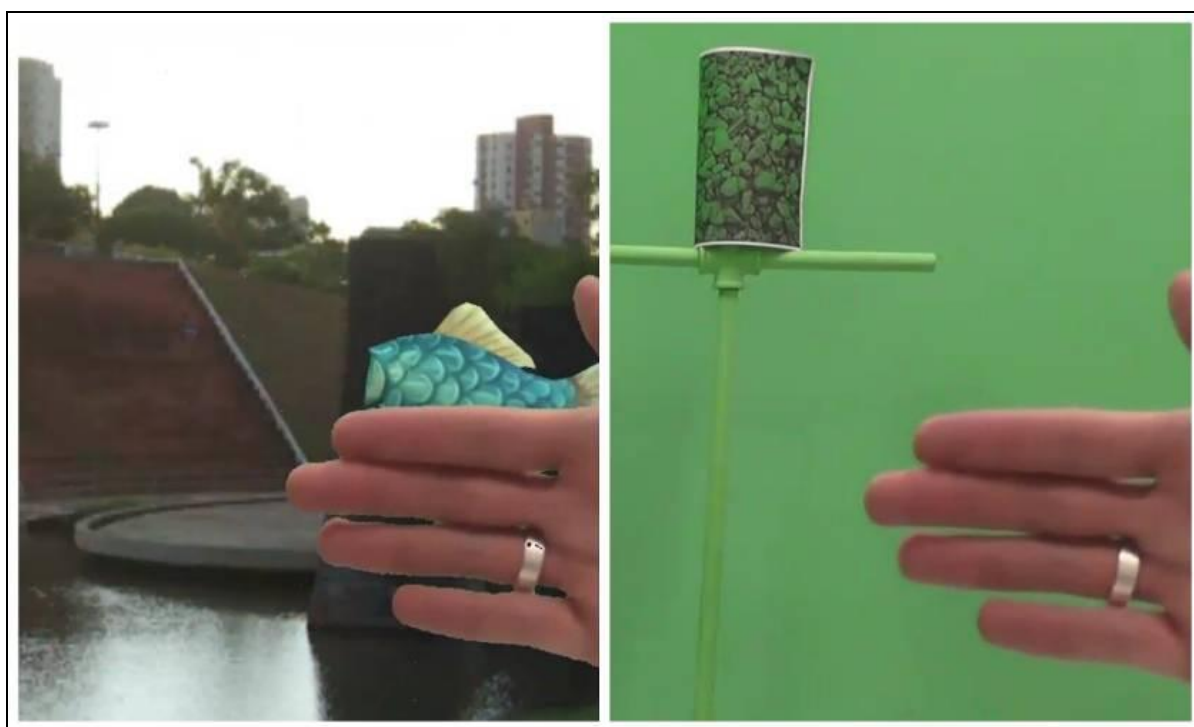


Fonte: Elaborada pelo autor

Experimento 10 - Retirada do marcador por meio da substituição do plano de fundo

O propósito deste experimento foi testar a funcionalidade implementada para retirar o marcador da cena por meio do uso da técnica de *chroma-key*, baseado no método desenvolvido por Sanches (2007), e não pelo método de fixação de marcadores. Desse modo é possível que o marcador não apareça na cena, mas que continue sendo rastreado e, portanto, sujeito a manipulações. A Figura 45 ilustra essa retirada do marcador.

Figura 45 – Retirada do marcador ao substituir o plano de fundo



Fonte: Elaborada pelo autor

Observa-se que o marcador foi gerado com uma tonalidade próxima a cor do fundo (nesse caso, verde), para que no processo de *chroma-key*, o fundo e o marcador pudessem ser retirados e um novo fundo inserido no lugar. Mesmo que o marcador não apareça, ele está sendo rastreado e, portanto, o elemento virtual à ele associado aparece na cena.

Experimento 11 – Multi-marcadores

Este experimento visou testar a funcionalidade da inserção de elementos virtuais na cena utilizando multi-marcadores, marcadores tridimensionais na forma de cubo ou paralelepípedo, onde cada face tem um marcador de textura associado (Figura 46), e utilizar

esse multi-marcador acoplado a uma interface tangível¹⁵ para auxiliar o ator na manipulação de objetos virtuais.

Figura 46 - Multi-marcador cúbico utilizado nos experimentos



Fonte: Elaborada pelo autor

Pode-se observar que este multi-marcador tridimensional, em um formato cúbico, possui somente cinco faces com texturas, onde cada lado do cubo faz referência a um dos lados do elemento virtual. O sexto lado não era necessário, pois estava colado a interface tangível e não apareceria em cena. As Figuras 47 e 48 exemplificam a translação e rotação, respectivamente, de um multi-marcador cúbico, e a consequente transposição do mesmo movimento no elemento virtual. Foi realizado um deslocamento no centro do objeto virtual para que o mesmo coincidissem com o centro do cubo.

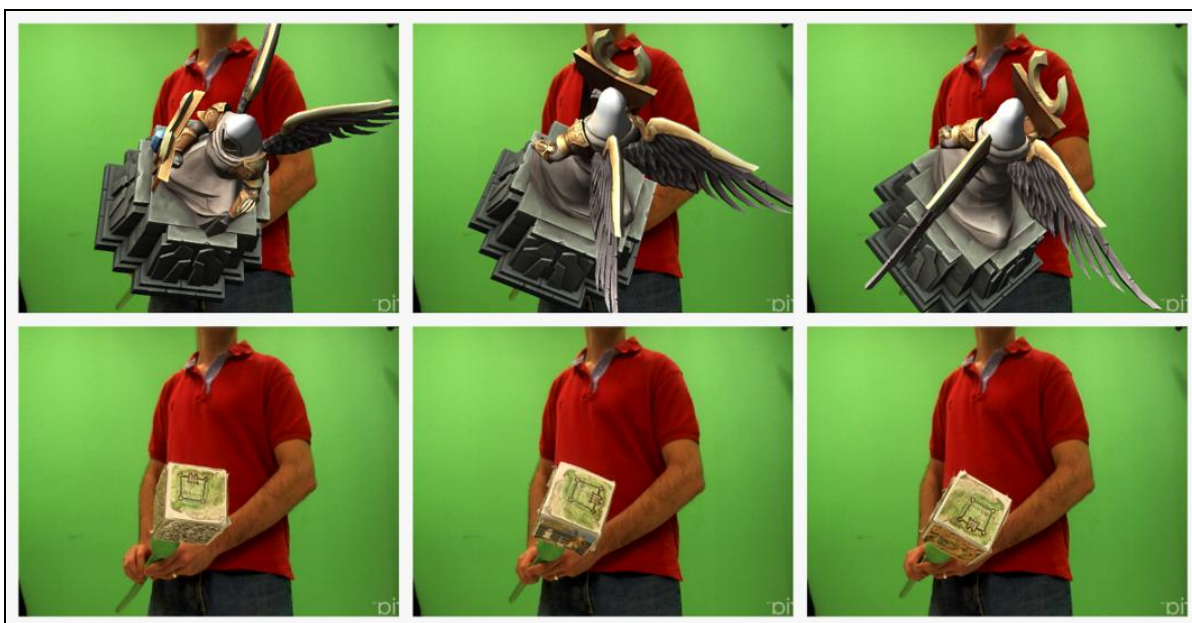
¹⁵ Interfaces que suportam a interação direta entre o mundo físico e o mundo virtual, enfatizando o uso de objetos e ferramentas reais e físicas (AZUMA et al., 2001).

Figura 47 - Experimento com um multi-marcador cúbico, translação de um objeto virtual em cena



Fonte: Elaborada pelo autor

Figura 48 - Experimento com um multi-marcador cúbico, rotação de um objeto virtual em cena



Fonte: Elaborada pelo autor

Dessa forma foi possível manipular objetos virtuais por meio de uma interface tangível. Essa técnica fez com que o ator possa manipular um objeto virtual com naturalidade. Nota-se também que o objeto virtual dispõe de um posicionamento em relação ao centro do cubo, um formato e uma dimensão de maneira a sobrepor o marcador cúbico em sua

totalidade. Isso faz com que não haja um erro na cena aumenta devido ao aparecimento do marcador.

Experimento 12 - Marcador cilíndrico e recurso de sobreposição de camadas

Neste experimento visou-se testar a funcionalidade da utilização de um marcador cilíndrico para melhorar a robustez do rastreo do marcador em cena. O marcador cilíndrico poderá ser composto por uma superfície lateral e dois círculos (base superior e inferior). Neste caso foi utilizada somente a superfície lateral (Figura 49). Esse marcador contém de uma a três texturas de referência. A Figura 50 exemplifica a translação e rotação de um marcador cilíndrico e a consequente transposição do mesmo movimento no elemento virtual. Foi realizado um deslocamento no centro do objeto virtual para que o mesmo coincidissem com o centro do cilindro.

Figura 49 – Marcador cilíndrico utilizado nos experimentos



Fonte: Elaborada pelo autor

Figura 50 - Experimento com um marcador cilíndrico, translação e rotação de um objeto virtual



Fonte: Elaborada pelo autor

Essa técnica permite manipular objetos virtuais por meio de uma interface cilíndrica tangível. Assim como no multi-marcaador cúbico, essa técnica também permite com que o ator possa manipular um objeto virtual como se de fato estivesse segurando, rotacionando e transladando aquele objeto de forma real durante a gravação.

O marcaador cilíndrico preso à uma tiara, posicionada na cabeça do ator, possibilitou que o mesmo movesse e girasse a cabeça de forma a mover e girar o elemento virtual associado. Na cena da Figura 51, assas de morcego com animação, foram posicionadas atrás do ator, utilizando a técnica de renderização por camadas.

Figura 51 - Experimento de um marcador cilíndrico com a técnica de renderização por camadas e o objeto virtual atrás do corpo do ator



Fonte: Elaborada pelo autor

Dessa forma é possível posicionar objetos virtuais em relação ao corpo do ator, de forma que esses objetos sempre aparecem em cena enquanto o marcador estiver em cena, facilitado pelo uso do marcador cilíndrico, por possibilitar que o ator girasse em 360 graus e o marcador continuasse a ser rastreado pela câmera. Nota-se que na Figura 51, o objeto virtual não está mais posicionado no centro do marcador, mas sim posicionado atrás do corpo do ator.

Na Figura 52, o corpo de um peixe virtual (sem cabeça) foi posicionado atrás da cabeça do ator de modo a acompanhar seus movimentos.

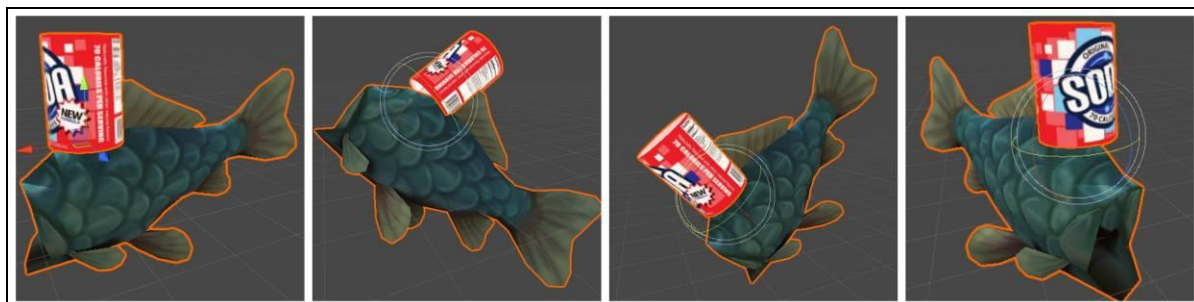
Figura 52 - Experimento de um marcador cilíndrico com a técnica de renderização por camadas e o objeto virtual atrás da cabeça do ator



Fonte: Elaborada pelo autor

Nota-se que na Figura 52, o objeto virtual está sempre atrás da cabeça do ator. Foi possível visualizar a cabeça do ator dentro do corpo do peixe, com maior destaque na primeira e última coluna, devido ao fato do ator estar com um tecido verde ao redor de sua face, possibilitando que somente a face do ator seja visível e o resto do corpo retirado da cena pelo uso da técnica de *chroma-key*. Para que o objeto virtual pudesse ser movimentado e rotacionado de forma correta, foi necessário um ajuste prévio na origem (posição), rotação e escala do objeto virtual em relação à face do ator. A Figura 53 apresenta a mudança de origem do objeto virtual em relação ao marcador e como isso influenciou seu posicionamento na cena.

Figura 53 – Movimentação e rotação de um objeto virtual com origem alterada, em relação um marcador cilíndrico



Fonte: Elaborada pelo autor

Na Figura 54, uma nave espacial foi posicionada de forma que o ator ficasse “dentro” dela. Para haver essa percepção do ator estar dentro, o modelo 3D foi dividido em duas partes, uma parte que ficará sempre à frente e outra que ficará sempre atrás do ator, utilizando a técnica de renderização por camadas, onde as duas partes são associadas ao mesmo marcador (marcador cilíndrico) com uma distância mínima entre elas.

Figura 54 – Elementos virtuais à frente e atrás do ator, criando o efeito de que o mesmo está no interior do objeto virtual



Fonte: Elaborada pelo autor

Experimento 13 - Interação por meio de botões virtuais

O objetivo principal deste experimento foi testar uma forma de interação alternativa do ator com o objeto virtual, além da interação por meio da rotação e translação dos marcadores. O modo apresentado aqui é a possibilidade de o ator realizar a mudança do objeto virtual associado ao marcador, por meio da oclusão de “botões virtuais”. A Figura 55 ilustra duas situações distintas de escolha.

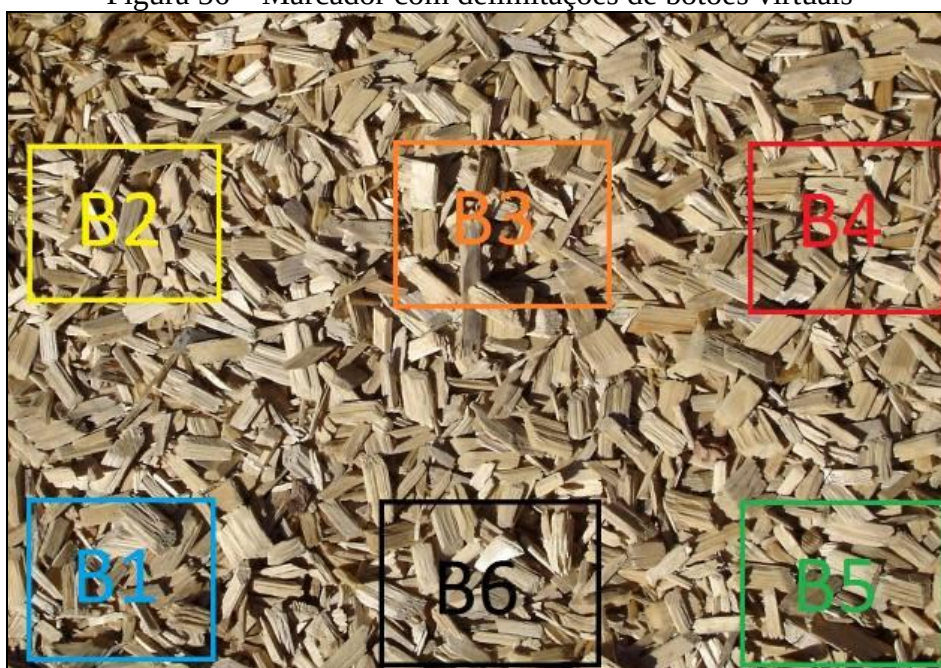
Figura 55 - Utilização de botões virtuais na interação do ator, trocando entre objetos virtuais



Fonte: Elaborada pelo autor

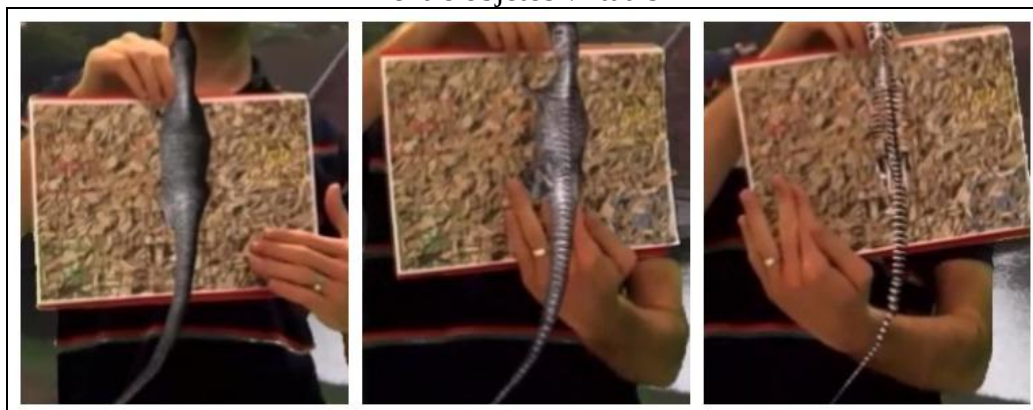
Observa-se que existem dois botões virtuais que aparecem sobre o marcador, em locais pré-definidos. O acionamento de um desses botões, por meio de uma interação entre a mão do ator e a posição do botão virtual, faz com que seja acionada uma troca entre os objetos virtuais ali apresentados. Os botões virtuais que aparecem sobre o marcador são somente para fins ilustrativos de auxílio de posição, os quais podem também ser posicionados de forma transparente. Para que haja um auxílio visual físico com relação à posição dos botões virtuais, foram desenhados e numerados retângulos no marcador, conforme a Figura 56. Na Figura 57 é possível observar um experimento similar, com três botões e objetos virtuais, porém com botões virtuais transparentes sobre o marcador, onde as delimitações dos retângulos/posicionamento dos botões virtuais sobre o marcador serviu de auxílio.

Figura 56 – Marcador com delimitações de botões virtuais



Fonte: Elaborada pelo autor

Figura 57 - Experimento de botões virtuais transparentes na interação do ator com a troca entre objetos virtuais

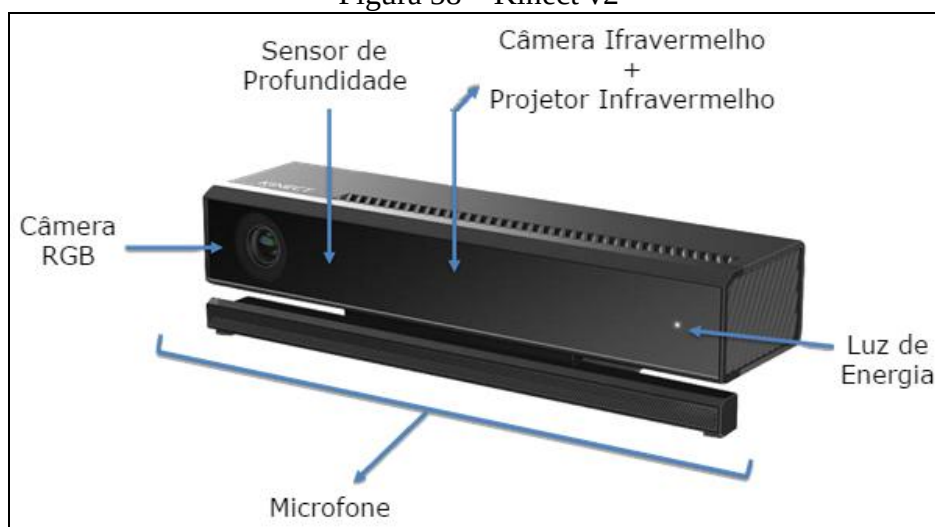


Fonte: Elaborada pelo autor

Experimento 14 - Animação de personagens virtuais por meio de captura de movimentos

Este experimento visou testar a funcionalidade da captura de movimentos¹⁶ (*Human Motion Capture* ou *MoCap*) para realizar a transposição dos movimentos, do ator sendo capturado, a um humanoide virtual, em tempo real. A captura do ator e de seus movimentos é realizada por meio do uso do sensor Kinect v2 (Figura 58), da empresa Microsoft, e de um *plugin* desse sensor para o Unity (MICROSOFT, 2016; MICROSOFT, 2017). Esse sensor possibilita a captura, em tempo real, de 25 juntas do corpo humano, conforme mostrado pela Figura 59 (MSDN MICROSOFT, 2017). A Figura 60 ilustra a captura de movimentos de um ator e a transposição do mesmo em um humanoide virtual.

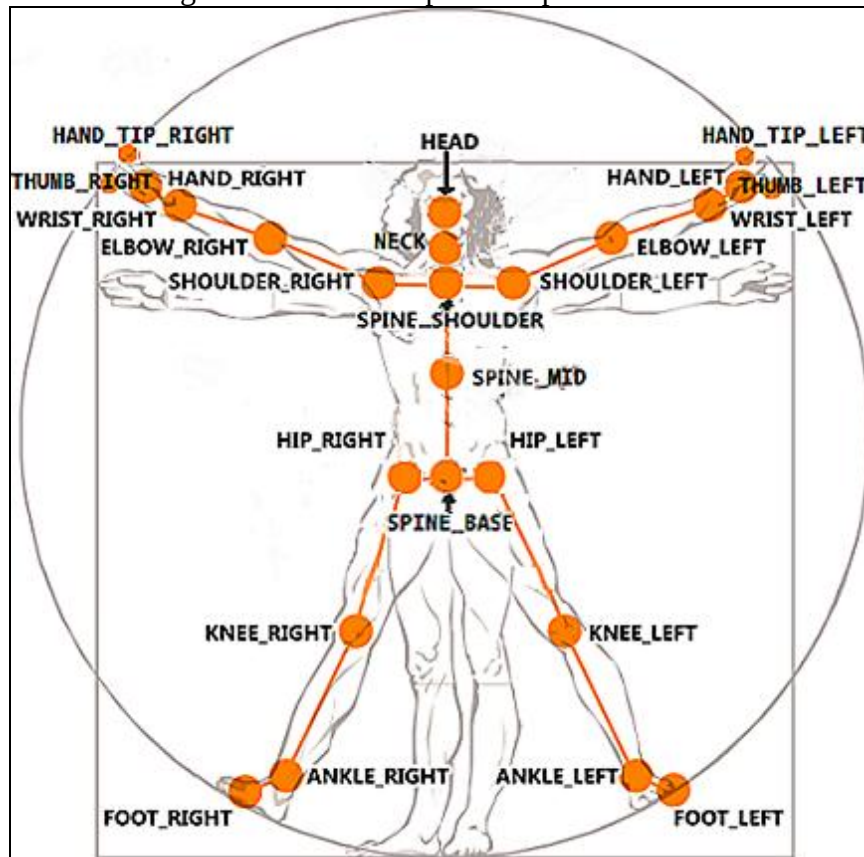
Figura 58 – Kinect v2



Fonte: MICROSOFT, 2016

¹⁶ O processo de análise e registro do corpo humano, e de seu movimento (translação e rotação), em um espaço de tempo (MOESLUND; GRANUM, 2001).

Figura 59 – Juntas capturadas pelo Kinect v2



Fonte: MSDN MICROSOFT, 2017

Figura 60 – Experimento de captura de movimentos e a transposição desses movimentos em um humanoide virtual



Fonte: Elaborada pelo autor

Observa-se que na primeira linha e coluna, o ator e o humanoide virtual estão em uma posição de “T”. Essa posição é necessária no início de cada processo de captura de movimentos, pois o algoritmo de captura, elaborado pelo Vizoni (2017), utiliza como base as translações e rotações a partir dessa posição “T” inicial. Nota-se que o esqueleto verde com os pontos representa as juntas que foram capturadas, naquele *frame*, pelo Kinect v2 e que a transposição da posição e rotação dessas juntas é realizada no humanoide virtual ao lado do ator. Com essa captura e transposição, é possível animar humanoides virtuais em tempo real, durante a gravação de cenas. A Figura 61 ilustra o uso da captura de movimentos em uma cena com dois atores.

Figura 61 – Captura de movimentos e o ator contracenando com um humanoide virtual



Fonte: Elaborada pelo autor

Nessa Figura 61, pode-se observar que existem dois atores, um deles (ator na direita na segunda linha) tendo o seu movimento capturado pelo sensor, o que manipula o movimento do humanoide virtual, e o outro contracenando com o humanoide virtual (primeira linha da Figura 61). Nessa cena, a câmera real está apontada em uma direção e o Kinect v2 em outra para que o ator de captura de movimento não apareça em cena com o outro ator.

Experimento 15 - Uso da técnica HDRI

Uma dificuldade importante na composição entre os elementos reais e virtuais é a concordância da iluminação e das sombras entre o mundo real e virtual (FOURNIER, 1994).

Uma forma de minimizar esse problema é pelo uso de iluminação virtual que coincida com o posicionamento, dimensão, tipo e intensidade da iluminação existente no mundo real (AGUSANTO; CHUANGUI; SING, 2003). Essa iluminação virtual pode ser obtida com o uso da técnica de iluminação baseada em imagens (*Image Based Lighting*¹⁷ - IBL) e pelo uso do método de ampla faixa dinâmica (*High Dynamic Range Imaging*¹⁸ - HDRI) para a geração de um mapa de iluminação (DEBEVEC, 2005).

Nesse experimento, baseado no método desenvolvido pelo Lagarde, Lachambre e Jover (2016), teve como objetivo testar o uso de um mapa de iluminação HDRI no mundo virtual, a partir de um ponto no mundo real. Esse mapa de iluminação foi gerado a partir de fotos capturadas em 360 graus, utilizando a câmera omnidirecional RICOH THETA S, com diversos níveis de exposição. Essas exposições foram compostas e uma imagem final do tipo HDR foi gerada (Figura 62).

Figura 62 – Imagem HDR gerada a partir da câmera omnidirecional



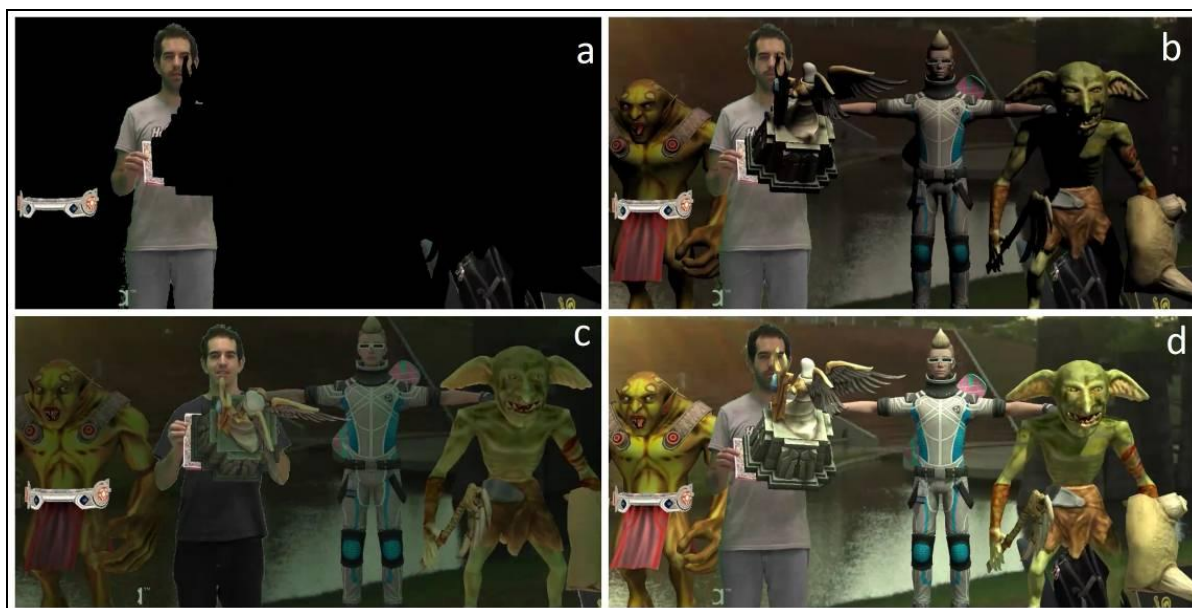
Fonte: Elaborada pelo autor

Essa imagem é então inserida e configurada no Unity, criando assim a iluminação correta no mundo virtual, com referência a aquele ponto e aquela iluminação do mundo real de onde foram capturadas as fotos. A Figura 63 apresenta as diferenças entre a utilização ou não da iluminação HDRI.

¹⁷ Processo de iluminar a cena e objetos virtuais com imagens, panorâmicas, de iluminação do mundo real (DEBEVEC, 2005; REINHARD, et al., 2010).

¹⁸ Processo que envolve capturar uma série de fotos, do ambiente real, com variações nos níveis de exposição e depois criando uma imagem composta que abrange esse intervalo de exposição (DEBEVEC, 2005).

Figura 63 – Experimento de iluminação HDRI: a) cena sem luz virtual e sem iluminação HDRI; b) cena com luz virtual e sem iluminação HDRI; c) cena sem luz virtual mas com iluminação HDRI; d) cena com luz virtual e iluminação HDRI



Fonte: Elaborada pelo autor

A Figura 63 foi dividida em quatro áreas. As áreas apresentam uma cena e seus elementos: a) sem uma fonte de luz virtual e sem iluminação HDRI; b) com uma fonte de luz virtual e sem iluminação HDRI; c) sem uma fonte de luz virtual e com uma iluminação HDRI; d) com uma fonte de luz virtual e com uma iluminação HDRI.

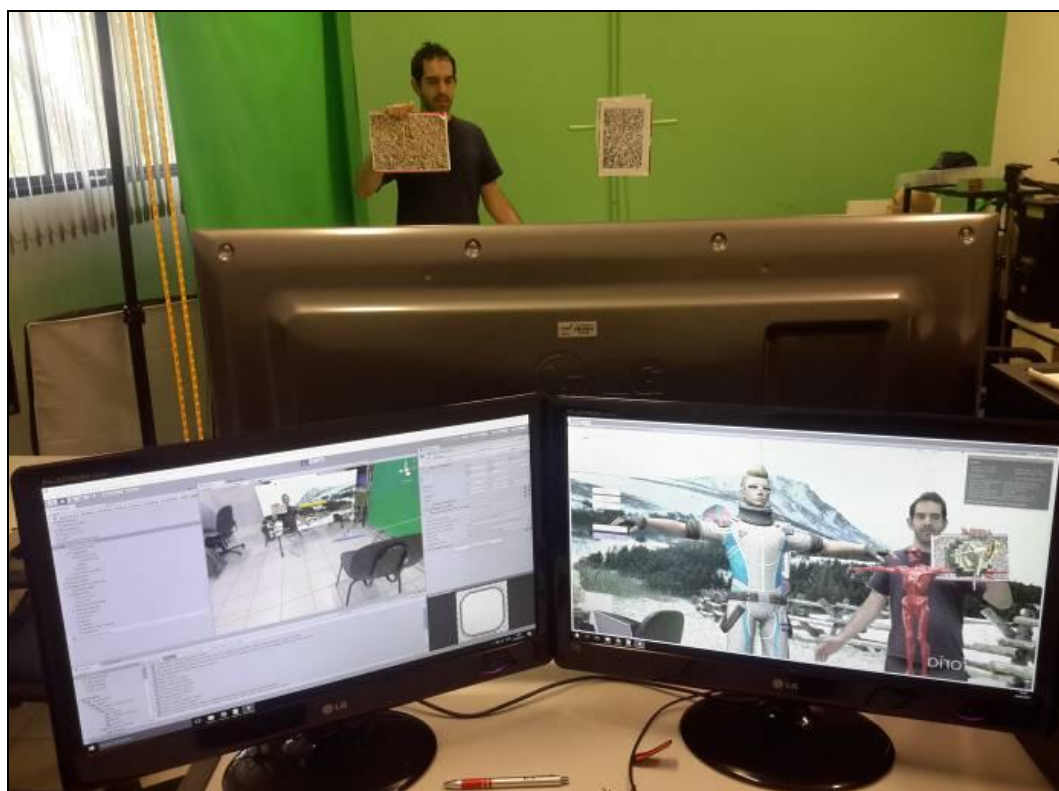
Observando-se a Figura 63 nota-se que com o uso do mapa de iluminação HDRI, é possível ter-se uma iluminação no mundo virtual que se aproxima da iluminação do mundo real. Apesar desse método gerar uma iluminação na cena virtual, observa-se que não existem sombras nos objetos do mundo virtual quando é utilizado somente a iluminação HDRI (Figura 63c). Isso ocorre, porque no Unity a geração de sombras é conseguida pelo uso das luzes virtuais. Portanto, para que possa haver uma iluminação sobre o objeto virtual, próxima a iluminação do mundo real, é necessária a utilização tanto da iluminação HDRI quanto das fontes de luz virtuais, no mundo virtual (Figura 63d).

Experimento 16 - Visualização e entrada por dispositivos externos

Este experimento teve o propósito de testar duas formas adicionais de visualização e de gravação de vídeo aumentado (modo operador de câmera e modo diretor), em alternativa ao uso do modo padrão do sistema, este último mostrado na Figura 64. O primeiro modo, modo operador de câmera, apresenta ao operador tanto o vídeo real, pelo do

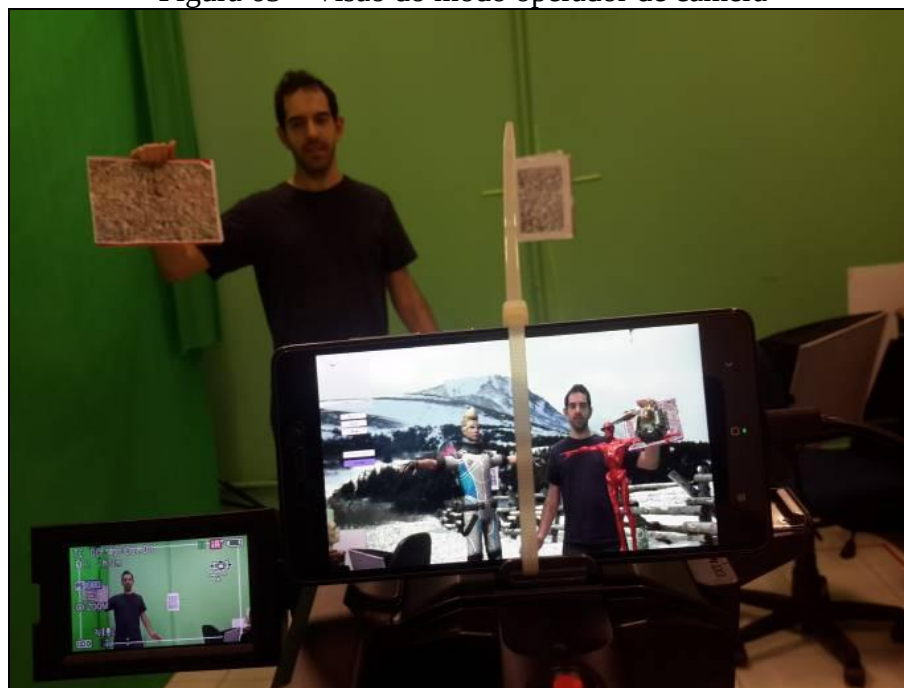
visor da câmera real, quanto o vídeo aumentado, por meio de um *smartphone* (Xiaomi Redmi 3S) acoplado a câmera real, com uma interface para iniciar/encerrar a gravação (Figura 65). O segundo modo, modo diretor, apresenta ao diretor o vídeo aumentado, por meio de um *tablet* (Samsung Galaxy Note 10.1), com uma interface para iniciar/encerrar a gravação (Figura 66). Para a realização deste experimento foi necessária a instalação do aplicativo Unity Remote 5 no *smartphone* e no *tablet* (GOOGLE PLAY, 2017).

Figura 64 – Visão do operador do sistema



Fonte: Elaborada pelo autor

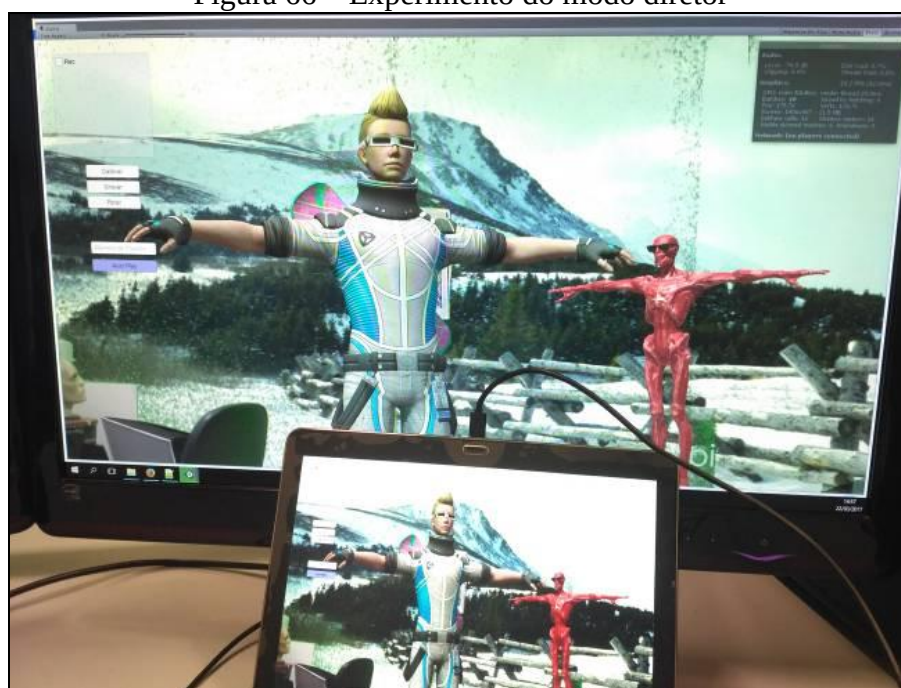
Figura 65 – Visão do modo operador de câmera



Fonte: Elaborada pelo autor

A Figura 65 apresenta a visão que o operador de câmera terá ao operar a câmera real durante a gravação da cena. O operador poderá observar a cena real olhando para o local de gravação, o vídeo real a ser gravado pelo visor da câmera e o vídeo aumentado pelo *smartphone* acoplado a câmera real.

Figura 66 – Experimento do modo diretor



Fonte: Elaborada pelo autor

A Figura 66 apresenta a visão que o operador do sistema terá, no monitor do computador, e a visão do modo do diretor, no *tablet*. Observe-se que as visualizações do operador do sistema, operador de câmera e do diretor são espelhadas. Isso ocorre devido à utilização da visão espelhada (invertida), do vídeo aumentado, para o retorno ao ator (Figura 67).

Figura 67 – Visão do ator com o uso do retorno espalhado ao ator



Fonte: Elaborada pelo autor

Experimento 17 – Transmissão ao vivo

Nos últimos anos, houve um aumento pelo interesse por serviços de transmissão ao vivo (*streaming*¹⁹) na Internet. Os sistemas de transmissão ao vivo de vídeo gerados por usuários (como o Twitch e o YouTube Live) são serviços que permitem que qualquer pessoa transmita um fluxo de vídeo pela Internet (PIRES; SIMON, 2015).

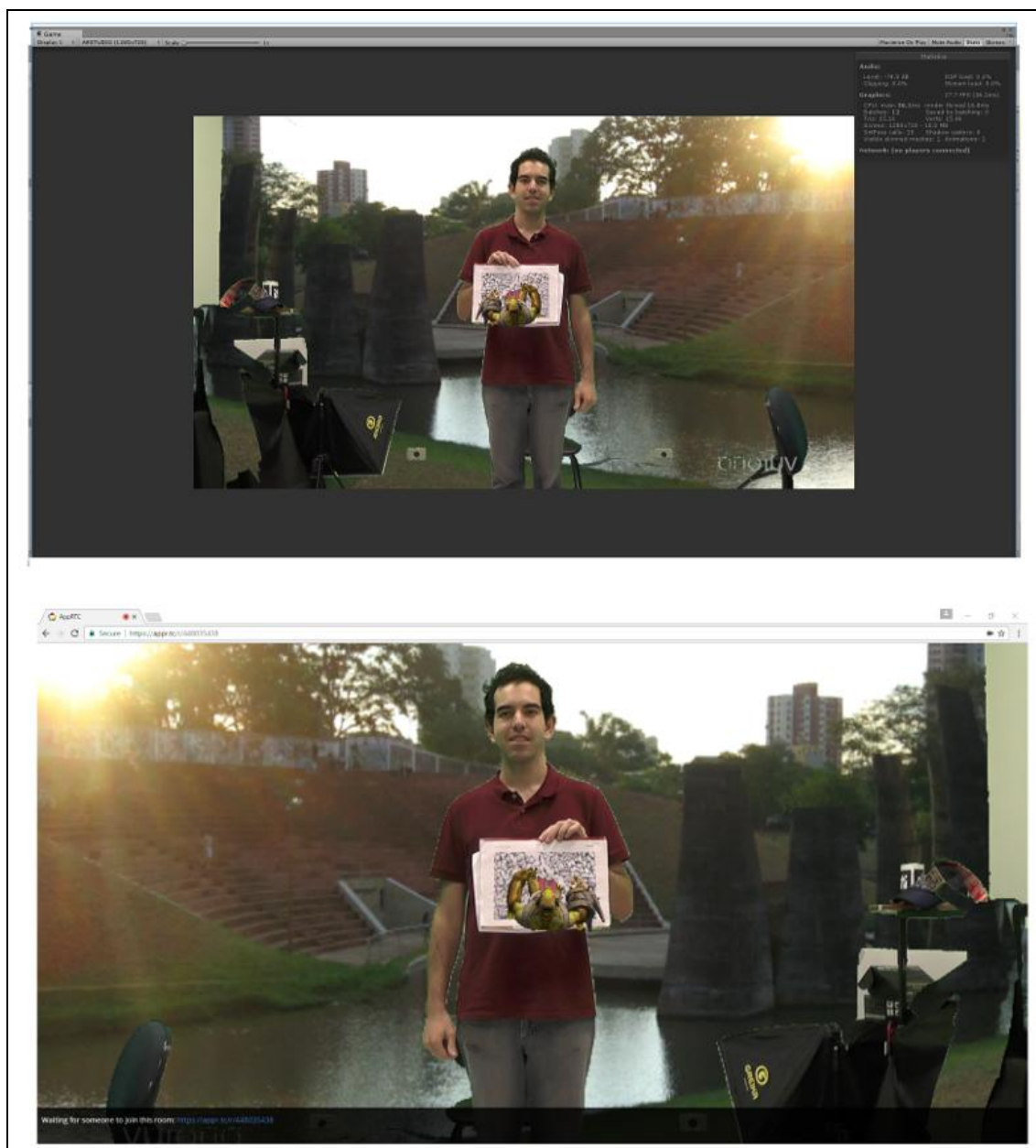
Em 2016, o Twitch, serviço online de transmissão ao vivo da empresa Amazon, teve mais de 100 milhões de usuários. Esses usuários assistiram 292 bilhões de minutos de conteúdo, 35% deles por dispositivos móveis, a uma média de 1 hora e 46 minutos por dia. Havia mais de 2.2 milhões de usuários criadores/transmissores de conteúdo (TWITCH, 2017).

Com este contexto, o intuito deste experimento foi testar a transmissão ao vivo do vídeo aumentado, gerado em tempo real pelo ARSTUDIO 2.0, na internet. O Unity não possui, nativamente, recurso para a transmissão de conteúdo através de sites de transmissão de vídeo ao vivo. Para realizar essa transmissão, foi utilizado o algoritmo “UnityCam”

¹⁹ Método de transmissão de um arquivo de mídia em um fluxo contínuo de dados que podem ser lidos antes da transmissão completa do arquivo (GOLDSTEIN et al., 1999; PRITCH et al., 2007).

desenvolvido por Saraiji (2017). Esse algoritmo gera uma câmera virtual que transmite o vídeo aumentado como se fosse uma câmera conectada ao computador que estivesse capturando o mundo real. Na Figura 68, é ilustrado o vídeo aumentado no Unity e um navegador utilizando a câmera virtual “UnityCam” para transmitir, pela internet, o vídeo gerado pelo Unity.

Figura 68 – Transmissão ao Vivo pelo ARSTUDIO 2.0



Fonte: Elaborada pelo autor

A transmissão de conteúdo via internet, proveniente de uma câmera, é sujeito a uma taxa de *frames* por segundo (FPS) inconstante (GOLDSTEIN et al., 1999; PRITCH et al.,

2007). No teste de transmissão foi utilizado um marcador plano, um objeto virtual e a aplicação do *chroma-key*, a uma taxa de 27,7 FPS. Uma carência que esse modo de transmissão ao vivo possui é a falta de transmissão do áudio da cena virtual, podendo transmitir somente o áudio capturado pelo microfone. Uma possível solução seria a geração de uma fonte de áudio virtual dentro do Unity.

Experimento 18 – Pós-Produção no ARSTUDIO 2.0

A finalidade deste experimento foi de testar o modo de pós-produção no estúdio virtual, onde o processamento não necessitaria ser realizado em tempo real.

O armazenamento do fluxo de vídeo bruto, com áudio, se dá por meio de dois modos. O primeiro modo, compatível com qualquer câmera de entrada, é executado dentro do sistema, onde ao invés de armazenar o vídeo aumentado durante a gravação é realizado o armazenamento do vídeo bruto. O segundo modo, compatível somente com a câmera profissional, armazena o vídeo bruto no cartão de memória da câmera durante a gravação. O armazenamento das informações de configuração é feito pelo armazenamento da cena virtual e de suas configurações em arquivos no formato interno de cenas do Unity.

Após o armazenamento do vídeo bruto e das informações de configuração, foi necessário criar um modo de entrada e reprocessamento dos mesmos pelo sistema. Para que o Vuforia processasse o vídeo bruto armazenado, primeiro foi necessária a conversão do vídeo para o formato Ogg Theora, por ser o formato padrão de vídeo utilizado pelo Unity (XIPH, 2017; UNITY3D, 2017b). Depois da conversão, o vídeo bruto serviu de entrada para o Vuforia. A entrada das informações de configuração ao sistema foi realizada pela importação da cena do Unity anteriormente criada.

O problema que esse modo apresenta, em seu estado atual, é a falta de exportação/importação das ações realizadas via a interface de usuário durante a produção. Somente as ações/interações realizadas pelo ator ou aquelas pré-programadas por código causarão modificações na cena durante o modo de pós-produção.

5.2.2 EXPERIMENTOS OBJETIVOS

Como os objetivos específicos foram de verificar a viabilidade de usar um motor de jogos como base de um sistema virtual e a viabilidade de interação e visualização no mesmo por meio da RA, foi necessário realizar uma avaliação do desempenho do sistema virtual desenvolvido. Para que o desempenho do ARSTUDIO 2.0 pudesse ser avaliado, além

dos experimentos das funcionalidades descritos na subseção 6.2.1, foram realizados mais 24 experimentos.

Existem diversos sistemas de vídeo de televisão (analógico e digital) e cada sistema tem o seu padrão de FPS. Alguns sistemas de vídeos de televisão existentes são: Comitê Nacional dos Sistemas de Televisão (National Television Systems Committee - NTSC), Linha de Fase Alternada (Phase Alternating Line - PAL), Cor Sequencial com Memória (Séquentiel Couleur à Mémoire - SECAM), Transmissão de Vídeo Digital (Digital Video Broadcasting - DVB), Comitê do Sistema Avançado de Televisão (Advanced Television Systems Committee - ATSC), Serviço Integrado de Transmissão Digital Terrestre (Integrated Services Digital Broadcasting - ISDB) e Sistema Brasileiro de Televisão Digital (SBTVD). No cinema a Sociedade de Engenheiros de Cinema e Televisão²⁰ (Society of Motion Picture and Television Engineers - SMPTE) é quem estabelece o padrão de taxa de FPS (BEDICKS, et al., 2006; JACK, 2007; STANDARD, 2011; LANKINEN, et al., 2014; PFEIFFER; GREEN, 2015; SMPTE, 2017).

A Tabela 1 exibe os padrões de taxas de *frames* utilizados em nesses sistemas.

Tabela 1 – Taxas de *frames* por segundo

Sistema de Vídeo	Taxa de <i>Frames</i> por Segundo
SECAM	24
Cinema	24
DVB	24 a 30
ATSC	24 a 30
ISDB	24 a 30
SBTVD	24 a 30
PAL	24 ou 25
NTSC	29,97
PAL-M	29,97

Fonte: Elaborada pelo autor

Com esse contexto, nesses experimentos os dados de resolução e FPS, da imagem composta, foram obtidos. No aspecto de FPS, dois modos foram avaliados, o primeiro pela quantidade durante a execução do sistema sem a exportação do vídeo aumentado em forma de arquivo de vídeo (FPS Execução), e o segundo pela quantidade no arquivo de vídeo exportado

²⁰ Uma associação profissional que estabelece padrões, práticas e diretrizes para a indústria cinematográfica e de televisão, incluindo o áudio que acompanha dos vídeos.

(FPS Arquivo). Para a avaliação o valor mínimo aceitável é com base no padrão de cinema de uma taxa de 24 FPS.

Esses experimentos foram divididos em três classificações de captura: Webcam 1080p, onde se utilizou a webcam como a câmera de captura a uma resolução de 1920 por 1080 pixels (1080p); Webcam 720p, onde se utilizou a webcam como a câmera de captura a uma resolução de 1280 por 720 pixels (720p); e Câmera Profissional 720p, onde se utilizou a câmera profissional como a câmera de captura a uma resolução de 1280 por 720 pixels (720p). Dentro dessas classificações de captura, foram avaliados quatro cenários de testes. As características específicas de cada um dos cenários são apresentadas na Tabela 2.

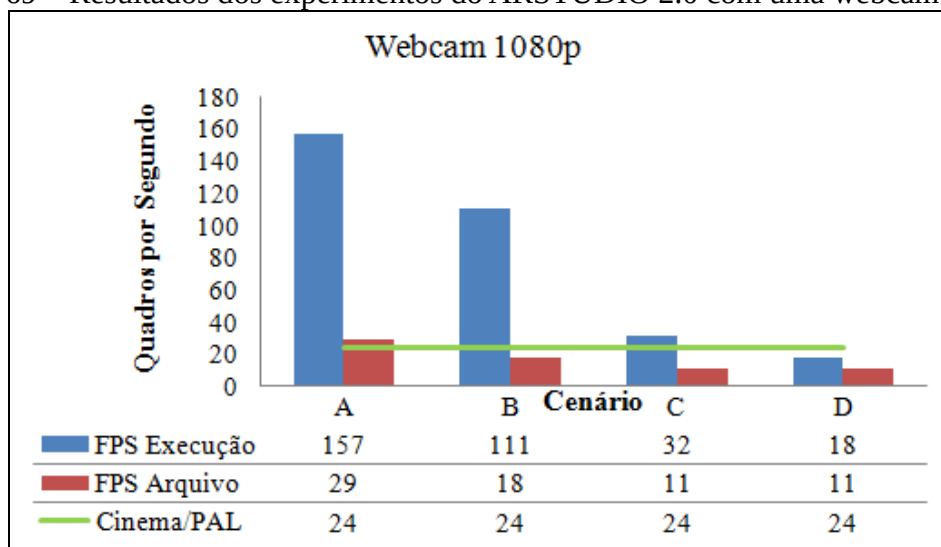
Tabela 2 – Cenários de testes e suas características principais

CARACTERÍSTICAS	CENÁRIO A	CENÁRIO B	CENÁRIO C	CENÁRIO D
Chroma-key	Não utilizado	Imagem de fundo com resolução 2048 x 1024 pixels	Não utilizado	Imagem de fundo com resolução 2048 x 1024 pixels
Luz virtual direcional	1 fonte	1 fonte	1 fonte	1 fonte
Iluminação HDRI	Não utilizada	Não utilizada	Não utilizada	Utilizada
Rastreamento de marcadores	Não utilizado	1 multi-marcador	1 marcador plano	1 marcador plano, 1 multi-marcador e 1 marcador cilíndrico
Renderização por camadas	Não utilizada	Não utilizada	Não utilizada	Utilizada
Objetos Virtuais	Não utilizado	2173 vértices e 2066 faces	5643 vértices e 7978 faces	77228 vértices e 21772 faces
Áudio	Não utilizado	Capturado do microfone	Capturado do microfone	Capturado do microfone e 5 áudios da cena virtual
Captura de Movimentos	Não utilizado	Não utilizado	Utilizado	Utilizado

Fonte: Elaborada pelo autor

As Figuras 69, 70 e 71 ilustram os resultados dos experimentos da Webcam 1080p, Webcam 720p e da Câmera Profissional 720p, respectivamente para cada um dos 4 cenários.

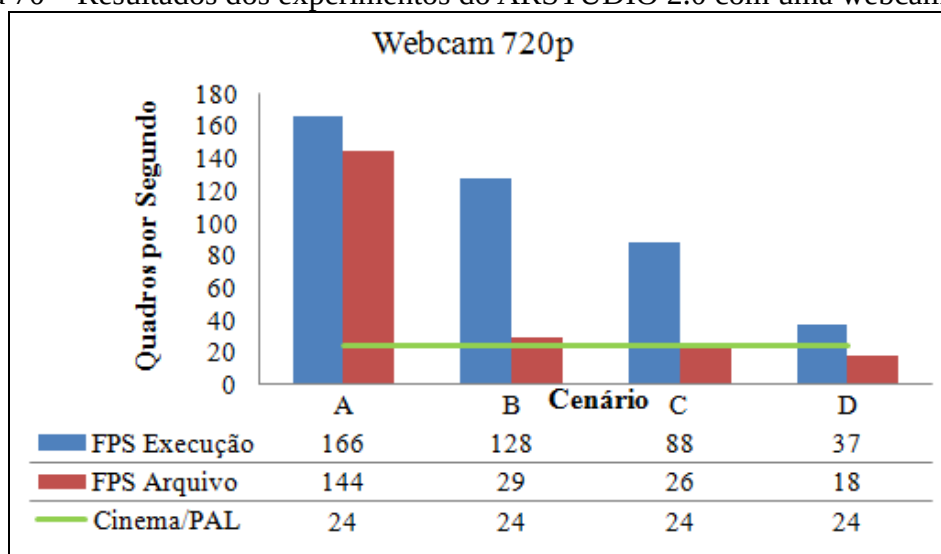
Figura 69 – Resultados dos experimentos do ARSTUDIO 2.0 com uma webcam a 1080p



Fonte: Elaborada pelo autor

Analisando os resultados exibidos no gráfico da Figura 69 é possível observar que houve uma queda significativa entre a taxa de FPS durante a execução do sistema (sem a finalidade de gerar um arquivo de vídeo) e do arquivo de vídeo gerado. Nas taxas de FPS Execução, todos, exceto o cenário D, obtiveram taxas aceitáveis (24 ou mais). Nas taxas de FPS dos arquivos gerados, somente o cenário A teve resultado aceitável, os demais foram todos abaixo dos 24 FPS. No cenário D, o cenário que mais demandou do processamento, teve um resultado baixo de 18 e 11 FPS nos dois modos. Devido a essas baixas taxas de FPS dos arquivos gerados, foram realizados experimentos a 720p de resolução, cujos resultados são exibidos na Figura 70.

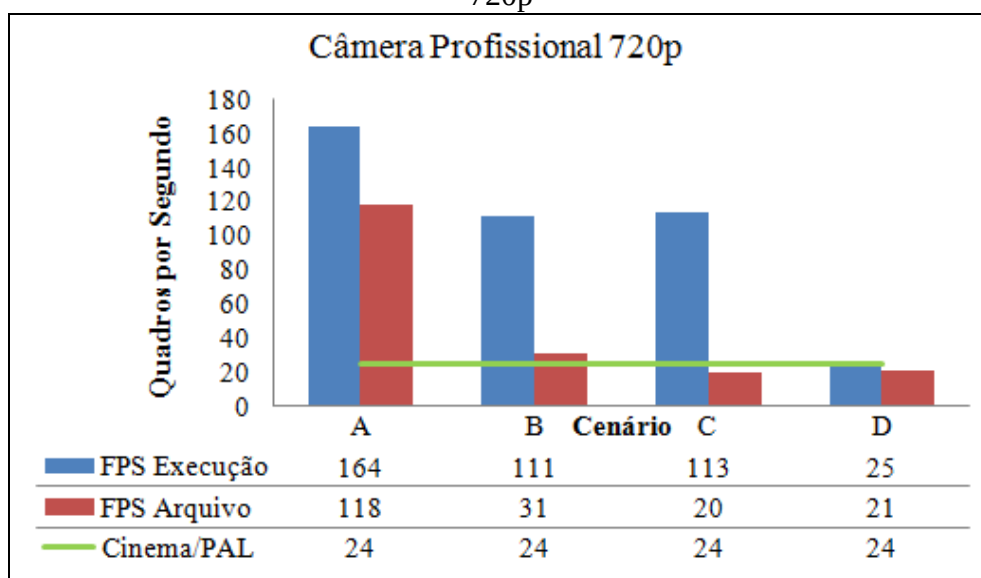
Figura 70 – Resultados dos experimentos do ARSTUDIO 2.0 com uma webcam a 720p



Fonte: Elaborada pelo autor

Analizando os resultados na Figura 70 é possível observar que a taxa de FPS, em ambos os modos aumentaram. A taxa de FPS na execução (sem salvamento em arquivo) foi acima de 24 FPS em todos os cenários. O cenário D teve um aumento significativo nos resultados chegando a 37 e 18 FPS nos dois modos. Com essa melhora nos resultados, foram realizados experimentos com a câmera profissional a 720p (Figura 71).

Figura 71 – Resultados dos experimentos do ARSTUDIO 2.0 com uma câmera profissional a 720p



Fonte: Elaborada pelo autor

Analizando os resultados na Figura 71 é possível observar que a taxa de FPS Execução teve resultados acima de 24 em todos os casos. No modo FPS Arquivo, os cenários A e B tiveram taxas acima de 24 FPS, enquanto nos cenários C e D tiveram taxas abaixo.

Analizando os resultados como um todo, é possível concluir que a resolução da imagem de entrada teve um impacto na taxa de FPS, tanto no modo FPS Execução quanto no modo FPS Arquivo. O uso da RA teve impacto na taxa de FPS, pelo fato do computador ter que processar a cada *frame* a imagem da câmera a procura dos marcadores. O uso da captura de movimento também teve impacto na taxa de FPS. A uma captura a 720p, os resultados apontam que é viável a utilização de um motor de jogos como base de um estúdio virtual que tem a RA como forma de interação e visualização em tempo real, devido ao fato dos resultados apresentarem taxas acima de 24 FPS no modo FPS Execução. No aspecto de geração de conteúdo, exportação do vídeo gerado em forma de arquivo de vídeo, é possível atingir valores aceitáveis quando não se utiliza processamento elevado, como nos cenários C e D.

5.3 COMPARAÇÃO ENTRE ESTÚDIOS VIRTUAIS

O estúdio virtual ARSTUDIO 2.0 mostrou-se bastante interessante, uma vez que permite a geração, visualização e armazenamento de conteúdo midiático com o uso de um motor de jogos e da RA, por ser um sistema de relativamente baixo custo e por apresentar características distintas dos outros estúdios virtuais encontrados na literatura.

Uma comparação entre as características do estúdio virtual ARSTUDIO 2.0 e do estúdio virtual ARSTUDIO 1.0 é apresentada no Quadro 2.

Quadro 2 – Características dos ARSTUDIO 1.0 e do ARSTUDIO 2.0

	ARSTUDIO 1.0	ARSTUDIO 2.0
Ano	2015	2017
Renderização	3D	3D
Realidade Aumentada	ARToolKit	Vuforia
Uso de motor de jogos	Não	Sim
Pré-visualização da cena no estúdio virtual	Sim	Sim
Exportação do vídeo aumentado em forma de vídeo	Sim	Sim
Exportação do vídeo aumentado em forma de imagem	Não	Sim
Exportação do vídeo aumentado em forma de OpenEXR	Não	Sim
Chroma-key	Sim	Sim
Chroma-key de duas aplicações	Não	Sim
Retorno para o ator	Sim	Sim
Inserção de objetos virtuais pela Realidade Virtual	Não	Sim
Uso de arquivo de áudio dentro do sistema	Não	Sim
Mixagem de áudio dentro do sistema	Não	Sim
Interação do ator com objetos virtuais	Sim	Sim
Interação por meio de controle de videogame	Não	Sim
Marcadores fiduciais	Sim	Sim
Marcadores de textura	Não	Sim
Multi-marcador cúbico	Sim	Sim
Marcador cilíndrico	Não	Sim
Marcador com botão virtual	Não	Sim
Fixação de marcadores	Sim	Sim
Oclusão mútua por marcadores	Sim	Sim
Oclusão mútua por sensor de profundidade	Sim	Não
Captura de movimentos para animação de personagens virtuais	Não	Sim
Iluminação virtual HDRI	Não	Sim
Integração do sistema com tablets e smartphones	Não	Sim
Transmissão ao Vivo pela Internet	Não	Sim
Modo Pós-Produção	Não	Sim

Fonte: Elaborada pelo autor

Algumas das principais características que o ARSTUDIO 2.0 difere do seu antecessor são os usos de: um motor de jogos como a base gráfica do sistema, mixagem de

áudio dentro do sistema, marcadores cilíndricos, captura de movimentos para a animação de personagens virtuais, iluminação HDRI, a integração do sistema com *tablets* e *smartphones*, a transmissão do vídeo aumentado ao vivo pela internet, um modo de pós-produção.

Uma comparação entre as características do estúdio virtual ARSTUDIO 2.0 e de outros encontrados na literatura também foi realizada e é apresentada no Quadro 3.

Quadro 3 – Características dos estúdios virtuais encontrados na literatura e do ARSTUDIO

2.0

	Synthevision	RTFX	Goussencourt & Bertolini (2015)	Dreamspace	ARSTUDIO 1.0	ARSTUDIO 2.0
Ano	1989	2011	2015	2016	2015	2017
Renderização	2D	3D	3D	3D	3D	3D
Realidade Aumentada	Não	Não	Não	Sim	Sim	Sim
Rastreamento de câmeras	Sim	Não	Sim	Sim	Não	Não
Retorno ao ator	Não	Sim	Sim	Sim	Sim	Sim
Mixagem de áudio	Não	Não	Não	Não	Não	Sim
Iluminação HDRI	Não	Não	Sim	Não	Não	Sim
Iluminação automática/ correspondente entre o mundo real e o virtual	Não	Não	Não	Sim	Não	Não
Captura de movimentos	Não	Sim	Sim	Sim	Não	Sim
Pré-visualização da cena	Sim	Sim	Sim	Sim	Sim	Sim
Oclusão mútua por sensor de profundidade	Não	Não	Sim	Sim	Sim	Não
Motor de jogos	Não	Sim	Sim	Sim	Não	Sim
Dispositivos conectados via rede	Não	Sim	Não	Sim	Não	Não
Interação do ator com objetos virtuais	Não	Não	Sim	Sim	Sim	Sim
Entrada por dispositivos externos	Não	Sim	Não	Sim	Não	Sim
Transmissão via Internet	Não	Não	Não	Não	Não	Sim
Modo Pós-Produção	Não	Não	Não	Não	Não	Sim
Sistema comercial	Sim	Não	Não	Não	Não	Não

Fonte: Elaborada pelo autor

Algumas funcionalidades que o ARSTUDIO 2.0 ainda não possui são: o rastreamento da câmera real; a alteração de forma automática das luzes nos ambientes real e virtual; oclusão mútua por meio de um sensor de profundidade e o uso da rede para conectar com outros dispositivos. O ARSTUDIO 2.0 se destaca com relação a algumas características dos estúdios virtuais encontrados na literatura, tais como: no uso da RA, poucos outros estúdios utilizam desse recurso para a visualização de e interação com objetos virtuais; a mixagem de áudio dentro do sistema; nas manipulações do sistema por meios além do computador principal (mouse e teclado), como nos usos do *tablet*, *smartphone* e do controle de videogame; na transmissão do vídeo aumentado ao vivo pela internet; e em um modo de pós-produção.

CAPÍTULO 6 – CONCLUSÃO

Este trabalho apresentou o desenvolvimento e teste do sistema ARSTUDIO 2.0, um sistema de estúdio virtual de baixo custo, que utiliza técnicas de realidade aumentada, a qual permite a criação de cenas aumentadas compostas por elementos reais e virtuais, para a geração de conteúdo midiático baseado no motor de jogos Unity 3D.

Resumidamente, as funcionalidades implementadas no ARSTUDIO 2.0 incluem: a visualização e interação com a cena aumentada renderizada em tempo real; a transmissão, ao vivo, via internet da cena aumentada; a exportação da cena aumentada; e o reprocessamento da mesma dentro do próprio sistema. Com os resultados obtidos, é possível observar como a adição de elementos virtuais e efeitos especiais, em tempo real, permitem que o usuário tenha uma prévia de como o vídeo ficará após a pós-produção e corrigir problemas durante a fase de geração e gravação de conteúdo. A inovação do sistema é em otimizar a linha de produção da geração de conteúdo midiático com elementos virtuais e efeitos especiais.

Nesta nova estrutura, usando as técnicas da realidade virtual e aumentada, animações, objetos e cenários virtuais podem ser vistos e manipulados por atores, operadores de câmera e diretores no curso da produção, otimizando assim a cadeia de produção de conteúdos midiáticos.

Os experimentos realizados, tanto os com inspeção visual quanto os quantitativos, indicaram a viabilidade das funcionalidades implementadas, levando-se em consideração as características dos estúdios virtuais do estado da arte, apesar desses últimos terem sido projetados para dar suporte para grandes produções, as quais dispõem, normalmente, de equipamentos e infraestruturas profissionais.

Neste aspecto, a maior contribuição deste trabalho consistiu em demonstrar que os motores de jogos e as bibliotecas de RA atuais possuem capacidade suficiente para servirem de base para a construção de estúdios virtuais com desempenho em tempo real. Esta capacidade viabiliza, por exemplo, a produção de conteúdo ao vivo com um mínimo de atividades de pós-produção. Outra contribuição do trabalho diz respeito a viabilidade no uso de equipamentos convencionais (além da possibilidade de equipamentos profissionais), tais como webcams e câmeras com sensores de profundidade projetadas para uso em jogos (como o Kinect v2, da Microsoft), em estúdios virtuais. Essa é uma característica do ARSTUDIO 2.0 que pode ser de interesse de estações de televisão universitárias, produtoras e usuários domésticos.

Como trabalhos futuros, novas funcionalidades podem ser adicionadas ao ARSTUDIO, tais como: o uso de informações de profundidade obtidas de um sensor de profundidade para a realização da oclusão mútua, entre elementos reais e virtuais, e a segmentação do plano de fundo; o uso de efeitos de partículas (fumaça, fogo, neve, chuva, entre outros) para aumentar a abrangência do sistema com os efeitos especiais; o aperfeiçoamento da interface de usuário no controle do sistema; implementação do *matting* digital; entradas ao sistema por outros dispositivos via rede; a captura das ações realizadas via a interface de usuário durante a fase de produção; a captura do grafo de cena que compõe a cena aumentada, bem como os objetos virtuais e suas animações, posições, escalas e rotações em todos os *frames* do vídeo capturado; e exportar essas informações obtidas para vários softwares que são usados tradicionalmente na etapa de pós-produção, como o Maya, o 3DS Max, Nuke, Houdini, After Effects e outros.

REFERÊNCIAS

- ADOBE AFTER EFFECTS, **Help and Tutorials**, 2013. Disponível em: <http://help.adobe.com/archive/en/after-effects/cs6/after_effects_reference.pdf>. Acesso em: 20 de abr. 2017.
- AGUSANTO, K., LI, L.; CHUANGUI, Z.; SING, N. W. Photorealistic rendering for augmented reality using environment illumination. In: **Mixed and Augmented Reality**, 2003. Proceedings. The Second IEEE and ACM International Symposium on. IEEE, 2003. p. 208-216.
- AZUMA, R. T. A survey of augmented reality. **Presence: Teleoperators and virtual environments**, v. 6, n. 4, p. 355-385, 1997.
- AZUMA, R.; BAILLOT, Y.; BEHRINGER R.; FEINER, S.; JULIER, S.; MACINTYRE, B. Recent advances in augmented reality. **IEEE computer graphics and applications**, v. 21, n. 6, p. 34-47, 2001.
- BAILLOT, Y.; DAVIS, L.; ROLLAND, J. A survey of tracking technology for virtual environments. **Fundamentals of wearable computers and augmented reality**, p. 67, 2001.
- BAJURA, M.; NEUMANN, U. Dynamic registration correction in video-based augmented reality systems. **IEEE Computer Graphics and Applications**, v. 15, n. 5, p. 52-60, 1995.
- BARATOFF, G.; NEUBECK, A.; REGENBREHCT, H. Interactive multi-marker calibration for augmented reality applications. **Proceedings of the 1st International Symposium on Mixed and Augmented Reality**, p. 107, 2002.
- BARNWELL, Jane. **The Fundamentals of Film Making**. 1 ed. Lausanne: AVA Publishing, 2008. 208 p.
- BEDICKS, G.; YAMADA, F.; SUKYS, F.; DANTAS, C. E. S.; RAUNHEITTE, L. T. M.; AKAMINE, C. Results of the ISDB-T system tests, as part of digital TV study carried out in Brazil. **IEEE Transactions on Broadcasting**, v. 52, n. 1, p. 38-44, 2006.
- BILLINGHURST, M.; GRASSET, R.; LOOSER, J. Designing augmented reality interfaces. In: **SIGGRAPH Computer Graphics**. v. 39, n. 1, p. 17-22, 2005.
- BLONDÉ, L.; BUCK, M.; GALLI, R.; NIEM, W.; PAKER, Y.; SCHMIDT, W.; THOMAS, G. A Virtual Studio for Live Broadcasting: The Mona Lisa Project. In: **IEEE Multimedia**, v. 3, n. 2, p. 18-29, 1996.
- BREEN, D.; WHITAKER, R. T.; ROSE, E.; TUCERYAN, M. Interactive occlusion and automatic object placement for augmented reality. In: **Computer Graphics Forum**. Blackwell Science Ltd, v. 15, n. 3, p. 11-22, 1996.
- BUTZ, A.; HÖLLERER, T.; FEINER, S.; MACINTYRE, B.; BESHES, C. Enveloping Users and Computers in a Collaborative 3D Augmented Reality. **Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality**. p. 35, 1999.
- CAMBRIDGE DICTIONARY. Disponível em: <<http://dictionary.cambridge.org/dictionary/english/humanoid>>. Acesso em: 18 de abr. 2017.

CAMPOS, G. M.; MUKUDAI, L. M.; IWAMURA, V. S.; SEMENTILLE, A. C. Sistema de Composição de Estúdios Virtuais Utilizando Técnicas de Realidade Aumentada. **Proceedings - XII Symposium on Virtual and Augmented Reality - SVR 2010**, v. 1, p. 22-30, 2010.

CHUANG, Y. Y.; AGARWALA, A.; CURLESS, B.; SALESIN, D. H.; SZELISKI, R. Video matting of complex scenes. **Proceedings of the 29th annual conference on Computer graphics and interactive techniques**. v. 21, n. 3, p. 243-248, 2002.

CHUANG, Y. Y.; CURLESS, B.; SALESIN, D. H.; SZELISKI, R. A Bayesian approach to digital matting. In: **Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition**. v. 2, p. 264-271, 2001.

COMPARATO, Doc. **Da Criação ao Roteiro: teoria e prática**. São Paulo: Summus, 2000. 486 p.

DAWSON, Sean. **Unity3D Chroma Key Shader**, 2014. Disponível em: <<https://unexpectedend.wordpress.com/2014/11/04/unity3d-chroma-key-shader/>>. Acesso em: 20 de ago. 2015.

DEBEVEC, P. Image-based lighting. In: **ACM SIGGRAPH 2005 Courses**. ACM, 2005. p. 3.

DOBBINS, P. **3D Rendering in Computer Graphics**. 1 ed. Delhi: White Word Publications, 2012. 108 p.

FIALA, M. ARTag, a fiducial marker system using digital techniques. In: **Computer Vision and Pattern Recognition, IEEE Computer Society Conference on**. v. 2, p. 590-596, 2005.

FILM ENGINE, **Real Time VR Film**. Disponível em: <<http://www.filmengine.com/>>. Acesso em: 15 de fev. 2017.

FOSTER, Jeff. **The Green Screen Handbook: Real-World Production Techniques**. 1 ed. Sybex, 2010. 384 p.

FOURNIER, A. **Illumination problems in computer augmented reality**. 1994.

GASPARI, T.; SEMENTILLE, A. C.; VIELMAS, D. Z.; AGUILAR, I. A.; MARAR, J. F. ARSTUDIO: a virtual studio system with augmented reality features. In: **Proceedings of the 13th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry**, p. 17-25, 2014.

GIBBS, S.; ARAPIS, C.; BREITENEDER, C.; LALIOTI, V.; MOSTAFAWY, S.; SPEIER, J. Virtual studios: An overview. **IEEE multimedia**, v. 5, n. 1, p. 18-35, 1998.

GLEBAS, Francis. **Directing the Story: Professional Storytelling and Storyboarding Techniques for Live Action and Animation**. 1. ed. Burlington: Elsevier, 2009. 346 p.

GOLDSTEIN, S.C.; SCHMIT, H.; MOE, M.; BUDIU, M.; CADAMBI, S.; TAYLOR, R.R.; LAUFER, R. PipeRench: a co/processor for streaming multimedia acceleration. **ACM SIGARCH Computer Architecture News**, v.27, n.2, p.28-39, 1999.

GOOGLE PLAY. **Unity Remote 5**, 2016. Disponível em: <<https://play.google.com/store/apps/details?id=com.unity3d.genericremote>>. Acesso em: 15 de jan. 2017.

GOUSSENCOURT, T.; BERTOLINO, P. Using the unity® game engine as a platform for advanced real time cinema image processing. **In Image Processing (ICIP), 2015 IEEE International Conference on**. IEEE, p. 4146-4149, 2015.

GOUSSENCOURT, T.; DELLAC, J.; BERTOLINO, P. A Game Engine as a Generic Platform for Real-Time Previz-on-Set in Cinema Visual Effects. **In International Conference on Advanced Concepts for Intelligent Vision Systems**. Springer International Publishing, p. 883-894, 2015.

GRAU, O. A 3D production pipeline for special effects in TV and film. In: MIRAGE 2005 CONFERENCE, 2005, Rocquencourt. **IEEE Proceedings – Vision, Image Signal Processing**. Rocquencourt: INRIA, 2005.

GRAU, O.; HEZLE, V.; JORIS, E.; KNOP, T.; MICHOD B.; SLUSALLEK, P.; BEKAERT P.; STARCK J. Dreamspace: a platform and tools for collaborative virtual production. **IET IBC 2016 Conference Proceedings**, 2016.

GRAU, O.; PRICE, M.; THOMAS, G. A. Use of 3-D Techniques for Virtual Production. **Proceedings of SPIE 4309, Videometrics and Optical Methods for 3D Shape Measurement**, 2002.

GRAU, O.; PULLEN, T.; THOMAS, G. A. A Combined Studio Production System for 3-D Capturing of Live Action and Immersive Actor Feedback. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 14, n. 3, p. 370-380, 2004.

GREGORY, Jason. **Game Engine Architecture**. 2 ed. CRC Press, 2009. 1052 p.

GÜNSEL, B.; TEKALP, A. M.; VAN BEEK, P. J. L. Object-based video indexing for virtual-studio productions. **IEEE Computer Society Conference on Computer Vision and Pattern Recognition**, p. 769-774, 1997.

HAILEY, KEITH R. Photographic **system using chroma-key processing**. US6441865 B1. 5 Sep 1997, 27 Aug 2002. US Patent 6,441,865.

HEZLE, V.; GRAU, O.; KNOP, T. Virtual Production. In: MAGNOR, M. A.; GRAU, O.; SORKINE-HORNUNG, O.; THEOBALT, C. **Digital Representations of the Real World: How to Capture, Model and Render Visual Reality**. Boca Raton, CRC Press, 2015. cap. 23, p. 347-357.

HUGHES, John F.; DAM, Andries; MCGUIRE, Morgan; SKLAR, David F.; FOLEY, James D.; FEINER, Steven K.; AKELEY, Kurt. **Computer Graphics: Principles and Practice**. 3 ed. Addison-Wesley Professional, 2013. 1264 p.

HUGHES, Michael. **Digital Filmmaking for Beginners: A Practical Guide to Video Production**. 1 ed. McGraw-Hill, 2012. 200 p.

IDDAN, G. J.; YAHAV, G. Three-dimensional imaging in the studio and elsewhere. **Proceedings SPIE 4298**. p. 48-55, 2001.

JACK, Keith. **Video Demystified: A Handbook for the Digital Engineer**. 5 ed. Newnes, 2007. 944 p.

KATO, H.; BILLINGHURST, M.; POUPYREV, I.; IMAMOTO, K.; TACHIBANA, K. **Virtual object manipulation on a table-top AR environment**. In: **Augmented Reality, ISAR**. Proceedings. IEEE and ACM International Symposium on. IEEE, p. 111-119, 2000.

KATO, Hirokazu; BILLINGHURST, Mark; POUPYREV, Ivan. **ARToolKit**, version 2.33, 2000. Disponível em: <<http://www.tinmith.net/lca2004/ARToolkit/ARToolKit2.33doc.pdf>>. Acesso em: 10 de fev. 2017.

KIYOKAWA, K.; KURATA, Y.; OHNO, H. An optical see-through display for mutual occlusion with a real-time stereovision system. **Computers & Graphics**, v. 25, n. 5, p. 765-779, 2001.

LACHAT, E.; MACHER, H.; MITTET, M.; LANDES, T.; GRUSSENMEYER, P. First experiences with Kinect v2 sensor for close range 3D modelling. **The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences**, v. 40, n. 5, p. 93, 2015.

LAGARDE, Sébastien; LACHAMBRE, Sébastien; JOVER, Cyril. **An Artist-Friendly Workflow for Panoramic HDRI**, 2016. Disponível em: <http://blog.selfshadow.com/publications/s2016-shading-course/unity/s2016_pbs_unity_hdri_notes.pdf>. Acesso em: 18 de abr. 2017.

LANKINEN, K.; SAARI, J.; HARI, R.; KOSKINEN, M. Intersubject consistency of cortical MEG signals during movie viewing. **NeuroImage**, v.92, p. 217-224, 2014.

MICROSOFT, **Kinect for Windows v2**. Disponível em: <<http://blogs.microsoft.co.il/msdn/2015/01/07/kinect-for-windows-v2/>>. Acesso em: 18 de dez. 2016.

MICROSOFT, **Kinect tools and resources**. Disponível em: <<https://developer.microsoft.com/en-us/windows/kinect/tools>>. Acesso em: 10 de abr. 2017.

MILLERSON, Gerald; OWENS, Jim. **Television Production**. 14. ed. Oxford: Elsevier, 2009. 423 p.

MOESLUND, T. B.; GRANUM, E. A survey of computer vision-based human motion capture. **Computer vision and image understanding**, v. 81, n. 3, p. 231-268, 2001.

MSDN MICROSOFT. **JointType Enumeration**. Disponível em: <<https://msdn.microsoft.com/en-us/library/microsoft.kinect.jointtype.aspx>>. Acesso em: 5 de abr. 2017.

NAVAB, N.; BASCLE, B.; APPEL, M.; CUBILLO, E. Scene augmentation via the fusion of industrial drawings and uncalibrated images with a view to marker-less calibration. In: **Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality**. p. 125-133, 1999.

NORTHAM, L.; ISTEAD, J.; KAPLAN, C. RTFX: on-set previs with unrealengine3. In **International Conference on Entertainment Computing**, p. 432-435, 2011.

NORTHAM, L.; ISTEAD, J.; KAPLAN, C. A collaborative real time previsualization tool for video games and film. **In ACM SIGGRAPH 2012 Posters**, p. 121, 2012.

OPENEXR, **OpenEXR**. 2017. Disponível em: <<http://www.openexr.com/>>. Acesso em: 15 de abr. 2017.

OWEN, C. B.; XIAO, F.; MIDDLELIN, P. What is the best fiducial?. **In: Augmented Reality Toolkit**, The First IEEE International Workshop. IEEE, p. 8, 2002.

OWEN, C. B.; ZHOU, J.; TANG, K. H.; XIAO, F. Augmented imagery for digital video applications. **Handbook of Video Databases: Design and Applications**, p. 319, 2003.

PFEIFFER, Silvia; GREEN, Tom. **Beginning HTML5 Media**: Make the most of the new video and audio standards for the Web. 2 ed. Apress, 2015. 304 p.

PIRES, K; SIMON, G. YouTube live and Twitch: a tour of user-generated live streaming systems. **In Proceedings of the 6th ACM Multimedia Systems Conference**. p. 225-230, 2015.

PRITCH, Y.; RAV-ACHA, A.; GUTMAN, A.; PELEG, S. Webcam synopsis: Peeking around the world. **In Computer Vision, ICCV 2007**. IEEE 11th International Conference on. p.1-8, 2007.

RAHBAR, K.; POURREZA, H. R. Inside looking out camera pose estimation for virtual studio. **Graphical Models**, v. 70, n. 4, p. 57-75, 2008.

RAJA, Y.; MCKENNA, S. J.; GONG, S. Tracking and segmenting people in varying lighting conditions using colour. **In: Automatic Face and Gesture Recognition**, Proceedings. Third IEEE International Conference on. IEEE, p. 228-233, 1998.

RATTHALER, M. Virtual Studio Technology: An overview of the possible applications in television programme production. **European Broadcasting Union Technical Review**, v. 268, 1996.

REINHARD, Erik; HEIDRICH, Wolfgang; DEBEVEC, Paul; PATTANAIK, Sumanta; WARD, Greg; MYSZKOWSKI, Karol. **High dynamic range imaging: acquisition, display, and image-based lighting**. 2. ed. Morgan Kaufmann, 2010. 672 p.

SANCHES, Silvio Ricardo Rodrigues. **A utilização da técnica de chroma-key para composição de cenas em ambientes de realidade misturada**. 2007. 150 f. Dissertação (Mestrado em Ciência da Computação) - Centro Universitário Eurípides de Marília, Fundação de Ensino Eurípides Soares da Rocha, Marília.

SANCHES, S. R. R.; SEMENTILLE, A. C.; RODELLO, I. A.; BREGA, J. R. F. The generation of scenes in mixed reality environments using the chromakey technique. **In Artificial Reality and Telexistence**, 17th International Conference on. IEEE, p. 296-297, 2007.

SARAJI, Yamen. **UnityCam**. Disponível em: <<https://github.com/mrayy/UnityCam>>. Acesso em: 20 de abr. 2017.

SCHENK, Sonja; LONG, Ben. **The Digital Filmmaking Handbook**. 4 ed. Boston: Cengage

Learning PTR, 2011. 528 p.

SCHMALSTIEG, D.; WAGNER, D. Experiences with handheld augmented reality. In **Mixed and Augmented Reality, ISMAR 2007**. p. 3-18, 2007.

SEMENTILLE, A. C.; AMÉRICO, M.; BELDA, F. R.; MARAR, J. F.; CUNHA, A. K. ARSTUDIO: Estúdio Virtual para Produção de Conteúdos Audiovisuais em Realidade Aumentada para TV Digital. In: **Tram [p] as de la Comunicación y la Cultura**. p. 89-98, 2014.

SHIMODA, S.; HAYASHI, M.; KANATSUGU, Y. New chroma-key imagining technique with Hi-ision background. **IEEE Transactions on broadcasting**, v. 35, n. 4, p. 357-361, 1989.

SMPTE, **Society of Motion Picture & Television Engineers**. Disponível em: <<https://www.smpte.org/about>>. Acesso em: 16 de abr. 2017.

STANDARD, S. **Archive Frame Rates for D-Cinema**. 2011.

THOMAS, G. Mixed reality techniques for TV and their application for on-set and pre-visualization in film production. In: **International Workshop on Mixed Reality Technology for Filmmaking**, p. 31-36. 2006.

TROTTNOW, J.; GÖTZ, K.; SEIBERT, S.; SPIELMANN, S.; HELZLE, V.; EINABADI, F.; SIELAFF, C.; GRAU, O. Intuitive virtual production tools for set and light editing. In: **Proceedings of the 12th European Conference on Visual Media Production**. ACM, p. 6, 2015.

TWITCH, **2016 Year In Review**, 2017. Disponível em: <<http://www.twitch.tv/year/2016/index.html>>. Acesso em: 28 de abr. 2017.

UNIFYWIKI, **Xbox360Controller**. 2015. Disponível em: <<http://wiki.unity3d.com/index.php?title=Xbox360Controller>>. Acesso em: 15 de fev. 2017.

UNITY TECHNOLOGIES JAPAN, **FrameCapturer**. 2015. Disponível em: <<https://github.com/unity3d-jp/FrameCapturer>>. Acesso em: 15 de fev 2016.

UNITY3D, **Unity**, 2017a. Disponível em: <<https://unity3d.com/unity>>. Acesso em: 16 de abr. 2017.

UNITY3D, **Movie Texture**, 2017b. Disponível em: <<https://docs.unity3d.com/Manual/class-MovieTexture.html>>. Acesso em: 18 de abr. 2017.

UNITY3D, **Components**, 2017c. Disponível em: <<https://docs.unity3d.com/Manual/Components.html>>. Acesso em: 21 de abr. 2017.

VAN DEN BERGH, F.; LALIOTI, V. Software chroma-keying in an immersive virtual environment. **South African Computer Journal**, n. 24, p. 155-162, 1999.

VAUGHAN, William. **Digital Modeling**. Berkeley: Pearson Education, 2012. 434 p.

VIZONI, Marcelo Vilela. **Ferramenta de suporte à captura e armazenamento de**

movimentos corporais para a animação de personagens virtuais. 2017. 52 f. Dissertação (Bacharel em Sistemas de Informação) - Universidade Estadual Paulista “Júlio de Mesquita Filho”, Bauru.

VUFORIA, **Features**. Disponível em: <<https://www.vuforia.com/Features>>. Acesso em: 16 de abril 2017.

WANG, J.; COHEN, M. F. Image and Video Matting: A Survey. **Foundations and Trends in Computer Graphics And Vision**, v. 3, n. 2, p. 97-175, 2007.

WEICHERT, F.; BACHMANN, D.; RUDAK, B.; FISSELER, D. Analysis of the accuracy and robustness of the leap motion controller. **Sensors**, v. 13, n. 5, p. 6380-6393, 2013.

WONG, H. Previsualization: assisting filmmakers in realizing their vision. **In: SIGGRAPH Asia 2012**. p.1-20, 2012.

WRIGHT, Steve. **Digital Compositing for Film and Video**. Focal Press 3 ed., 2010. 512 p.

XIPH, **The Ogg container format**, 2011. Disponível em: <<https://www.xiph.org/ogg/>>. Acesso em: 20 de abr. 2017.

ZHOU, F.; DUH, H. B.-L.; BILLINGHURST, M. **Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR**. In: Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality. IEEE Computer Society. p. 193-202. 2008.