

**UNIVERSIDADE ESTADUAL PAULISTA
“JÚLIO DE MESQUITA FILHO”
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE COMPUTAÇÃO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

IVAN ABDO AGUILAR

**A UTILIZAÇÃO DO SISTEMA ARSTUDIO NA GERAÇÃO DE CONTEÚDOS E
UMA ANÁLISE DA MODELAGEM DE OBJETOS VIRTUAIS**

BAURU
2015

IVAN ABDO AGUILAR

**A UTILIZAÇÃO DO SISTEMA ARSTUDIO NA GERAÇÃO DE CONTEÚDOS E
UMA ANÁLISE DA MODELAGEM DE OBJETOS VIRTUAIS**

Monografia para Trabalho de Conclusão de Curso do
Curso de Bacharelado em Ciência da Computação da
Universidade Estadual Paulista “Júlio de Mesquita
Filho”, Faculdade de Ciências, campus Bauru.

Orientador: Dr. Antonio Carlos Sementille

BAURU
2015

DEDICATÓRIA

Dedico este trabalho aos meus pais, Francisco Carlos Aguilar e Glaucy Maria de Avila Abdo Aguilar.

AGRADECIMENTO

Gostaria de agradecer não só todas as pessoas envolvidas neste trabalho, mas também a todas as pessoas que me ajudaram a chegar onde estou na minha vida pessoal e acadêmica.

Agradeço primeiramente aos meus pais por toda a compreensão e ajuda nesses anos, por todas as oportunidades que me proporcionaram e todo o apoio em cada nova fase da minha vida.

Aos meus professores, desde quando eu era pequeno, por sempre me incentivar a me esforçar, alcançar mais e ir além do esperado e o normal.

Ao Prof. Dr. Antonio Carlos Sementille por ter me aceitado como orientando, pela paciência, disponibilidade e contribuição oferecidas para que este trabalho pudesse ser concluído. Agradeço também pela excelente oportunidade profissional que me proporcionou.

Aos meus colegas da UNESP, Thiago de Gaspari, Everton Motta, Rafael Pedroso e Érika Barbosa por terem me ajudado e apoiado nesta monografia.

Muito obrigado.

RESUMO

Tanto as produções de cinema quanto as de televisão podem se beneficiar das técnicas de combinação de elementos virtuais (2D e 3D) e elementos reais fornecidos pela Realidade Aumentada (RA) e, portanto, a aplicação destas técnicas em sistemas de estúdio virtual tem se revelado uma abordagem para a geração de conteúdo bastante flexível e inovadora. Na cadeia de produção tradicional, os elementos virtuais 3D geralmente são inseridos somente na fase de pós-produção, e, por este motivo, o conteúdo virtual só é visível depois que todo processo de edição é finalizado. No estúdio real, o diretor, os atores e operadores de câmera são guiados apenas por sinais visuais simples, como marcações no chão, onde os personagens ou objetos virtuais devem aparecer. No entanto, com a utilização de técnicas de RA é possível, em tempo real, que os estúdios virtuais permitam a inserção, interação e visualização de objetos virtuais durante, potencialmente, todas as etapas da cadeia de produção.

Considerando-se este contexto, atualmente encontra-se em desenvolvimento no Laboratório de Sistemas Adaptativos e Computação Inteligente (SACI) do Departamento de Computação da UNESP/Bauru, um protótipo de sistema de estúdio virtual com técnicas de RA denominado ARSTUDIO. Dada a inovação e a complexidade deste sistema, tem-se que o enfoque principal deste trabalho consistiu no levantamento e teste de suas funcionalidades e limitações, por meio da geração de um conteúdo digital completo (programa piloto), visando seu futuro aperfeiçoamento. Uma vez que, para a geração de conteúdo digital é essencial a modelagem de objetos virtuais, realizou-se, também, experimentos quanto as técnicas de modelagem de objetos tridimensionais consideradas mais importantes. Os resultados obtidos, tanto na geração de conteúdo quanto na modelagem de objetos tridimensionais, foram promissores indicando a viabilidade do ARSTUDIO.

Palavras-chave: Estúdio Virtual; Realidade Aumentada; ARSTUDIO

ABSTRACT

Both film and television productions can benefit from techniques such as the combination of virtual elements (2D and 3D) and real elements provided by Augmented Reality (AR) and, therefore, the application of these techniques in virtual studio systems have proven to be an approach for the generation of highly flexible and innovative content. In a traditional production pipeline, the 3D virtual elements are usually placed only on the post-production phase and, therefore, the virtual content is only visible after all editing process is concluded. In the real studio, the director, the actors and camera operators are guided only by simple visual signals, such as markings on the ground where the characters or virtual objects should appear. However, with the use of AR techniques it is possible, in real time, that virtual studios allow the viewing, insertion and interaction of virtual objects during, potentially all stages of the production pipeline.

Considering this context, currently under development at Laboratório de Sistemas Adaptativos e Computação Inteligente (SACI), Computer Department at UNESP/Bauru, is a virtual studio system prototype with AR techniques called ARSTUDIO. Given the novelty and complexity of this system, the main focus of this work was the survey and testing of its features and limitations, through the generation of a complete digital content (a pilot program), in view of future improvement. Since, for the generation of digital content, it is essential the modeling of virtual objects, experiments on modeling techniques of three-dimensional objects were conducted. The results obtained, both in the content generation as well as in the modeling of three-dimensional objects, were promising indicating the viability of the ARSTUDIO.

Keywords: Virtual Studio; Augmented Reality; ARSTUDIO

LISTA DE ABREVIATURAS E SIGLAS

2D	Plano Bi-dimensional
3D	Espaço Tri-dimensional
API	<i>Application Programming Interface</i>
CAD	<i>Computer-aided Design</i>
CAM	<i>Computer-aided Manufacturing</i>
CCD	<i>Charge-coupled device</i>
CPU	<i>Central Processing Unit</i>
DOF	<i>Degrees of Freedom</i>
GPU	<i>Graphics Processing Unit</i>
GUI	<i>Graphical User Interface</i>
LADAR	<i>LAser Detection And Ranging</i>
LED	<i>light-emitting diode</i>
LIDAR	<i>Light Detection and Ranging</i>
NRSfM	<i>Nonrigid Structure from Motion</i>
NURBS	<i>Non-Uniform Rational B-Splines</i>
OpenCV	<i>Open Source Computer Vision Library</i>
OpenGL	<i>Open Graphics Library</i>
OSGART	<i>OpenSceneGraph + ARToolKit</i>
RA	Realidade Aumentada
RADAR	<i>RAdio Detection And Ranging</i>
RGB	<i>Red, Green, Blue</i>
RGB-D	<i>Red, Green, Blue, Depth</i>
SACI	Sistemas Adaptativos e Computação Inteligente

SDK	<i>Software Development Kit</i>
SfM	<i>Structure from Motion</i>
SIFT	<i>Scale Invariant Feature Transform</i>
SLAM	<i>Simultaneous Localization and Mapping</i>
SODAR	<i>SOnic Detection And Ranging</i>
SONAR	<i>SOund Navigation And Ranging</i>
SURF	<i>Speeded Up Robust Features</i>
ToF	<i>Time of Flight</i>
UV	<i>Horizontal, Vertical</i>

LISTA DE ILUSTRAÇÕES

Figura 1 - Uso do <i>Matting</i> digital	23
Figura 2 - Realidade Aumentada	26
Figura 3- Cadeia de produção tradicional.....	27
Figura 4 - Cadeia de produção otimizada.....	28
Figura 5 - Módulos do ARSTUDIO.	29
Figura 6 - Cena aumentada, gerada pelo ARSTUDIO.	31
Figura 7 - Bibliotecas usadas na implementação do ARSTUDIO.	32
Figura 8 - Oclusão mútua.....	35
Figura 9 - Modelo de <i>wireframe</i>	37
Figura 10 - Modelo de superfície.....	38
Figura 11 - Mapa de textura.	39
Figura 12 - Objeto 3D com textura.	40
Figura 13 - Esquema do <i>Structure from Motion</i>	48
Figura 14 - Nuvem de pontos.	56
Figura 15 - Equipamentos utilizados com o sistema ARSTUDIO.	61
Figura 16 - Outros equipamentos utilizados com o sistema ARSTUDIO.	62
Figura 17 - Equipamentos utilizados na modelagem de objetos virtuais.....	62
Figura 18 - Outros equipamentos utilizados na modelagem de objetos virtuais.....	63
Figura 19 - Visão da Mesa do Operador.	68
Figura 20 - Cenário Real.	69
Figura 21 - Área de Produção.....	69

Figura 22 - Conteúdo Piloto, Cena 1	71
Figura 23 - Conteúdo Piloto, Cena 2	72
Figura 24 - Conteúdo Piloto, Cena 3	73
Figura 25 - Conteúdo Piloto, Cena 4	74
Figura 26 - Conteúdo Piloto, Cena 5	75
Figura 27 - Área de experimentação com o Kinect v2	86
Figura 28 - Resultado com o Kinect v2	87
Figura 29 - Erro com o Kinect v2	87
Figura 30 - Experimentação com o Kinect v2 e uma plataforma giratória.....	89
Figura 31 - Resultados do Kinect v2 com uma plataforma giratória.....	90
Figura 32 - Experimentação com o Scanner 3D.....	93
Figura 33 - Reconstruções obtidas com o uso do Scanner 3D	93
Figura 34 - Resultados da cabeça de manequim com o 123D Catch.....	98
Figura 35 - Resultados da cabeça de manequim com o ReCap 360.....	99
Figura 36 - Resultados do dragão pelo VisualSfM com o CMPMVS.....	100
Figura 37 - Resultados do dragão com o ReCap 360.....	101
Figura 38 - Área de experimentação do SfM.	103
Figura 39 - Resultados de uma pessoa com o ReCap 360.	104
Figura 40 - Ambiente de experimentação em uma sala da aula.	105
Figura 41 - Cabeça do manequim com características.....	106
Figura 42 - Resultados do telefone com o ReCap 360.....	107
Figura 43 - Textura da sala antes e após o tratamento.....	108

Figura 44 - Modelo da sala gerada antes do tratamento de imagem..... 108

Figura 45 - Modelo da sala gerada após o tratamento de imagem. 109

LISTA DE QUADROS

Quadro 1 - Resultados comparativos entre o 123D Catch e o ReCap 360.	97
Quadro 2 - Resultados comparativos entre o VisualSfM com o CMPMVS e o ReCap 360 .	100
Quadro 3 - Configurações dos programas de SfM.	101

ÍNDICE

1 INTRODUÇÃO.....	16
1.1 Problema	17
1.2 Justificativa	18
1.3 Objetivos.....	18
1.4 Estruturação da Monografia	18
2 FUNDAMENTAÇÃO TEÓRICA	20
2.1 Visão Geral dos Estúdios Virtuais.....	20
<i>2.1.1 Conceitos Principais</i>	<i>20</i>
<i>2.1.1.1 Matting Digital.....</i>	<i>22</i>
<i>2.1.1.1.1 Chroma-key.....</i>	<i>23</i>
<i>2.1.1.1.2 Color Difference Key.....</i>	<i>24</i>
<i>2.1.1.2 Realidade aumentada.....</i>	<i>25</i>
<i>2.1.2 Cadeia de produção</i>	<i>26</i>
2.2 Sistema ARSTUDIO	28
<i>2.2.1 Ambiente de desenvolvimento do ARSTUDIO.....</i>	<i>31</i>
<i>2.2.1.1 ARToolKit</i>	<i>32</i>
<i>2.2.1.2 OpenSceneGraph</i>	<i>33</i>
<i>2.2.1.3 OSGART</i>	<i>33</i>
<i>2.2.1.4 OpenCV.....</i>	<i>33</i>
<i>2.2.1.5 Qt Framework.....</i>	<i>33</i>

2.2.1.6 <i>FMOD Ex API</i>	34
2.2.1.7 <i>libfreenect</i>	34
2.3 Conteúdos tridimensionais.....	35
2.3.1 <i>Definições básicas</i>	35
2.3.2 <i>Modelagem 3D</i>	41
2.3.2.1 <i>Modelagem por software CAD</i>	41
2.3.2.2 <i>Modelagem por Scanners</i>	43
2.3.2.3 <i>Modelagem baseada em imagens</i>	46
2.3.2.4 <i>Uso do SfM na indústria</i>	59
3 MATERIAIS E MÉTODOS	60
3.1 Materiais.....	60
3.2 Métodos	63
3.1.1 <i>Levantamento Bibliográfico</i>	64
3.1.1.1 <i>Estúdios Virtuais</i>	65
3.1.1.2 <i>Modelagem 3D</i>	65
3.1.2 <i>Reconstrução do ambiente do ARSTUDIO</i>	66
3.1.2.1 <i>Ambiente de Software e Hardware</i>	66
3.1.2.2 <i>Ambiente físico</i>	66
4 DESENVOLVIMENTO DO PROJETO PROPOSTO	70
4.1 ARSTUDIO	70
4.1.1 <i>Geração de um conteúdo piloto utilizando o ARSTUDIO</i>	70
4.1.2. <i>Levantamento das funcionalidades oferecidas</i>	75

4.1.2.2 Dificuldades na geração do conteúdo.....	77
4.1.2.3 Levantamento das limitações presentes	79
4.1.3 Proposta de novas funcionalidades	80
4.2 Técnicas de geração de objetos tridimensionais.....	82
4.2.1 Técnicas estudadas.....	83
4.2.2 Experimentos com Câmeras RGB-D.....	84
4.2.2.1 Objeto fixo e câmera em movimento	86
4.2.2.2 Câmera fixa e objeto em movimento	88
4.2.2.3 Análise dos resultados a partir da câmera RGB-D	91
4.2.3 Scanner 3D	91
4.2.3.1 Experimentos utilizando o Scanner 3D	92
4.2.3.2 Análise dos resultados a partir do uso do scanner.....	94
4.2.4 Structure from Motion.....	95
4.2.4.1 Experimentos iniciais	96
4.2.4.2 Ambiente interno com características ao redor do objeto	101
4.2.4.3 Ambiente interno com características no objeto.....	104
4.2.4.4 Análise dos resultados a partir do Structure from Motion.....	109
4.2.4 Considerações Finais sobre os experimentos.....	110
5 CONCLUSÃO.....	112
REFERÊNCIAS.....	114

1 INTRODUÇÃO

As últimas décadas foram marcadas por avanços em inúmeras áreas da ciência, inclusive na área de tecnologia software, não só como consequência de melhoramento e geração de novos algoritmos, mas também pelo avanço e diminuição de custo de tecnologia de hardware dos computadores. Esses avanços proporcionaram drásticas mudanças na produção de conteúdo para cinema, comerciais de televisão, programação de televisão e até na internet, principalmente no quesito da gravação e pós-produção. Com o avanço das tecnologias, não só em poder de processamento e memória mas também na leveza, portabilidade e flexibilidade, e diminuição dos custos, que proporciona uma distribuição a uma rede maior de consumidores profissionais, estúdios iniciaram uma transição de incorporação de técnicas voltadas à manipulação das imagens virtualmente, utilizando o máximo do potencial dos computadores, não só independentes mas também em clusters, para realização de processamento de vídeo e imagem em tempo real.

Como consequência, surgiram os estúdios virtuais, um estúdio cujo conceito é proporcionar a composição de imagens sintéticas, o processo de combinar duas ou mais camadas de imagens (WRIGHT, 2010), com vídeo real. Conhecido também como realidade virtual em terceira pessoa, a composição permite que um "sinal mixado" seja assistido de modo que exista uma combinação entre objetos físicos com ambientes e objetos virtuais (GIBBS et al., 1998).

No modo tradicional de produção de filmes, fase de Story Board, fase de planejamento, fase on-set e fase de pós-produção (GRAU, 2005), tem-se que após a fase de planejamento, o cenário e objetos de cena precisam ser construídos. Para que um cenário e os seus objetos sejam construídos com qualidade e realismo, é demandado muito tempo, esforço e recursos financeiros, tanto para a criação e utilização, quanto para a montagem e sua desmontagem após a filmagem. Com o uso de estúdios virtuais, o cenário e objetos podem ser construídos virtualmente, proporcionando uma flexibilidade e diminuição de custos e tempo de criação (BLONDÉ et al., 1996). Esses cenários também podem ser modificados em tempo real.

Este trabalho propôs o levantamento e teste das funcionalidades e limitações no ARSTUDIO, um estúdio virtual iniciado em 2010 (SEMENTILLE et al., 2014) , por meio da

geração de um conteúdo digital completo, visando seu futuro aperfeiçoamento. Este trabalho também apresenta os resultados dos experimentos quanto as técnicas de modelagem de objetos tridimensionais consideradas mais importantes e a partir disto a elaboração de uma análise para a modelagem de objetos virtuais.

1.1 Problema

Produções cinematográficas e de televisão exigem a construção de um cenário com objetos, ou personagens virtuais, para a gravação de determinadas cenas e a interação do atores com esses objetos e personagens. A construção física do cenário e dos objetos exigem alto custo, um complexo planejamento, uma equipe numerosa, tanto para a criação, ou confecção física, quanto para o transporte, montagem, desmontagem, armazenamento e esses aspectos podem ser ainda mais custosos e trabalhosos quando se trata de alterações ou modificações nos mesmos. Isso produz um alto custo no processo de uma produção televisiva (BLONDÉ et al., 1996).

Existem também cenas que precisam de efeitos especiais e efeitos visuais. Efeitos visuais é a modificação, criação ou manipulação de imagens, enquanto que efeitos especiais são feitos na cena que então são fotografadas. Essa manipulação de imagem no efeito visual é normalmente feito por um compositor, alguém especializado compor uma cena com os elementos, sejam eles de objetos virtuais, cenário, etc., de forma que ficam artisticamente fotorrealista (WRIGHT, 2010).

Os estúdios virtuais têm como objetivo combinar imagens reais com imagens sintéticas para a produção de uma cena final, e apresentar essa combinação em tempo real. Com a utilização de estúdios virtuais, animações, efeitos especiais, objetos e cenários virtuais podem ser visualizados e manipulados no decorrer da produção. Com os objetos virtuais, ao invés de físicos e de cenários virtuais é possível diminuir o custo de produção, bem como o esforço e trabalho necessário, como por exemplo na movimentação da equipe de produção para filmagem em cenários reais distintos. Estúdios virtuais ainda carecem de muitas funcionalidades para atingir todo o seu potencial. Existe também uma carência por análises da modelagem destes conteúdos virtuais e tridimensionais a serem utilizados durante a produção.

1.2 Justificativa

A melhoria nas tecnologias está solucionando cada vez mais os problemas visuais, o que vem aumenta a qualidade de efeitos visuais com uma diminuição na dificuldade e tempo de criação, que gera então uma ascensão na procura, pelo espectador, por filmes com efeitos cada vez mais espetaculares. Com esse aumento na qualidade e poder das tecnologias, necessita-se, também, haver um aumento nos programas que podem tratar, criar e compor essas cenas de pós-produção, para que a produção seja mais rentável, rápido e com maior qualidade (WRIGHT, 2010).

A utilização de um estúdio virtual pode trazer uma diminuição nos altos custos de produção, diminuição no tempo de geração de conteúdo, por exemplo em utilizar cenários virtuais ao invés de ter que filmar a cena em um outro local para obter os aspectos visuais e naturais, o que traz uma flexibilidade tanto de possibilidades numerosos de ambientes de cena quanto a criatividade dos roteiristas e diretores.

Portanto, a análise de estúdios virtuais, como o ARSTUDIO, visando o levantamento das funcionalidades já existentes, bem como as limitações é extremamente útil e benéfico para as produtoras cinematográficas que usam ou poderão utilizar estúdios virtuais na sua cadeia de produção. Para o uso deste sistema é necessário que haja conteúdos tridimensionais disponíveis e com fotorrealismo.

1.3 Objetivos

Realizar uma análise, teórica e prática, do ARSTUDIO em suas competências, funções oferecidas e passíveis de serem incorporadas, visando seu futuro aperfeiçoamento. Um outro objetivo consistiu na elaboração de uma análise quanto a modelagem de objetos virtuais, dada as diversas técnicas atualmente existentes.

1.4 Estruturação da Monografia

Esta monografia foi estruturada da forma que segue:

- O Capítulo 2 expõe as principais características dos estúdios virtuais, dado um enfoque especial para o estúdio virtual ARSTUDIO, utilizado neste trabalho.

São também descritos os conceitos de conteúdos tridimensionais, bem como as principais técnicas para a geração do mesmo;

- No Capítulo 3 é apresentada a metodologia utilizada no desenvolvimento deste trabalho, assim como os materiais utilizados para a realização dos experimentos;
- No Capítulo 4 tem-se a descrição de um conteúdo piloto obtido pela utilização do ARSTUDIO, bem como uma análise deste estúdio virtual a partir da experiência de geração deste conteúdo. Também é apresentado técnicas de criação de conteúdos tridimensionais e uma análise a partir de experimentações;
- O Capítulo 5 apresenta as conclusões deste trabalho e indicação dos trabalhos futuros para o aperfeiçoamento do estúdio virtual e de técnicas para a modelagem de objetos virtuais.

2 FUNDAMENTAÇÃO TEÓRICA

Em um estúdio virtual, diversas técnicas são aplicadas, como de realidade aumentada, segmentação de imagem e interação, por exemplo, nas filmagens de um programa de televisão ou de um filme para o cinema, para compor o resultado final da cena apresentada ao usuário. Consequentemente, é indispensável definir alguns conceitos para que se possa compreender a concepção e utilização de um software de estúdio virtual, bem como a geração de uma análise da modelagem de objetos que possam ser utilizados em tal ambiente.

Neste capítulo, serão apresentados os aspectos essenciais de estúdios virtuais, *matting* digital, realidade aumentada, cadeia de produção, o ARSTUDIO, conteúdos tridimensionais, bem como formas de geração destes conteúdos.

2.1 Visão Geral dos Estúdios Virtuais

2.1.1 Conceitos Principais

São estúdios de cenários virtuais, permitem a composição de vídeo real capturado com imagens sintéticas inseridas pelo computador. Segundo Millerson e Owens (2009), embora no início o custo da criação de um sistema de estúdio virtual integrado com câmeras possa ser bastante significativo, as economias de não ter que mudar rapidamente muitos tipos diferentes de cenas físicas pode compensar pelo preço, a longo prazo. Esses estúdios, também chamados de Realidade Virtual em terceira pessoa, permitem uma técnica de composição que as pessoas que assistem à este "sinal composto" conseguem visualizar outras pessoas (atores) e objetos físicos combinados com um ambiente virtual (GIBBS et al. 1998). O sistema de estúdio virtual é uma ferramenta importante para estúdios de produção, proporcionando muitas oportunidades criativas, bem como a redução de custos. A flexibilidade oferecida por mudar facilmente de um cenário para outro elimina as restrições de custo e tempo de construção, armazenamento, transporte, e re-ajuste nas produções atuais (BLONDÉ et al., 1996). Sistemas de estúdio virtual atuam compondo filmagens de cenas reais com objetos 3D que são renderizados em tempo real e sincronizados com o movimento da câmera (GÜNSEL, TEKALP, VAN BEEK, 1997). Além disso, um estúdio virtual torna possível a visualização de efeitos de vídeo em tempo real ao invés de visualizar-los durante a pós-produção.

Segundo Hayashi et al. (1996), o sistema de estúdio virtual desenvolvido por eles, o "Desktop Virtual Studio (DVS)", é baseado nos seguintes processos:

- Gerar ou converter cenários de estúdios, atores e câmeras como forma reproduutível;
- Reconstituir os mesmos em um espaço de estúdio virtual implementado em um computador;
- Filmar cenas com uma câmera virtual e modificá-los por meio de interfaces homem-máquina especiais disponíveis em um computador de mesa.

Rahbar e Pourreza (2008) dizem que, sistemas de estúdios virtuais contêm três componentes principais:

- Um sistema de rastreamento de câmera, que crie um fluxo de dados descrevendo a perspectiva da câmera, geralmente chamado de processo de estimativa da pose da câmera;
- Um software de renderização em tempo real, que use os dados de rastreamento da câmera e gera uma imagem sintética de um estúdio;
- Um sistema de mixagem de vídeo, que combina a saída da câmera do estúdio com o vídeo do software de renderização em tempo real, a fim de produzir o vídeo final combinado.

Segundo Grau, Pullen e Thomas (2004) o controle em um sistema de estúdio virtual consiste nos seguintes componentes:

- O controle da câmera, que 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 captura, usado para iniciar e parar o serviço de captura.
- O controle de animação, capaz de começar animações pré-definidas em 3-D ou controlá-las ao vivo.

- A pré-visualização 3-D oferece uma pré-visualização da filmagem para o diretor e os operadores de câmera. Uma malha 3-D texturizada do ator é inserida no cenário virtual para dar uma prévia da composição final da cena.

De acordo com Ratthaler (1996), no caso da tecnologia de estúdio virtual, ambas as câmeras, a real e a virtual, estão permanentemente interligadas. Para alcançar isto, os parâmetros de posição da câmera real necessitam ser determinados.

Estes são:

- As coordenadas x, y, z 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;
- A distância focal e ajuste focal da lente da câmera.

2.1.1.1 Matting Digital

No *Matting* digital é necessário que um elemento do primeiro plano seja extraído de uma imagem de fundo, estimando uma cor e opacidade para o elemento de primeiro plano em cada pixel. O valor da opacidade de cada pixel é geralmente chamado de alfa, e a imagem da opacidade, tomada como um todo, é referido como o mate alfa ou *key*. Opacidades fracionárias (entre 0 e 1) são importantes para a transparência e borrões causados pelo movimento do elemento no primeiro plano, bem como para a cobertura parcial de um pixel de fundo em torno da borda do objeto de primeiro plano. *Matting* é usada para compor o elemento em primeiro plano em uma nova cena (CHUANG et al., 2001).

Matting de vídeo é uma operação crítica na televisão comercial 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 (CHUANG et al., 2002). A Figura 1 ilustra o uso da técnica do *Matting* digital.

Figura 1 - Uso do *Matting* digital.



Fonte: elaborada pelo autor.

2.1.1.1.1 Chroma-key

O *chroma-key* é um dos métodos mais utilizados nas abordagens de *matting* digital em produções televisivas e cinematográficas. Usando o *chroma-key*, uma filmagem é feita em um estúdio com um plano de fundo, *background*, de cor uniforme e os atores à frente do mesmo, no *foreground*. O ator não deve vestir roupas com a mesma cor do plano de fundo para que o ator possa ser diferenciado do plano de fundo pela cor (VAN DEN BERGH e LALIOTI, 1999).

Técnicas de *matting* digital dêem um grande potencial ao *background* e a efeitos especiais. Na televisão, *chroma-key* é usado extensivamente para criar backgrounds e baseia-se num princípio muito simples (MILLERSON e OWENS, 2009).

A cor chave é a cor uniforme escolhida para constituir o *background*. Usualmente escolhe-se as cores azul ou verde como cor chave pelo fato de não serem predominantes na pigmentação da pele humana, causando assim menos interferência, ou dúvida, no processo de *chroma-key* com atores em cena. Porém a cor verde apresenta mais um fator de preferência sobre as demais cores: o mosaico de filtros de cor que fica sobre os fotosensores dos dispositivos de captura de imagens (como 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).

A técnica de chroma-key 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 e YAHAV, 2001).

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 atores e os objetos da cena possam ser transpostos para o fundo 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, em particular quando utilizados em grandes estúdios:

- Dificuldade de proporcionar uma iluminação uniforme e brilhante sobre uma grande área;
- Dificuldade em 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 particulares;
- Áreas onde sombras escuras não podem ser evitados, como debaixo das mesas, podem causar problemas;
- Luz colorida espalhada pelo fundo e piso iluminados podem incidir sobre atores e objetos de cena, mudando sua tonalidade e dando uma aparência anormal. Conhecido como "derrame" ou "*spill*".

2.1.1.2 Color Difference Key

Esse algoritmo, assim como o chroma-key, utiliza uma cor chave para realizar a extração do plano de frente. A idéia do algoritmo é determinar a transparência de cada pixel da imagem baseado na diferença entre os canais das cores R, G e B (*Red, Green, Blue*). Tendo selecionado a cor chave a ser utilizada pelo algoritmo, uma comparação é feita entre as duas cores restantes para determinar qual delas tem maior intensidade naquele *pixel*. Com as duas cores selecionadas, a cor chave é a cor com maior intensidade, é calculada a diferença entre as mesmas. (SCHULTZ, 2006)

2.1.1.2 Realidade aumentada

Realidade aumentada (RA) é uma tecnologia emergente, sendo o seu surgimento na década de 90. Esta tecnologia faz uso do mundo virtual para ampliar a percepção das pessoas do mundo real, pela combinação de informações virtuais geradas por computador com o ambiente capturado do mundo real. A tecnologia de RA contém três principais características: registro de objetos virtuais, combinação entre o real e o virtual e a interação em tempo real (LI, QI, WU, 2012).

Azuma (1997) diz que um sistema de realidade aumentada é composto por três aspectos:

- Combinação de objetos reais e virtuais em um ambiente real;
- Registro, alinhamento, de objetos reais e virtuais entre si;
- Funciona de forma interativa, em três dimensões, e em tempo real.

A realidade aumentada pode ser aplicada a diversas áreas como, Ambientes colaborativos (BUTZ et al., 1999), Visualização de dutos em fábricas (NAVAB et al., 1999), Arte (BILLINGHURST, GRASSET, LOOSER, 2005), Estúdios Virtuais (GASPARI et al., 2014), entre outros. A Figura 2 representa a utilização da realidade aumentada em estúdios virtuais.

Figura 2 - Realidade Aumentada.



Fonte: elaborada pelo autor.

2.1.2 Cadeia de produção

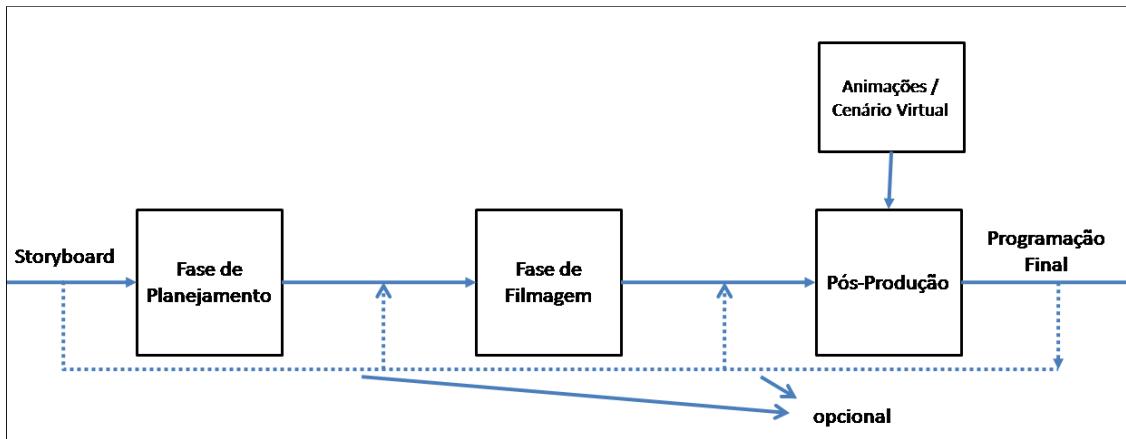
Dependendo da natureza e escopo do projeto, precisa ser decidido o caminho e a agenda que a produção siga desde a idéia inicial até o produto acabado. Isso é chamado de *production pipeline*, ou cadeia de produção. Um *pipeline* nos permite processar atividades em uma sequência: cada fase realiza alguma tarefa e passa o resultado para a próxima fase; a fase original pode, em seguida, começar a realizar sua tarefa nos próximos dados. Quando uma tal cadeia de produção é concebida adequadamente isto pode resultar na melhoria do rendimento (VAUGHAN, 2012; HUGHES et al., 2013).

Segundo Millerson e Owens (2009) existem três etapas principais de produção que a maioria das produções televisivas passam:

1. Planejamento e preparação. A preparação, organização e ensaio antes de começar a produção. Grande parte do trabalho em uma produção geralmente está na fase de planejamento e preparação;
2. Produção. A filmagem do programa em si;
3. Pós-produção. Edição, tratamento adicional, e duplicação.

Previvamente na indústria, nas produções convencionais, o conteúdo 3D era utilizado somente na fase de pós-produção e o conteúdo virtual era visível somente após que todas as gravações das cenas estivessem completas. No cenário real os atores, o diretor e os operadores de câmera só tinham indicações visuais simples, como marcações no chão onde os personagens ou objetos virtuais deveriam aparecer. (GRAU, 2005) A Figura 3 ilustra essa forma de produção.

Figura 3- Cadeia de produção tradicional.

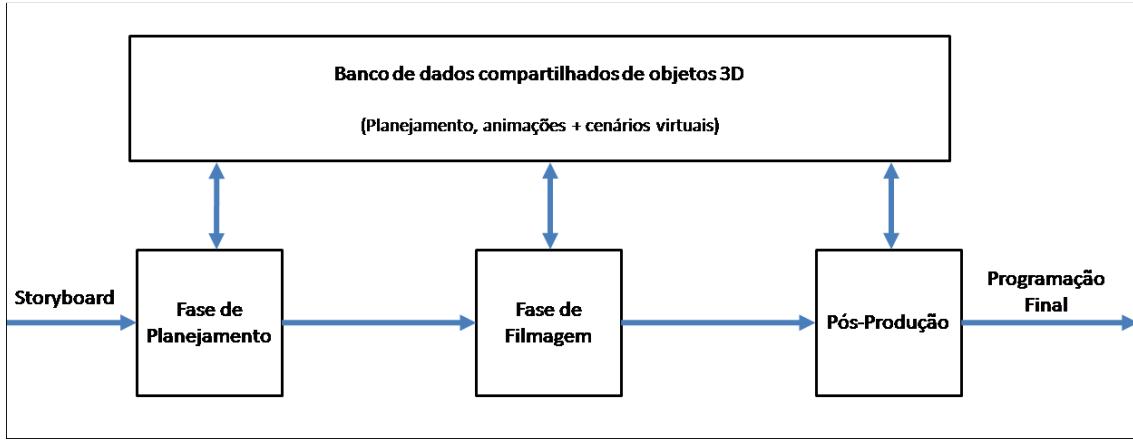


Fonte: GRAU, 2005.

Segundo Millerson e Owens (2009) a etapa de pós-produção, nas produções convencionais, ocorre depois que todos os materiais do programa (vídeo, áudio e gráficos) foram compilados. As filmagens ou segmentos escolhidos são então colocados juntos na ordem apropriada para criar o programa final. A pós-produção é usado para a inserção de objetos virtuais na cena e também é usado para inserção de sons como o riso e aplausos, a criação de montagens, sequências em câmera lenta, mudança de cor, adicionar efeitos sonoros, música,etc.

Grau (2005) diz que a produção de programas de TV ou de cinema que envolvem efeitos especiais com computação gráfica de alta qualidade normalmente contém três fases, como apresentada na Figura 4 abaixo:

Figura 4 - Cadeia de produção otimizada.



Fonte: GRAU, 2005.

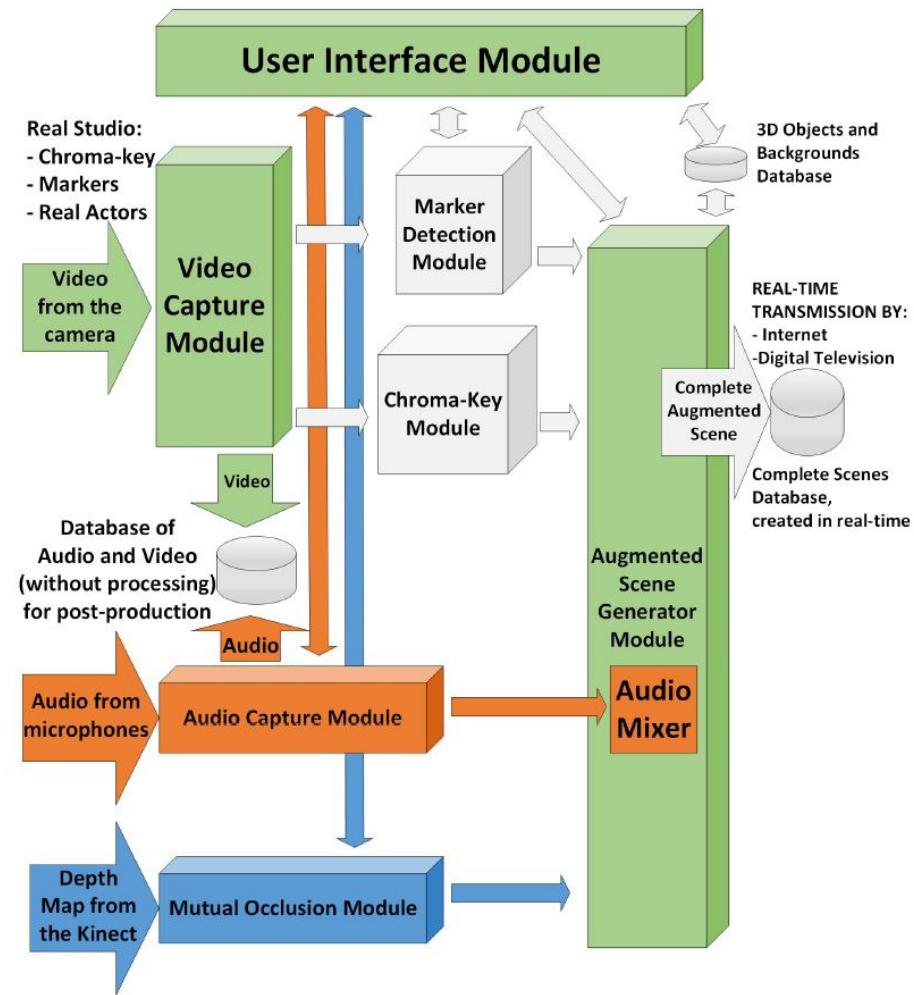
- Fase de Planejamento (Pré-produção): fase onde as ideias concepcionais, normalmente demonstradas através de um storyboard, são transformadas em um roteiro junto com uma lista de cenas e instruções técnicas de como obter os mesmos.
- Fase de Filmagem (*On-Set*): fase onde a filmagem ocorre de acordo com o roteiro.
- Pós-Produção: fase onde o conteúdo virtual é integrado com a filmagem, proveniente da câmera, e as cenas são editadas dentro do conteúdo final.

O storyboard é o início da pré-produção, uma ferramenta de pré-visualização concebida para apresentar de forma visual uma sequência de desenhos quadro-a-quadro adaptadas do roteiro de filmagem. São desenhos de conceito que iluminam e ampliam a narrativa do roteiro e permitem que a equipe de produção organize toda a ação exigida pelo mesmo antes que a filmagens reais sejam realizadas. (HART, 2007)

2.2 Sistema ARSTUDIO

O sistema de estúdio virtual estudado nesse trabalho é o ARSTUDIO (CAMPOS et al., 2010; SEMENTILLE et al., 2014; GASPARI et al., 2014). Ele foi desenvolvido em uma arquitetura modular, para facilitar a manutenção e reutilização do código. Os sete módulos desta arquitetura são: captura de vídeo, interface com o usuário, detecção de marcadores, *chroma-key*, captura de áudio, oclusão mútua e gerador de cena combinada (GASPARI et al., 2014). A Figura 5 apresenta os módulos do sistema.

Figura 5 - Módulos do ARSTUDIO.



Segundo Gaspari et al. (2014), as funções dos módulos são:

- Módulo de captura de vídeo: responsável pela captura de vídeo proveniente da câmera, pela transmissão dessas informações para que outros módulos possam utilizar e processar e pelo armazenamento local do mesmo;
- 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;

- Módulo de Detecção do 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;
- Módulo *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*;
- Módulo de Captura de Áudio: responsável por capturar o áudio de microfones no cenário real;
- Módulo de Oclusão Mútua: executa, com informações de profundidade, a oclusão mútua entre objetos virtuais e os atores reais em cena;
- Módulo Gerador de Cena Aumentada: a partir das informações geradas por todos os outros módulos, gera a cena aumentada final, em tempo real, para transmissão pela internet ou TV digital, ou armazena o fluxo de vídeo localmente.

A cena aumentada gerada pelo sistema é estruturado 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). A Figura 6 ilustra esta cena aumentada gerada pelo sistema.

Figura 6 - Cena aumentada, gerada pelo ARSTUDIO.



Fonte: elaborada pelo autor.

2.2.1 Ambiente de desenvolvimento do ARSTUDIO

Esta seção apresenta as bibliotecas utilizadas na implementação do ARSTUDIO. (GASPARI et al., 2014)

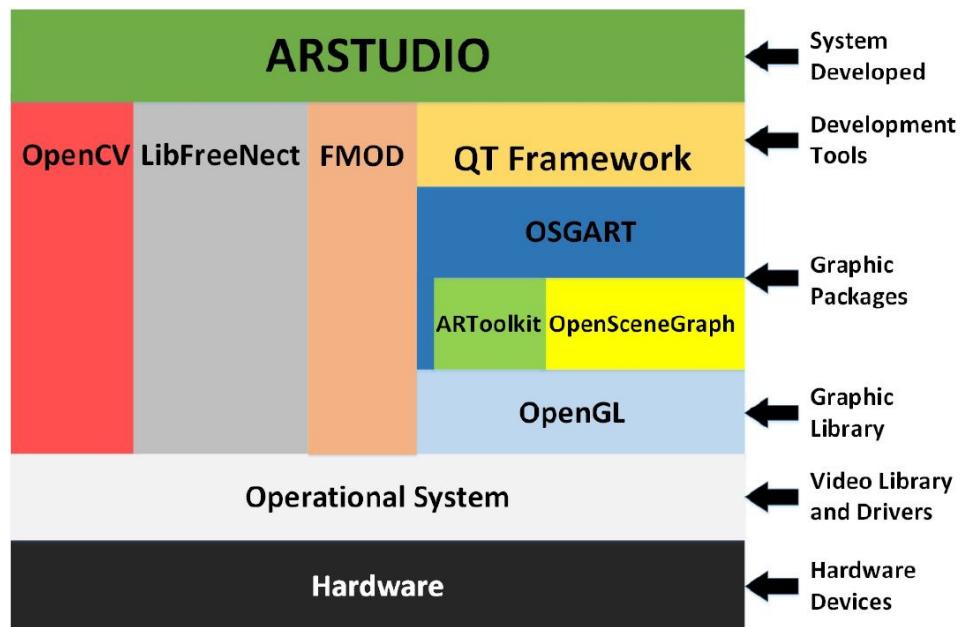
As bibliotecas utilizadas pelo projeto original do ARSTUDIO foram:

- ARToolKit;
- OpenSceneGraph;
- OSGART;
- OpenCV;
- Qt Framework;
- FMOD Ex API;
- libfreenect.

Para desenvolver o ARSTUDIO, optaram-se por usar bibliotecas multi-plataforma e software livre. A Figura 7 mostra a hierarquia das bibliotecas utilizadas para a construção do sistema. Nesta hierarquia, a camada de aplicação é o nível mais alto, com base no arcabouço de desenvolvimento Qt (THELIN, 2007). A biblioteca OpenCV (BRADSKI e KAEHLER,

2008) se encaixa em um nível intermediário entre aplicativos e sistemas operacionais, bem como a biblioteca libfreenect (OpenKinect Project) e a API FMOD (FMOD). Abaixo, bibliotecas de realidade virtual (OpenSceneGraph (WANG e QIAN, 2010)) e realidade aumentada (ARToolKit (KATO, 2002)) estão localizadas, e a biblioteca OSGART (LOOSER et al., 2006) apresenta um nível que englobam as duas anteriores. Em direção ao hardware está a biblioteca de gráficos OpenGL , que suporta as outras bibliotecas, e, finalmente, o sistema operacional. (GASPARI et al., 2014)

Figura 7 - Bibliotecas usadas na implementação do ARSTUDIO.



Fonte: GASPARI et al., 2014.

2.2.1.1 ARToolKit

O ARToolKit é uma biblioteca que facilita o desenvolvimento de aplicações de realidade aumentada. Esta biblioteca utiliza técnicas de visão computacional para calcular a posição da câmera real e a orientação relativa dos marcadores, permitindo a sobreposição de objetos virtuais sobre esses marcadores (KATO, 2000).

2.2.1.2 *OpenSceneGraph*

O OpenSceneGraph é uma biblioteca *middleware* de renderização que eleva o nível de abstração e diminui a complexidade do uso da API de renderização OpenGL, uma API que utiliza comandos mais próximos aos comandos de baixo nível (WANG e QIAN, 2010).

2.2.1.3 *OSGART*

A biblioteca OSGART foi desenvolvida para servir como uma extensão da biblioteca OpenSceneGraph. A biblioteca implementa uma abordagem hierárquica baseada no grafo de cena, com a utilização dos marcadores de realidade aumentada através do ARToolKit, criando uma transição entre ambientes virtuais imersivos e realidade aumentada (LOOSER et al., 2006).

2.2.1.4 *OpenCV*

O OpenCV é uma biblioteca, de código aberto, desenvolvida pela pesquisa da Intel, e tem como objetivo original a criação de avanços na área de aplicações orientadas a uso intensivo de CPU. Atualmente esta biblioteca é utilizada na área de visão computacional com foco em aplicações em tempo real. Tendo como objetivo o fornecimento de uma infraestrutura de visão computacional simples de usar, contendo centenas de algoritmos já implementados (BRADSKI e KAEHLER, 2008).

2.2.1.5 *Qt Framework*

Qt é um *framework* de GUI (*GraphicalUser Interface* ou Interface Gráfica do Usuário) que inclui conjuntos de APIs de renderização gráfica, mecanismos de *layout* e folha de estilo, *widgets*, e ferramentas que podem ser usadas para criar interfaces de usuário. A variedade de *widgets* vai desde objetos simples, como botões e rótulos, aos avançados como editores completos de texto, calendários e objetos com navegadores *web* completos (MIKKONEN; TAIVALSAARI; TERHO, 2009).

O Qt *Framework* foi utilizado no ARSTUDIO para auxiliar na criação da interface do usuário. (GASPARI et al., 2014)

2.2.1.6 FMOD Ex API

A FMOD Ex API faz parte de um conjunto de ferramentas para criação de conteúdo de áudio desenvolvido pela Firelight Technologies. Ela é uma API de baixo nível, contendo recursos que auxiliam a manipulação de áudio, como *mixers*, módulos de saída da interface de hardware, recursos de som 3D, entre outros. (FIRELIGHT TECHNOLOGIES, 2015)

No estúdio virtual ARSTUDIO, essa biblioteca é utilizada no módulo de áudio, que permite a captação e edição do áudio. (GASPARI et al., 2014).

2.2.1.7 libfreenect

A biblioteca usada para acessar os recursos do Kinect foi a libfreenect, uma biblioteca de código aberto desenvolvida e mantida pela comunidade OpenKinect que permite a obtenção das imagens capturadas pela câmera RGB, controle dos LEDs (*light-emitting diode* ou diodo emissor de luz) e dos motores de inclinação do dispositivo, acesso às informações do acelerômetro do Kinect e às informações de profundidade geradas pelo sensor de profundidade.

A oclusão mútua é quando um elemento real no primeiro plano não obstrui um virtual no fundo. Na maioria das aplicações de realidade aumentada, os elementos virtuais são sempre gerados como o primeiro plano, porque não há informações sobre a profundidade dos elementos reais na cena (SANCHES et al., 2012).

A informação de profundidade que é obtida pelo Kinect é representado por um mapa de disparidade, ou profundidade. De acordo com Zhang (2012), este mapa é calculado através de uma comparação entre o padrão, do pontos infravermelho projetados, conhecidos (padrão memorizado pelo Kinect de um plano com uma profundidade conhecida) e do padrão de pontos capturadas pela câmera IR no ambiente.

Com as informações provenientes do Kinect, nesse caso a profundidade, foi possível distinguir qual objeto real está a frente ou atrás do objeto virtual e realizar uma oclusão mútua em que os objetos virtuais pudessem ser colocados tanto a frente quanto atrás dos objetos reais na cena no vídeo composto (GASPARI et al., 2014). A Figura 8 ilustra esta oclusão mútua do ARSTUDIO. Pode-se observar que o primeiro ator real está atrás do sofá e do avião virtual, enquanto o segundo ator real está a frente do sofá virtual.

Figura 8 - Oclusão mútua.



Fonte: elaborada pelo autor.

2.3 Conteúdos tridimensionais

2.3.1 Definições básicas

Na geometria 2D, plano bi-dimensional, utilizamos um plano representado por um sistema de coordenadas retangular, ou cartesiana, com os eixos x e y que representam, por convenção, eixos na horizontal e vertical respectivamente em relação ao ponto de intersecção chamado de origem. Nesse mesmo plano, pontos podem existir e suas coordenadas são obtidas pelo número x e y que correspondem respectivamente a abscissa e ordenada dos pontos. Estes pontos estão situados no plano Euclidiano, R^2 , que é constituído por todos os pares ordenados de números reais (a, b). (CORRAL, 2011)

Na geometria 3D, espaço tri-dimensional, o sistema de coordenadas retangular é estendido para conter um terceiro eixo denominado de eixo-z. Nesse sistema de espaço Euclidiano, R^3 , o plano horizontal, em relação a origem, é o plano-xy, aquele do espaço bi-dimensional, e o eixo-z está no sentido vertical a origem. As coordenadas de um ponto contido nesse espaço são representados por conjuntos de triplos (x,y,z) de números reais. (CORRAL, 2011)

Uma face é uma superfície plana em um polígono tri-dimensional. Um vértice é ponto onde duas ou mais linhas se encontram. (BELL, 2007)

Textura, um tema em visão computacional e computação gráfica. Ao contrário das quantidades tangíveis de forma ou cor, a textura é um termo mais abstrato abrangendo variações tanto aleatórios e deterministas de ambos albedo e altura da superfície. Texturas em 2D contém propriedades em relação ao albedo ou variação de cor, enquanto textura em 3D contem, além das propriedades em 2D, a propriedade de altura. (DANA et al., 1999)

Malhas geométricas, ou simplesmente malhas, são consistidas, normalmente, por muitos triângulos unidos ao longo dos suas arestas para formar uma superfície. Um malha na computação gráfica, além das posições geométricas de suas vértices, muitas vezes contém vários outros atributos de aparência utilizados na prestação de sua superfície. Estes atributos de aparência pode ser classificada em dois tipos: atributos discretos e atributos escalares. Atributos discretos geralmente estão associados com faces da malha. Um atributo discreto comum, o identificador de material, determina a função de sombreamento usado na geração de cada face da malha. Muitos atributos escalares são frequentemente associados com uma malha, incluindo normais e coordenadas de textura. (HUGHES et al., 2013 e HOPPE, 1996)

A subdivisão em uma malha é o ato de substituir um único triângulo por vários triângulos pequenos. (HUGHES et al., 2013)

A simplificação em uma malha é o ato de substituir uma malha por uma outra malha similar, topológica ou geometricamente, mas tem uma estrutura mais compacta, constituída por menos polígonos. (HUGHES et al., 2013)

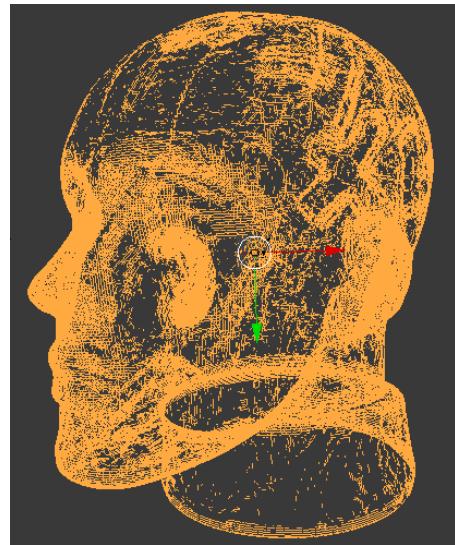
O nível de detalhe (ou *level of detail*) visa a junção da complexidade e do desempenho pelo controle da quantidade de detalhe usado nas representações geométricas em modelos 3D. (LUEBKE et al., 2002)

Na modelagem, ou geração de objetos 3D, o vértice, é o componente de nível mais baixo que compõe um objeto 3D. Quando dois pontos estão conectados, é traçada uma linha. Quando três pontos estão ligados, eles podem se tornar cantos de superfícies em um modelo chamado um polígono. Vários polígonos podem compartilhar os mesmos pontos quando usado em uma malha contíguo. (VAUGHAN, 2012)

O modelo de *wireframe* (ou *wireframe model*) são definidos pelos vértices e as arestas que ligam esses vértices. Eles fixam os contornos de um objeto e permitem uma visão através de qualquer ponto de vista. Entre as arestas não existe alguma relação, uma relação com as

faces não está definida. Informação sobre o interior e o exterior do objeto não está disponível. Pontos e arestas são os únicos elementos geométricos e estão representados por estruturas de lista no computador. Esta é uma vantagem para objetos simples, mas reduz a capacidade de leitura de objetos mais complexos. Esta representação é, portanto, muitas vezes usado para objetos simples e visualizações rápidas (EGELS e MICHEL, 2001). A Figura 9 ilustra este modelo de *wireframe*.

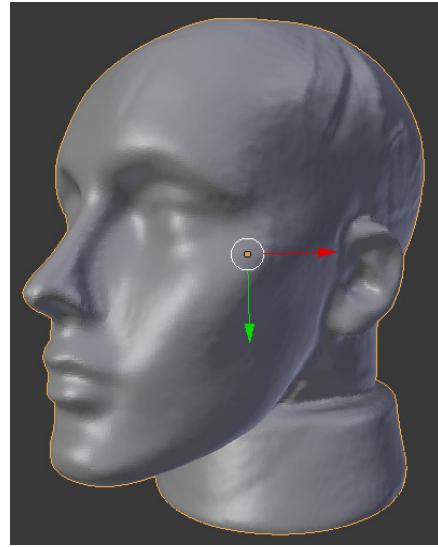
Figura 9 - Modelo de *wireframe*.



Fonte: elaborada pelo autor.

O modelo de superfície (ou *surface model*) representa um conjunto ordenado de superfícies no espaço tridimensional. São principalmente utilizados para a geração de modelos, cujas superfícies consistem em faces não descriptíveis facilmente analiticamente tendo diferentes curvaturas em diferentes direções. Alguns métodos para calcular e demonstrar o modelo são: aproximações Bézier e interpolações do tipo *splines* e *B-spline*. (EGELS e MICHEL, 2001). A Figura 10 ilustra este modelo de superfície.

Figura 10 - Modelo de superfície.



Fonte: elaborada pelo autor.

O modelo volumétrico (ou *volumetric model*) viabiliza o uso de operações booleanas, bem como o cálculo do volume, centro de gravidade e área de superfície. Modelagem de superfície é a método que mais demanda processamento na modelagem. Os modelos sólidos sempre representam a hierarquia do objeto, no qual as primitivas e operações são definidas. As coordenadas tridimensionais da superfície são uma função das coordenadas de superfície, que são utilizados para parametrizar a superfície. (EGELS e MICHEL, 2001)

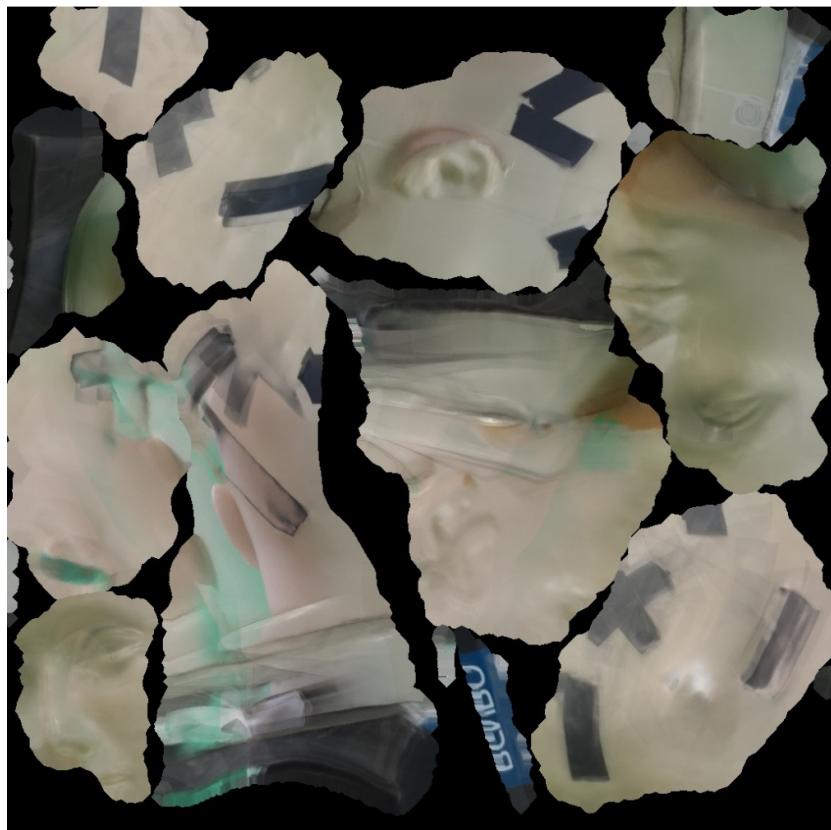
Vaughan (2012) diz que objetos 3D, ou modelos digitais, podem ser classificadas em 3 tipos:

- modelos poligonais são constituídos por um conjunto de pontos, arestas e polígonos.
- superfícies NURBS consistem de uma rede de curvas com superfícies lisas entre eles.
- superfícies de subdivisão são semelhantes aos modelos poligonais, por serem constituídos de pontos, arestas e polígonos, mas também compartilham alguns dos benefícios de superfícies NURBS, colocando-os em sua própria categoria.

Um mapa de textura é uma imagem bitmap, ou mapa de bits, que é aplicada a uma superfície de um polígono. É uma maneira de adicionar detalhes visuais de uma cena. Esses

mapas, ou mapas UV, armazenam informações sobre o posicionamento da textura em um objeto. Esse mapeamento acrescenta duas coordenadas extras para os pontos no objeto 3D; os eixos U (horizontal) e V (vertical), alinhando horizontalmente e verticalmente através da superfície plana do mapa de textura. Coordenadas UV são uma representação 2D do espaço 3D. Eles formam uma relação entre uma imagem bidimensional e a superfície tridimensional da imagem que será aplicada (MICROSOFT, 2015; CHRONSITER, 2015). A Figura 11 representa um mapa de textura. A Figura 12 ilustra este mapa de textura aplicado a superfície de um objeto.

Figura 11 - Mapa de textura.



Fonte: elaborada pelo autor.

Figura 12 - Objeto 3D com textura.



Fonte: elaborada pelo autor.

Uma superfície NURBS, *Non-Uniform Rational B-Splines*, ou curvas básicas racionais não uniformes, é uma malha lisa definida por uma série de curvas contendo pelo menos dois pontos de controle, as quais são ligadas as curvas polinomiais. Esta superfície lisa é convertida em polígonos em vez de tornar, de modo que as superfícies NURBS pode conter um número arbitrário de polígonos. NURBS pode ser convertido em polígonos ou subdividido em superfícies e são úteis para a construção de formas orgânicas (VAUGHAN, 2012; PIEGL e TILLER, 1997). *Spline* é função numérica que é definida por funções polinomiais, e que possua um grau suficientemente elevado de suavidade nos locais onde as polinomiais conectam, conhecido como nós (FARIN, 2001). A Interpolação de uma Spline é muitas vezes preferido sobre interpolação polinomial porque o erro de interpolação pode ser minimizado, mesmo quando usando polinômios de baixo grau para o *spline*. (FARIN, 2001)

A função B-spline é a função de base interpolativo maximamente diferenciável. O *B-spline* é uma generalização da curva de Bezier, a *B-spline* sem nós interiores é uma curva

Bezier. *B-splines* são definidas pela sua ordem 'm' e número de nós interiores 'N', onde o grau não depende do número de pontos de controle. (PIEGL e TILLER, 1997). A curva de bézier é uma curva paramétrica freqüentemente usada em computação gráfica e áreas afins para modelar curvas suaves. Curva de Bezier também pode ser definido como *uma B-spline* sem nós internos, onde o grau polinomial é diretamente relacionado com o número de pontos de controle. (PIEGL e TILLER, 1997)

2.3.2 Modelagem 3D

De acordo com o Vaughan (2012), existem diversas técnicas para serem levados em consideração na hora de modelagem de objetos 3D. Aqui serão apresentados algumas dessas técnicas de forma sucinta.

2.3.2.1 Modelagem por software CAD

Existem muitos softwares modeladores utilizados na indústria. Esses softwares são conhecidos como softwares de CAD/CAM. CAD vem do inglês computer-aided Design, Design assistido pelo computador, e CAM vem do inglês Computer-aided Manufacturing, manufatura assista pelo computador. CAD pode ser definido como o uso do computador para auxiliar na criação, modificação, análise, ou otimização de um design. O computador precisa de um software e hardware especializado para realizar as funções requisitados pela respectiva empresa por trás do CAD. (GROOVER, 2006)

Existem softwares voltados para diversas áreas da indústria como a indústria automobilística, cinema e jogos, arquitetura, engenharia elétrica, médica, etc.

Alguns exemplos são: Maya, AutoCAD, Softimage e 3DS Max da AUTODESK; SketchUp da Google, Cinema 4D da Maxon, Houdini da Side Effects, LightWave da NewTek, ZBrush, entre outros. A vantagem desses softwares é que pode-se criar modelagens profissionais e específicas, com alto grau de detalhes. Em contrapartida, normalmente os softwares tem alto grau de complexidade e exige experiência do usuário. A maioria destes softwares possui as seguintes funcionalidades:

- **Building Out** (Contruir ao próximo): Esta técnica envolve a construção de detalhes terciárias de um modelo a partir do primeiro polígono criado. Uma vez que uma parte

da malha é completa, você constrói em direção a uma outra área do modelo e continua esse processo até que toda a malha é construída;

- **Point by Point** (Ponto por Ponto): Com esta técnica, começa-se gerando pontos para definir a forma da malha que pretende produzir e, em seguida, cria-se polígonos a partir desses pontos;
- **Edge Extend** (Extensão de Aresta): Um método de modelagem que geralmente começa com a criação de um polígono plano. Uma vez que o polígono plano foi criado, uma aresta é selecionada e estendida para produzir um novo polígono;
- **Primitive Modeling** (Modelagem Primitiva): É uma combinação de várias formas geométricas primitivas (como caixas, esferas, discos, etc.) e a alteração da sua forma, para formar o objeto final desejado;
- **Box Modeling** (Modelagem através de caixas): Em vez de usar várias primitivas para gerar a malha final, o modelador "gera" uma geometria adicional da geometria primitiva para criar toda malha. Esta geometria adicional é criado através da extensão de grupos de polígonos, obtendo dessa forma mais áreas geométricas para moldar o modelo;
- **Patch Modeling** (Modelagem por pedaços): Utiliza as curvas NURBS, é criado a superfície de um objeto em curvas com espaço entre as mesmas. As superfícies criadas entre as curvas são conhecidos como pedaços. Esses pedaços são controlados pelos pontos que compõem as curvas e são comumente chamados de pontos de controle;
- **Digital Sculpting** (Esculpir digitalmente): Esculpir digitalmente é um método de modelagem que é o mais próximo que um artista pode chegar a escultura tradicional. O modelador manipula uma malha de base usando um sistema baseado em pincel que permite a criação de malhas. A malha de base pode ser qualquer coisa, desde uma simples bola primitiva a um objeto que consiste em qualquer número de polígonos;
- **3D Scanning** (Digitalização 3D): A digitalização 3D permite a coleção de dados da superfície de um objeto do mundo real. Esta informação é gravada e então convertida em uma malha digital, geralmente constituída por milhões de pontos (vértices). Esta tecnologia é usada em toda a indústria do mercado como a de visualização médica, cinema, jogos, design industrial, e muito mais. Essa digitalização pode ser feita através de dois tipos de scanners, os de contato e os sem contato.

2.3.2.2 Modelagem por Scanners

Do acordo com Boehler e Marbs, (2002), o ponto de vista de um usuário sobre um scanner 3D é qualquer dispositivo que recolhe coordenadas 3D de uma determinada região de uma superfície de um objeto:

- automaticamente e em um padrão sistemático;
- a uma taxa elevada (centenas ou milhares de pontos por segundo);
- alcançando os resultados (ou seja, coordenadas 3D) em (quase) tempo real.

O scanner pode ou não pode entregar valores de refletividade para os elementos de superfície digitalizados, além das coordenadas 3D. Valores de reflexão da luz, LRV, é uma medida da quantidade total de luz refletida da superfície. Para a estética com variação significativa da cor, o valor reportado é uma média que representa o tom geral. A LRV pode ser utilizada para determinar o contraste visual entre dois materiais diferentes, onde é importante que um objeto seja visivelmente distinto. (BOEHLER e MARBS, 2002)

Algumas formas de utilização de Scanners 3D são:

- estacionária numa posição fixa (por exemplo, em linhas de produção para controle de qualidade);
- como sistemas móveis em tripés ou carrinhos para aplicações de perto e média distância;
- como sistemas de bordo para aplicações topográficas.

Vaughan (2012) diz que existem dois tipos de scanners:

- Scanners com contato: O scanner necessariamente precisa tocar fisicamente o objeto que está sendo digitalizado, normalmente pela ponta do scanner. Sensores, físicos ou por visão computacional, são utilizados para calcular a posição precisa em 3D da ponta do scanner em qualquer ponto tocado no objeto. A informação registrada é então traduzido em pontos de dados e processadas, e é criada uma malha 3D.
- Scanners sem contato: Também chamados de scanners ativos, emitem luz, raios-X ou ultra-som para capturar os dados de superfície do objeto que está sendo digitalizado. A luz ou radiação emitida a partir do scanner é refletida fora do objeto que está sendo

digitalizado e enviado de volta para o scanner, é então gravado a distância do scanner até aquele ponto na superfície.

Em um levantamento feito por Boehler e Marbs (2002), foram estudados dois tipos de scanners sem contato, os *ranging* scanners, calculam a distância entre o scanner e o objeto, e os *triangulation* scanners, calculam a distância a partir de uma triangulação de um ponto utilizando dois, ou mais, dispositivos.

Os *ranging* scanners são compostos por diversos tipos, como os de *Time of Flight*, *ToF* ou tempo de vôo; *Phase Comparison method*, método de comparação de fases; e o *Structured Light*, luz estruturada.

- ***ToF***: Um sensor *ToF* é consistido, muitas vezes, por uma fonte de luz modulada, como um laser, ou LED, uma matriz de pixels, cada um capaz de detectar a fase da luz de entrada, e um sistema óptico comum para focar a luz sobre o sensor. A luz é dada uma modulação rápida, ligando e desligando a fonte de luz. Neste sistema, o tempo, que é exigido pelo sinal de sondagem para viajar até o alvo e voltar para o receptor, ou matriz de pixels, é multiplicado pela velocidade do sinal no meio de propagação, o que retorna a distância ao objeto (GOKTURK, YALCIN, BAMJI, 2004). Outros tipos de *ToF* são os *SONAR*, *RADAR*, *LIDAR*, *SODAR* e *LADAR*.
- ***SONAR***: *SOund Navigation And Ranging*, é uma técnica que utiliza a propagação do som (normalmente subaquático) para navegar, comunicar com ou detectar objetos sobre ou sob a superfície da água, tais como outros navios. (SURLYKKE, PEDERSEN, JAKOBSEN, 2009);
- ***RADAR***: (RAdio Detection And Ranging), processo de transmissão, recepção, detecção e o processamento de uma onda electromagnética que reflete a partir de um alvo. Os sinais de RF (rádio frequência)/microondas são transmitidos e a radiação retrodifundida dispersos pelos alvos é medida, permitindo a localização dos alvos pelo tempo de vôo da onda. Velocidade de dispersão das ondas podem ser determinadas pela medição da mudança de frequência dos sinais de retorno. (RICHMOND e CAIN, 2010; VANDE e JOSHUA, 2015);
- ***LIDAR***: *Light Detection and Ranging*, semelhante ao radar na medida em que utiliza medições de tempo-de-voo, mas funciona em uma região diferente do

espectro electromagnético que varia de ultravioleta à luz infravermelha. (VANDE e JOSHUA, 2015);

- **SODAR:** *SOnic Detection And Ranging*, técnica relacionada com RADAR e LiDAR em que as ondas acústicas (em uma variedade de ângulos) são transmitidos e os sinais retroespelhados são detectados, a fim de determinar a velocidade do vento a um alcance a distâncias de até 1.000 m através da análise dos desvios de freqüência Doppler horizontais e verticais do ar em movimento. (VANDE e JOSHUA, 2015);
- **LADAR:** *LAser Detection And Ranging*, uma variação do LiDAR por causa da utilização de laser. (RICHMOND e CAIN, 2010).

- **Comparação de fases:** o feixe transmitido é modulado por uma onda harmônica e a distância é calculada utilizando a diferença de fase entre a onda transmitida e recebida. Os resultados obtidos podem ser mais precisos (à custa da taxa de medição), quando comparado aos resultados do *ToF*. Uma vez que é necessário um sinal de retorno bem definido, scanners, utilizando o método de comparação de fase também podem ter um alcance reduzido e tendem a produzir pontos mais erradas ou descartados (BOEHLER e MARBS, 2002);
- **Structured Light** (Luz Estruturada): Um sistema que é composto por um projetor que emite uma faixa (plano) de luz e uma câmera colocada num ângulo em relação ao projetor. Em cada ponto no tempo, a câmera obtém posições 3D para os pontos ao longo de um contorno 2D traçado sobre o objeto pelo plano de luz. A fim de obter uma imagem completa, é necessário varrer uma faixa ao longo da superfície ou projetar várias faixas. Apesar de a projeção de várias faixas levar a uma aquisição de dados mais rapidamente, é necessário algum método para diferenciar entre as faixas. Existem três principais formas de realizar essa diferenciação: assumir a continuidade da superfície para que as listras projetadas adjacentes estão ao adjacentes na imagem da câmera (PROESMANS e VAN GOOL, 1997), diferenciando as listras com base nas cores (BOYER e KAK, 1987), e codificação das listras variando sua iluminação ao longo do tempo (RUSINKIEWICZ, HALL-HOLT, LEVOY, 2002).

Existem vários tipos de scanners de triangulação, como aqueles com somente uma câmera e aqueles com duas câmeras. Suas principais características são:

- **Solução por uma única câmera:** Este tipo de scanner consiste de um dispositivo de transmissão, o envio de um feixe de laser em, um ângulo alterado de forma incremental definido a partir de uma extremidade de uma base mecânica em relação ao objeto, e uma câmera CCD (*charge-coupled device* ou Dispositivo de carga acoplada), no outro extremo da base, a qual detecta o ponto de laser (ou linha) no objeto. A posição 3D da superfície refletora pode ser derivada a partir do triângulo resultante. A partir daí, é também bem conhecido que a precisão da distância entre o instrumento e objeto diminui com o quadrado desta distância. Estes instrumentos têm um papel importante para distâncias curtas e pequenos objetos onde eles são mais precisos do que os *ranging* scanners;
- **Solução por duas câmeras:** Uma variação do princípio da triangulação é a utilização de duas câmaras CCD, cada uma na outra extremidade da base. O ponto ou padrão que será detectado é gerada separadamente por um projetor de luz, que não tem qualquer função de medição. A projeção pode ser constituída por um ponto ou linha de luz em movimento, ou por um padrão arbitrário estática (BOEHLER e MARBS, 2002).

2.3.2.3 Modelagem baseada em imagens

Photogrammetry, ou fotogrametria, é a ciência de extrair medições confiáveis a partir de imagens bidimensionais (2D), geralmente fotográficas. Esse campo de estudo envolve diversas áreas e disciplinas. Algumas das disciplinas incluem: óptica, geometria projetiva, sensoriamento remoto, e, mais recentemente, de visão computacional. A *photogrammetry* também é uma técnica utilizada para a geração de malhas tridimensionais, utilizando técnicas como *Structure from Motion* (SfM) e *Nonrigid Structure from Motion* (NRSfM). Esse uso envolve estimar coordenadas 3D de um objeto por meio da comparação de várias imagens fotográficas tiradas de posições diferentes. A partir dessas imagens múltiplas, um raio (ou linha) pode ser calculado para pontos 3D em um objeto (FOSTER e HALBSTEIN, 2014; QUAN, 2010; BREGLER, HERTZMANN, BIERMANN, 2000).

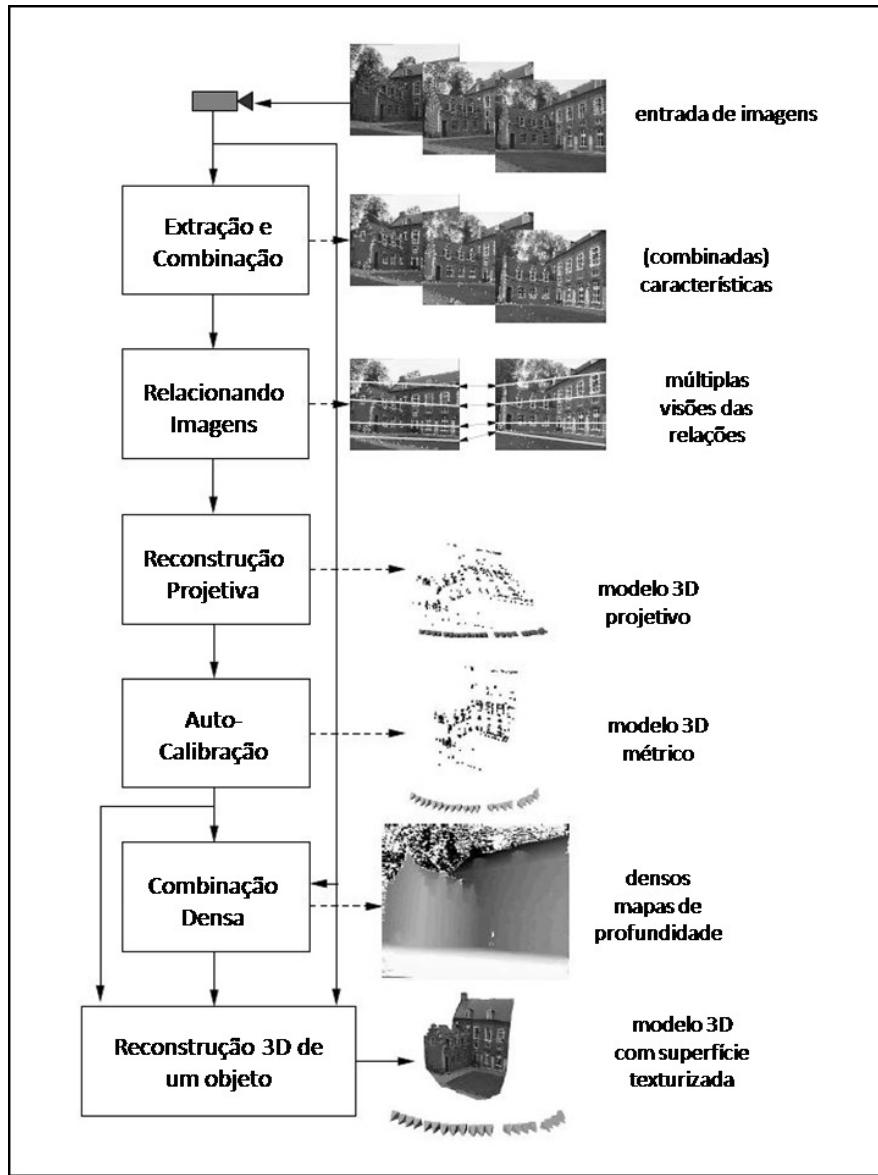
O SfM, estrutura do movimento, designa o processamento das poses de câmera e dos pontos 3D a partir de uma sequência de, pelo menos, duas imagens. É também chamado de reconstrução 3D baseado em fotos. Note-se que tanto a estrutura, que são pontos 3D, e os

movimentos, que são as poses da câmara, são simultaneamente calculadas a partir de pontos de imagem. (QUAN, 2010)

O NRSfM, estrutura não rígida do movimento, é uma área de estudo que designa o processamento das poses de câmera e dos pontos 3D a partir de imagens, sendo que o objeto a ser reconstruído não é rígido ou ele está em movimento. O SfM, descrito anteriormente, é utilizado com objetos rígidos e sem movimento. Algumas aplicações do NRSfM seria na reconstrução de objetos em movimento, animais, pessoas, objetos elásticos, entre outros (BREGLER, HERTZMANN, BIERMANN, 2000; TORRESANI, HERTZMANN, BREGLER, 2008; RABAUD e BELONGIE, 2008).

A Figura 13 apresenta um esquema de como é realizado o processo de reconstrução, ou modelagem, de um objeto real em 3D a partir do SfM.

Figura 13 - Esquema do *Structure from Motion*.



Fonte: (POLLEFEYS, 2002 e adaptada pelo autor)

Esse esquema segue o seguinte encadeamento de etapas:

- Entrada de imagens**: Etapa onde as imagens do mundo real, já capturadas, são inseridas como dados de entrada ao programa/algoritmo;
- Extração e Combinação**: Etapa de extração de características. As características das diferentes imagens são então comparadas, e realizadas combinações entre elas, utilizando medidas de similaridade e listas de possíveis correspondências são então estabelecidas;

C. Relacionando Imagens: Com base nestas listas, as relações entre os diferentes pontos de vistas são computados. Uma vez feito as relações das vistas, a estrutura e as características do movimento da câmera é computado;

D. Reconstrução Projetiva:

- a. Uma reconstrução inicial é então feita para as duas primeiras imagens da sequência. Para as demais imagens a câmera pose, estimativa da rotação e translação da câmera em um espaço 3D, é estimado no quadro definido pelas duas primeiras câmeras;
- b. Para cada imagem adicional nesta fase, as características correspondentes para pontos nas imagens anteriores são reconstruídas, refinadas ou corrigidas.

E. Auto-Calibração: O resultado deste passo é uma reconstrução de, tipicamente, algumas centenas de pontos característicos. Quando as câmeras não calibrados, desconhecendo os parâmetros intrínsecos e extrínsecos do mesmo, são usados na estrutura da cena e do movimento da câmera só é determinado até uma transformação projetiva arbitrária;

F. Combinação Densa:

- a. Neste ponto existe informação suficiente para voltar para as imagens e procurar correspondências para todos os outros pontos de imagem. Esta pesquisa é facilitada uma vez que a linha de visão correspondente a um ponto de imagem pode ser projetada para outras imagens, limitando a amplitude de busca para uma dimensão;
- b. Por pré-deformação da imagem, processo chamado de retificação, algoritmos de *photogrammetry* podem ser utilizados. Este passo permite encontrar características que correspondem para a maioria dos pixels nas imagens;
- c. A partir dessas correspondências a distância entre os pontos até o centro da câmera podem ser obtidos através da triangulação. Estes resultados são refinados e concluído pela combinação das características a partir de múltiplas imagens;

G. Reconstrução 3D de um objeto:

- a. Os resultados são integrados em uma reconstrução da superfície texturizada 3D da cena em questão;

- b. O modelo é obtido através da aproximação do mapa de profundidade, com um visualização do tipo *triangular wireframe*. A textura é obtida a partir das imagens e mapeada na superfície;

Na etapa de entrada de imagens, existem dois tipos de entradas de imagens que podem ser aceitas, o que levam a diferentes algoritmos que podem ser utilizados para obter uma reconstrução ótima (QUAN, 2010):

1. **Forma ordenada:** Onde a entrada é composta por uma sequência de imagens, onde as imagens estão organizadas de forma sequencial em relação ao movimento da câmera relativo ao objeto em questão que deseja reconstruir. Dois algoritmos utilizáveis são o *Sparse SfM*, quando a sequência toda está disponível, e o *Incremental sparse SfM*, quando é aceitada novas imagens de forma incremental. O objetivo desses algoritmos é, dada uma sequência de imagens sobrepostas, calcular-se uma reconstrução das matrizes da câmera e os pontos correspondentes da sequência;
 - a. ***Sparse SfM*:** Nesse algoritmo é necessário decompor a sequência de n imagens em consecutivos $n-1$ pares $(i, i+1)$ e trios $(i, i+1, i+2)$. Utilizando os pares, computa-se uma lista de correspondências de pontos em uma matriz fundamental, e utilizando os trios, computa-se as matrizes de projeções. Reconstruir uma sequência maior $(i..j)$ mediante a fusão de duas sequências menores $(i..k+1)$ e $(k..j)$, com dois quadros sobrepostos, k e $k+1$, onde k é o frame mediano do intervalo $(i..j)$.
 - b. ***Incremental sparse SfM*:** Nesse algoritmo é necessário primeiro inicializar uma reconstrução 3D com base nas primeiras duas ou três imagens, onde é computado uma lista de pontos correspondentes e as matrizes fundamentais ou uma reconstrução projetiva de três câmeras e dos pontos correspondentes, com base na escolha por utilizar duas ou três imagens respectivamente. Para cada nova imagem i na sequência, é calculado os pontos correspondentes, utilizando a técnica de três imagens, com o trio de imagens $i-2, i-1$ e i . É feito uma mescla do trio com a reconstrução 3D atual, estimando uma transformação espacial. É

realizado um ajuste na sistema todo atual de câmeras e pontos.

2. **Forma desordenada:** Onde a entrada é composta por uma coleção de imagens, onde as imagens não estão naturalmente ordenadas em uma sequência linear em relação ao movimento da câmera relativo ao objeto em questão que deseja reconstruir. Neste caso é utilizado um algoritmo chamado *unstructured sparse SfM*, que é uma extensão do *incremental sparse SfM*. Nesse algoritmo é necessário a geração de todos os pares de imagens, utilizando o método *SIFT* e o método *Nearest Neighbor*. É inicializado uma reconstrução 3D do melhor par na coleção, esse critério é determinado pelo número de pontos correspondentes; o *baseline*; e a configuração coplanar degenerada. É então adicionada a imagem mais adequada a reconstrução 3D, cujo parâmetro é o número de pontos correspondentes comuns. É realizado um ajuste no sistema todo atual de câmeras e pontos (SNAVELY, 2006; QUAN, 2010).

O *Nearest Neighbor query*, busca pelo vizinho mais próximo, é a mais comum de busca em relação à distância. Dado um ponto de busca P e um inteiro positivo k, a busca retorna os k objetos que são mais próximos de P, com base numa distância métrica (por exemplo, a distância euclidiana). (PAPADOPoulos, 2006)

A matriz fundamental é uma matriz singular, que representa o movimento da câmera não calibrada. (TORR e MURRAY, 1997; QUAN, 2010)

A restrição de coplanaridade é quando, geometricamente, dado um ponto de imagem no primeiro ponto de vista, este primeiro ponto projeta-se em uma linha no espaço, esta linha é re-projetada para o segundo ponto de vista como uma linha ao longo da qual todos os potenciais pontos correspondentes do ponto dado no primeiro vista estão localizados. De forma equivalente, isto é dizer que os pontos correspondentes em dois pontos de vista e o ponto no espaço estão no mesmo plano. (QUAN, 2010)

A baseline é uma linha imaginária que une os dois centro focais de duas câmeras ou imagens . (HARTLEY, 2004)

Na etapa de Extração e Combinação, dois algoritmos utilizáveis são o *SIFT (Scale Invariant Feature Transform)* e o *SURF (Speeded Up Robust Features)*. (GOVENDER, 2009)

O algoritmo *SIFT* transforma os dados de uma imagem em coordenadas de escala invariante em relação a características locais. Uma técnica utilizada na detecção de características em uma imagem. A seguir estão as principais fases do *SIFT* utilizado para gerar o conjunto de características da imagem:

1. **Detecção escala-espaco:** Pesquisa em todas as escalas e locais da imagem. Implementado de forma eficiente usando a função de Diferenças das Gaussianas para identificar pontos de interesse em potencial que são invariantes à escala e orientação;
2. **Localização de pontos-chave:** Em cada local candidato, um modelo detalhado está apto para determinar a localização e escala. Os pontos-chaves são selecionados com base em medidas de sua estabilidade;
3. **Atribuição de orientação:** Uma ou mais orientações são atribuídas a cada local de ponto-chave com base nas direções de gradientes locais das imagens. Todas as operações futuras são realizadas em dados de imagem que foram transformados em relação à orientação, escala e localização atribuída a cada característica, proporcionando assim invariância a essas transformações;
4. **Descriptor de pontos-chave:** Os gradientes locais da imagem são medidos, utilizando um histograma, na escala selecionada na região em torno de cada ponto-chave. Estes dados são transformadas e armazenados os seus intervalos em vetores de dimensão 128 (8 intervalos de orientação para cada um dos intervalos de localização 4×4), o que permite níveis significativos de distorção da forma local e as alterações na iluminação.

Um aspecto importante na abordagem *SIFT* é o grande número de características que densamente cobrem a imagem ao longo de toda a gama de escalas e locais. Uma imagem de tamanho 500x500 pixels dará origem a cerca de 2000 características estáveis (embora este número depende tanto do conteúdo da imagem e das escolhas para vários parâmetros). (LOWE, 2004)

O algoritmo *SURF* é um algoritmo que detecta e descreve um ponto de interesse, uma possível característica, com base na escala e rotação do mesmo. É inspirado pelo algoritmo

SIFT. A seguir estão as principais fases do *SURF* utilizado para gerar o conjunto de características da imagem . (GOVENDER, 2009; BAY, TUYTELAARS, VAN GOOL, 2006):

1. Pontos de interesse são selecionados em locais distintos na imagem, tais como *blobs*, regiões que diferem de outras regiões em propriedades, como brilho ou cor, e junções T.
2. A vizinhança de cada ponto de interesse é representado por um vetor de características. Este descritor tem que ser diferente e, ao mesmo tempo, robusta ao ruído, os erros de detecção, e deformações geométricas e fotométricas.
3. Os vetores de descrição são combinados entre as diferentes imagens. A combinação é muitas vezes baseada em uma distância entre a vetores, por exemplo, a distância Euclidiana.

A transformação projetiva é uma transformação entre dois planos que é representada como uma matriz 3x3 agindo em coordenadas homogêneas. Esta transformação demonstra os efeitos compostos de rotação 3D rígida e translação do plano real (parâmetros extrínsecos da câmera), a perspectiva de projeção para o plano da imagem, e uma transformação da imagem final (que abrange os efeitos da mudança de parâmetros intrínsecos da câmera). (FELDMAR e AYACHE, 1997; HARTLEY, 1999; ORRITE e HERRERO, 2004)

Os parâmetros intrínsecos da câmera são aqueles parâmetros internos e específicos para cada câmera. Por exemplo, o comprimento focal, deslocamento do ponto principal e a inclinação dos eixos. Os parâmetros extrínsecos da câmera denotam a sua posição no mundo real, ou a transformações do sistema de coordenadas do mundo 3D em coordenadas 3D da câmera. Isso significa que o vetor de translação e rotação entre matriz que relaciona o sistema de coordenadas do mundo precisam ser obtido relativos ao sistema de coordenadas da câmera. (ZHANG, 2004; TAN, SULLIVAN, BAKER, 1995)

Na etapa Relacionando Imagens, após a detecção de pontos, seja pelo método *SURF*, *SIFT* ou outro, é necessário realizar comparações destes pontos entre pontos em outras imagens para obter os pontos que combinam. Alguns métodos são:

1. **Combinação de Template:** Algoritmos tentam correlacionar os níveis de cinza da imagem nos pontos de vistas considerados, assumindo que eles apresentam alguma semelhança. (GOSHTASBY, GAGE, BARTHOLIC, 1984; CHOU e CHEN, 1990)
2. **Combinação de Características:** Algoritmos extraem pontos de destaque das imagens, como segmentos de borda ou contornos, e combiná-los em dois ou mais pontos de vista. Uma imagem pode então ser descrita por um gráfico com primitivas definindo os nós e as relações geométricas de ligações. O registro de dois mapas torna-se o mapeamento dos dois grafos: uma subgrafo de isomorfismo. Técnicas mais comuns são a pesquisa em árvore, relaxamento, etc. (SHAPIRO e HARLICK, 1981)
3. **Combinação através da Correlação:** Dado um ponto de interesse m_1 na imagem 1, pontos na imagem 2, m_2 , são procurados por meio de uma janela de correlação, uma área da procura na imagem 2 relacionada ao posicionamento do ponto m_1 da imagem 1. Para determinar se as janelas de correlação, os pontos de m_1 e m_2 , são idênticos, uma pontuação de correlação é calculado. A pontuação varia de -1, pontos que são totalmente distintos, para 1, pontos que são idênticos. (ZHANG et al., 1995)

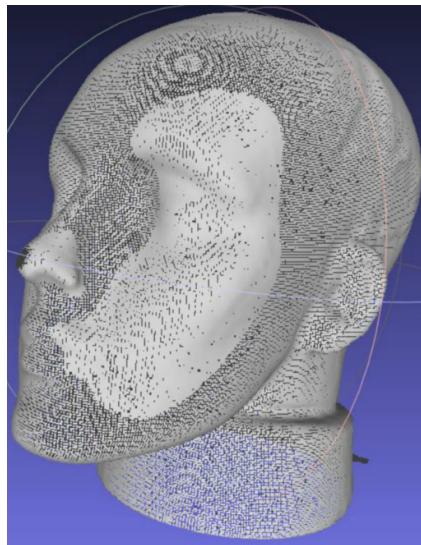
Na etapa de Reconstrução Projetiva, a reconstrução do objeto de interesse é realizado iniciando a estrutura do objeto utilizando duas imagens e depois utilizando o posicionamento e rotação da câmera, a pose da câmera, como critério para a reconstrução dos demais pontos 3D. O resultado desta etapa é a geração de *Point Clouds* (Nuvens de Ponto).

1. **Estrutura e Movimento inicial:** É necessário que características suficientes são combinadas entre estas imagens, por outro lado, as imagens não devem ser demasiado próximas umas das outras, de modo a que a estrutura inicial é bem condicionado, pois imagens muito próximas podem conduzir a uma aceitação de imagens novas de forma tendenciosa, por exemplo reconstruindo somente parte do objeto ou do fundo. A distância baseada na imagem é a distância média entre os pontos transferidos através de uma média entre a homografia plana e os pontos correspondentes na imagem de destino. Com a seleção destas imagens, é possível calcular suas matrizes de projeção (LI e WANG, 2014; POLLEFEYS, 2002; WU, 2013);

- a. Frame inicial, Duas imagens da sequência são usados para determinar um quadro de referência. A quadro do mundo real está alinhada com a primeira câmara. A segunda câmara é escolhida de modo que a geometria epipolar corresponde à matriz F , matriz fundamental.
 - b. Inicializando a Estrutura, Após a obtenção das matrizes de projeção inicial, as características serão reconstruídas através de triangulação.
2. **Atualizando a estrutura e movimento:** Etapa que determina como adicionar um ponto de vista a uma reconstrução já existente. Primeiro a pose da câmera é determinada, depois a estrutura é atualizada baseada no ponto de vista adicionado e então novos pontos são usados como base;
- a. Estimação da pose de projetiva, Para cada ponto de vista adicional uma pose da câmera em relação a reconstrução pré-existente é determinada, em seguida, a reconstrução é atualizada. O primeiro passo consiste em encontrar a geometria epipolar. Em seguida, os resultados que correspondem a pontos já reconstruídos são utilizados para inferir correspondências entre 2D e 3D. Com base nisto a matriz de projeção é determinada. Uma vez que esta matriz foi determinada a projeção de pontos já reconstruídas pode ser previsto. Isto permite encontrar algumas combinações adicionais para refinar a estimativa da matriz de projeção.
 - b. Relacionando a outros pontos de vista, O procedimento anterior apenas relaciona a imagem nova com a imagem anterior. Assume-se implicitamente que uma vez que um ponto fica fora de vista, ele não vai voltar. Suponha que um ponto 3D específico saiu de vista, mas que é visível novamente nos dois últimos pontos de vista. Neste caso, um novo ponto 3D será instanciado. Este ponto pode não causar problemas imediatamente, mas uma vez que estes dois pontos 3D são independentes do restante do sistema, nada obriga a suas posições a corresponderam. Para as sequências mais longas, onde a câmera é movida para trás e para frente sobre a cena, isso pode levar a resultados ruins devido a erros acumulados. (POLLEFEYS, 2002; WU, 2013)

Um *Point Cloud*, ou nuvem de pontos, é um conjunto de dados representados por pontos, estes pontos são geralmente definidas por coordenadas X, Y e Z, e muitas vezes têm a intenção de representar a superfície externa de um objeto (RUSU e COUSINS, 2011; LINSEN, MÜLLER, ROSENTHAL, 2001). A Figura 14 ilustra uma nuvem de pontos.

Figura 14 - Nuvem de pontos.



Fonte: elaborada pelo autor.

A geometria epipolar é a geometria da visão estéreo. Quando duas câmaras observam uma cena 3D a partir de duas posições distintas, há um certo número de relações geométricas entre os pontos 3D e as suas projeções para imagens 2D que levam a restrições entre os pontos de imagem. (ZHANG, 1998)

A homografia plana é uma transformação projetiva entre os pontos correspondentes nos dois planos. (ZHANG, 1998)

Na etapa de Auto-Calibração, a transformação projetiva arbitrária é transformada em uma métrica. A reconstrução obtido a partir de etapas anteriores só é determinado, sua precisão, até uma transformação projetiva arbitrária. Para definir essas informações para escalas adequadas, é preciso um método para transformar a reconstrução arbitrária em uma métrica real. Em geral três tipos de restrições podem ser aplicadas para atingir este objetivo: restrições de cena, restrições de movimento de câmera e restrições sobre os intrínsecos da câmera. Alguns métodos de calibração são:

1. **Conhecimento da Cena:** O conhecimento das distâncias e ângulos em cena (relativos) pode ser usado para obter informação sobre a estrutura métrica do objeto. Uma das formas de calibrar o local a um nível métrico é através do conhecimento métrico da posição relativa de 5 ou mais pontos na cena. (BOUFAMA, MOHR, VEILLON, 1993);
2. **Conhecimento da Câmera:** Conhecimento sobre a câmera também pode ser usado para transformar a reconstrução projetiva para o sistema métrico. Diferentes parâmetros da câmara podem ser conhecidos, como os parâmetros intrínsecos e extrínsecos. (CLARKE e FRYER, 1998; FOSTER e HALBSTEIN, 2014);
3. **Auto-Calibração:** Reduzindo a ambigüidade sobre a reconstrução, impondo restrições sobre os parâmetros da câmara intrínsecas. Essa auto-calibração pode ser feita através de métodos *Absolute Dual Quadric* (TRIGGS, 1997) ou pesos as varáveis. (THORMÄHLEN, BROSZIO, MIKULASTIK, 2006).

Na etapa de Combinação Densa, com a câmara de calibração determinada para todos os pontos de vista da sequência, algoritmos de *SfM* calibrados podem ser utilizados. Os algoritmo de combinação de características já oferecem um modelo de superfície esparsa com base nos pontos característica distintas, este, contudo, não é suficiente para reconstruir modelos geometricamente corretos. Esta tarefa é realizada por uma combinação de disparidade densa que estima combinações a partir de níveis de cinza das imagens diretamente através da exploração de restrições geométricas adicionais. O resultado desta etapa são *depth maps* (mapas de profundidade). Alguns desses métodos são:

1. ***Image pair rectification*** (retificação de imagens pares): Esta etapa consiste em transformar as imagens de modo que as linhas epipolares são alinhados horizontalmente. Neste caso algoritmos de combinação estéreo podem utilizar a restrição epipolar e reduzir o espaço de busca para uma dimensão. O esquema de rectificação tradicional consiste em transformar os planos de imagem para que os planos espaciais correspondentes são coincidentes. (PAPADIMITRIOU e DENNIS, 1996; LOOP e ZHANG, 1999);

2. **Stereo Matching** (Combinações Estéreo): Os métodos propostos podem ser classificados em combinação de características e combinação através da correlação (como visto na Etapa Relacionando Imagens);

O *depth map*, ou mapa de profundidade é uma imagem que contém informação relativa à distância das superfícies de objetos de cena a partir de um ponto de vista. (MALIK, 2011)

Na etapa de Reconstrução 3D de um objeto, as informações necessárias para a reconstrução de um modelo 3D, que já foram obtidas através das etapas anteriores, serão utilizadas para a geração do modelo tridimensional realísticos. Existem alguns tipos de modelos que podem ser gerados a partir desta reconstrução, porém somente o *Surface Reconstruction* será tratado.

Surface Reconstruction (Reconstrução da superfície): Para que esta reconstrução seja iniciado, há a necessidade de ter uma densidade alta na nuvem de pontos de forma que seja suficiente para reconstruir uma superfície lisa de topologia arbitrária, o que varia de objeto o objeto. Os pontos 3D reconstruídos são primeiramente segmentados para o objeto em primeiro plano e o fundo. O fundo inclui valores atípicos óbvios como pontos isolados e distantes da maioria dos pontos. Os pontos do objeto de primeiro plano são obtidos como o maior componente ligado a vizinhança de todos os pontos do grafo de tal modo que a distância entre quaisquer dois pontos deste grafo deve ser menor do que um múltiplo da mediana de incerteza dos pontos. As silhuetas de objetos são extraídos de forma interativa com cada imagem de entrada. Uma abordagem simples consiste em sobreposição de uma malha triangular 2D na parte superior da imagem e, em seguida, construir uma malha 3D correspondente colocando os vértices dos triângulos no espaço 3D de acordo com os valores encontrados no mapa de profundidade.

Alguns métodos para gerar a reconstrução da superfície são: *Poisson Reconstruction* (KAZHDAN, BOLITHO, HOPPE, 2006), *Delaunay triangulations* (BOISSONNAT, 1984; KOLLURI, SHEWCHUK, O'BRIEN, 2004), *alpha shapes* (EDELSBRUNNER e MÜCKE, 1994; BERNARDINI et al., 1999), *Voronoi diagrams* (AMENTA, BERN, KAMVYSELIS, 1998; AMENTA, CHOI, KOLLURI, 2001), entre outros.

2.3.2.4 Uso do SfM na indústria

Weta Digital, criadoras de filmes Como *The Hobbit*, *Dawn of the Planet of the Apes*, *Senhor dos Anéis*, *Homem de Ferro*, *Prometheus*, entre outros, escreveu um artigo sobre a utilização de Geração de modelos 3D a partir de fotos. Neste artigo disseram que a digitalização usando a fotogrametria requer muito menos entradas do usuário do que os outros métodos e acelera a linha de produção deles, que é então utilizado pelos seus artistas para criar os modelos finais em um ritmo muito mais rápido e fotorealista.

Descrevem o *photogrammetry* como a combinação das melhores características de múltiplas abordagens. No artigo é descrito o fluxo desse trabalho como tendo três partes: (1) a sessão de captura, (2) sessão de processamento da *photogrammetry*, e (3) sessão de geração de referência. As imagens são captadas, que então são processadas e, em seguida, os modelos tridimensionais que são gerados são entregues a artistas 3D para remodelação de modo a torná-los eficiente para uso no momento de renderização. (BHAT e BURKE, 2011)

Rendering, ou renderização é o processo final de criação da imagem 2D real ou animação da cena preparada. Renderização 3D é o processo de converter automaticamente cenas 3D em imagens 2D com efeitos 3D fotorealistas em um computador. (DOBBINS, 2012)

A técnica de *photogrammetry* também é utilizada na composição de imagens panorâmicas de alta qualidade (SILVA, 2012), monitoramento de estruturas de engenharia civil (SILVA, 2012), documentação arqueológica e conservação do patrimônio cultural (SILVA, 2012), arqueologia subaquática (SILVA, 2012), entre muitas outras.

3 MATERIAIS E MÉTODOS

Neste capítulo será abordado os métodos utilizados para a realização deste trabalho, bem como uma relato dos materiais utilizados para o desenvolvimento do mesmo. O texto é dividido em duas subseções, materiais e métodos, com suas respectivas divisões.

3.1 Materiais

O local dos experimentos, tanto para a análise das funcionalidades do ARSTUDIO quanto para a análise da modelagem de objetos virtuais, foi utilizado o laboratório SACI (Sistema Adaptativo e Computação Inteligente) no Central de Laboratórios de Pesquisa II no campus da Unesp de bauru. Além do espaço físico deste laboratório para a realização dos experimentos, também foram utilizados os computadores e dispositivos que este laboratório oferece.

Para os experimentos envolvendo o ARSTUDIO, foram utilizados um computador, uma câmera webcam de alta resolução, um dispositivo Kinect v1, uma televisão, dois tripés de suporte para câmera, dois tripés para suporte de iluminação, duas lâmpadas para fotografia, dois kits de iluminação e um kit de tela verde.

Para os experimentos envolvendo a geração de conteúdo 3D, foram utilizados um computador, um dispositivo Kinect v2, um *tablet*, uma câmera profissional de filmagem, um scanner 3D, uma televisão, dois tripés para câmera, dois tripés para iluminação, duas lâmpadas para fotografia, dois kits de iluminação e uma plataforma giratória.

As especificações do computador utilizado foram:

- Disco Rígido: de 2TB (7200RPM);
- Processador: Intel Core i7-4770 CPU @ 3.40 GHz;
- Memória RAM: 12.0 GB DDR3;
- Sistema Operacional: Windows 8.1 Pro-64bit;
- Mother Board: ASUS H81M-A/BR;
- BIOS: American Megatrends Inc., v 20.01;
- Placa de vídeo: NVIDIA GeForce GTX 650;

Os outros dispositivos utilizados foram:

- Logitech HD Pro Webcam C910;
- Microsoft Kinect Xbox 360 (modelo 1414);
- Microsoft Kinect for Windows v2 (modelo 1656);
- Samsung Galaxy Note 10.1 2014 Edition (modelo SM-P601);
- Panasonic AVCCAM HD (modelo AG-AC8);
- Sense 3D Scanner (modelo 3DS391230);
- LG HD 3D Smart TV 42" (modelo 42LB6500);
- Weifeng Tripod (modelo WT3730);
- Velbon Tripod (modelo DV-7000);
- Tripé para Iluminação (modelo Greika W-806);
- Photo Bulb E27 125W;
- Greika Softbox;
- Plataforma giratória.

A Figura 15 e a Figura 16 retratam os equipamentos utilizados na geração de conteúdos com o sistema ARSTUDIO e a Figura 17 e a Figura 18 ilustram os equipamentos utilizados na análise da modelagem de objetos virtuais.

Figura 15 - Equipamentos utilizados com o sistema ARSTUDIO.



Fonte: Elaborada pelo autor.

Figura 16 - Outros equipamentos utilizados com o sistema ARSTUDIO.



Fonte: Elaborada pelo autor.

Figura 17 - Equipamentos utilizados na modelagem de objetos virtuais.



Fonte: Elaborada pelo autor.

Figura 18 - Outros equipamentos utilizados na modelagem de objetos virtuais.



Fonte: Elaborada pelo autor.

3.2 Métodos

O desenvolvimento do trabalho foi feito em duas etapas maiores, um estudo das funcionalidades e limitações do ARSTUDIO e a geração de uma análise da modelagem de objetos virtuais, que foram divididas em etapas menores, cada uma visando alcançar os objetivos definidos para o mesmo. Além dessas etapas, foram necessárias duas etapas, uma de pesquisa e uma de reconstrução do ambiente do estúdio virtual. Essas etapas foram:

- A. Levantamento bibliográfico
 - A.1. Estúdios Virtuais
 - A.2. Modelagem 3D
- B. Reconstrução do ambiente do ARSTUDIO

B.1. Ambiente de Software e Hardware

B.2. Ambiente Físico

C. ARSTUDIO

C.1. Geração de um conteúdo piloto utilizando o ARSTUDIO

C.2. Análise do conteúdo gerado

C.2.1. Levantamento das funcionalidades oferecidas

C.2.2. Dificuldades na geração do conteúdo

C.2.3. Levantamento das limitações presentes

C.3. Proposta de novas funcionalidades

D. Técnicas de geração de objetos tridimensionais

D.1. Técnicas estudadas

D.2. Experimentos com Câmeras RGB-D

D.2.1. Objeto fixo e câmera em movimento

D.2.2. Câmera fixa e objeto em movimento

D.2.3. Análise dos resultados a partir da câmera RGB-D

D.3. Scanner 3D

D.3.1. Experimentos utilizando o Scanner 3D

D.3.2. Análise dos resultados a partir do scanner

D.4. *Structure from Motion*

D.4.1. Observações iniciais

D.4.2. Ambiente interno com características ao redor do objeto

D.4.3. Ambiente interno com características no objeto

D.4.4. Análise dos resultados a partir do *Structure from Motion*

D.5. Considerações Finais sobre os experimentos

3.1.1 *Levantamento Bibliográfico*

Neste seção será elaborada o modo como as pesquisas, levantamento de informações e base teórica, foram realizadas, tanto na área de estúdios virtuais quanto na área de modelagem 3D. Os resultados destas pesquisas se deram na forma de conceitos fundamentais e funcionalidades de estúdios virtuais, também do ARSTUDIO, bem como os conceitos

fundamentais e diferentes métodos para a modelagem 3D, ambos apresentado no segundo capítulo desta monografia.

3.1.1.1 Estúdios Virtuais

O levantamento bibliográfico para esta monografia, no aspecto de estúdios virtuais, foi baseado em livros obtidos através da Biblioteca da UNESP, Campus Bauru e artigos científicos encontrados em bases de dados como IEEEXplore *Digital Library*, ACM *Digital Library*, *Sensors*, *Optics InfoBase*, CiteSeerX, *ScienceDirect*, Springer *Link*, SPIE, *World Scientific*, *Nature*, e Google *Scholar*. Boa parte das buscas realizadas nestas bases de dados foram por palavras chaves e operadores lógicos. Por exemplo, quando se buscou a obtenção de materiais relacionados com o aspecto da câmera em estúdios virtuais, as buscas foram feitas com a seguinte *string* de busca ou uma variação da mesma:

((*"Virtual Studio"*) OR (*"Virtual Studios"*) OR (*"Virtual TV Studio"*) OR (*"Virtual TV Studios"*) OR (*"Pre-Visualization"*)) AND ((*"Camera"*) OR (*"Cameras"*))

Foi realizado uma extensa leitura dos artigos e livros na área de estúdios virtuais a fim de encontrar fundamentos teóricos para esta monografia. Outro instrumento de coleta de dados foi a leitura das monografias de anos anteriores a respeito de estúdios virtuais, sendo o foco no desenvolvimento do ARSTUDIO, e conversas com os seus respectivos autores.

3.1.1.2 Modelagem 3D

O levantamento bibliográfico para esta monografia, no aspecto de modelagem 3D, foi baseado em livros obtidos através da Biblioteca da UNESP, Campus Bauru e artigos científicos encontrados em bases de dados, nas mesmas bases citadas anteriormente. Boa parte das buscas realizadas nestas bases de dados foram por palavras chaves e operadores lógicos. Por exemplo, quando se buscou a obtenção de materiais relacionados a aplicações envolvendo métodos de modelagem com base em fotos, as buscas foram feitas com a seguinte *string* de busca ou uma variação da mesma:

((*"Photogrammetry"*) OR (*"Structure from Motion"*) OR (*"Image Based Modeling"*)) AND (*"Application"*)

Uma extensa leitura dos artigos e livros foi realizada na área de modelagem 3D a fim de encontrar fundamentos teóricos para esta monografia. Outros instrumento de coleta de dados foram o aprendizado por meio de vídeos práticos, sobre técnicas de modelagem, encontrados em sites como o *YouTube* e *Vimeo*, e conversas em fóruns digitais como o fórum educacional da *AUTODESK*.

3.1.2 Reconstrução do ambiente do ARSTUDIO

Por ser um ambiente que já estava em desenvolvimento antes do início deste trabalho, foi necessário reconstruir o ambiente original. Para isso, todas as bibliotecas utilizadas no projeto tiveram que ser instaladas no sistema.

Esta etapa de reconstrução do ambiente foi usada na aprendizagem do funcionamento do software ARSTUDIO, das bibliotecas utilizadas por ele, pois foi necessário estudar a estrutura do software e a função de cada uma das bibliotecas dentro dessa estrutura, e do hardware, dispositivos como as câmeras utilizadas pelo sistema.

3.1.2.1 Ambiente de Software e Hardware

Durante a instalação e configuração das bibliotecas e drivers dos dispositivos, como existem muitas ações específicas que são necessárias e podem passar despercebidas, como por exemplo adicionar ou comentar certas linhas de código, versões compatíveis de bibliotecas ou a ordem em que certas operações devem ser realizadas, foi gerado um manual de instalação, de 15 páginas, para auxiliar neste processo no futuro. As bibliotecas utilizadas são: ARToolKit, OpenCV, FMOD, libfreenect, Qt, OpenSceneGraph, OSGART, que foram explicadas no segundo capítulo desta monografia. Os dispositivos utilizados com o ARSTUDIO foram a Webcam da Logitech e o Kinect v1 da Microsoft, que foi descrito no segundo capítulo desta monografia.

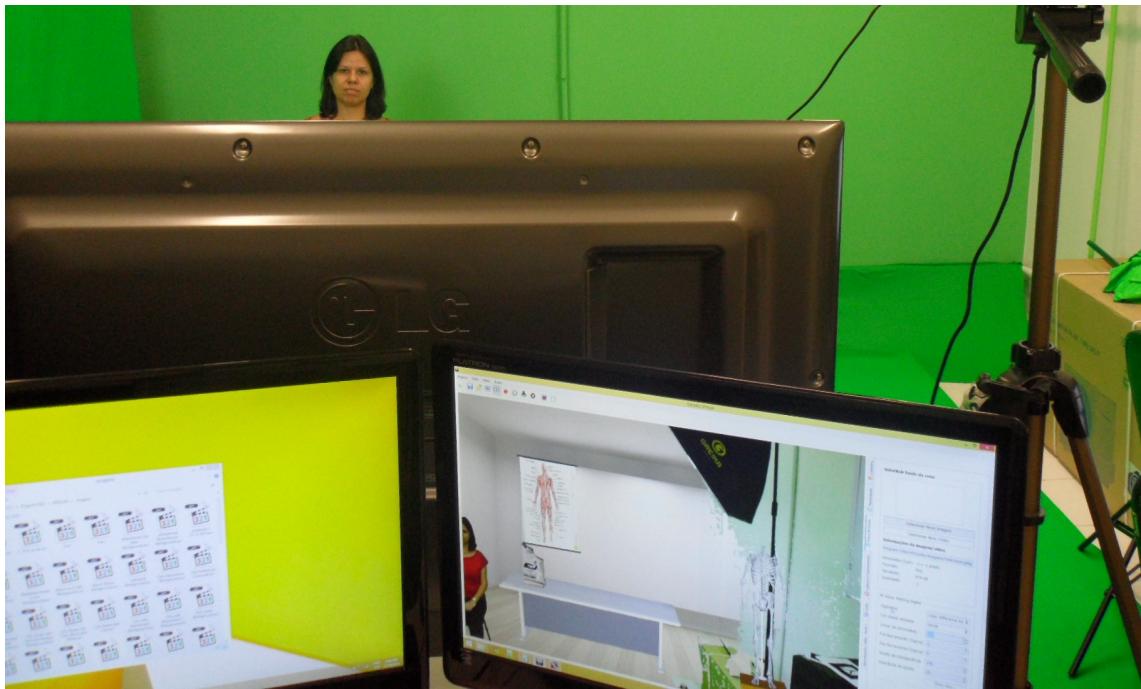
3.1.2.2 Ambiente físico

O ambiente físico seria o espaço físico utilizado para a realização dos testes com o ARSTUDIO. Este espaço é conhecido como o espaço do estúdio real. Este espaço é composto pela:

- a) **Mesa do Operador:** Local onde é realizado as operações no software em tempo real, armazenado os fluxos da câmera e do vídeo composto, e também o local onde é realizado a edição do conteúdo gerado;
- b) **Cenário Real:** Local a ser capturado pela câmera. Este local é onde os elementos de filmagem estão como os objetos e atores reais. Neste local também estão os marcadores fiduciais e as telas usadas para o *matting* digital, neste caso são telas verdes;
- c) **Área da Produção:** Este é o local para o armazenamento e uso do equipamento físico. Alguns destes materiais são as luzes para a iluminação do cenário na hora da produção, tripés para dar suporte as câmeras e luzes, marcadores fiduciais e câmeras não sendo utilizadas, televisão para o retorno ao ator, etc.

Foi necessário a configuração correta deste espaço, principalmente nos aspectos de enquadramento adequado da visão da câmera, pois era necessário capturar somente o espaço coberto por verde, a altura e a distância correta dos atores reais e do objetos virtuais, assim como o chão do ambiente; posicionamento correto das luzes, para que houvesse uma iluminação correta sobre a área verde, para auxiliar no *matting* digital, e nos atores; bem como o posicionamento correto do retorno ao ator e da mesa do operador, para que o ator pudesse ter um retorno visual de suas ações em relação ao mundo virtual de modo que não prejudicasse a sua atuação e para que o operador pudesse ter uma visão geral do ambiente que estava controlando. A Figura 19, a Figura 20 e a Figura 21 retratam este ambiente físico.

Figura 19 - Visão da Mesa do Operador.



Fonte: Elaborada pela autor.

Figura 20 - Cenário Real.



Fonte: BARBOSA, 2015.

Figura 21 - Área de Produção.



Fonte: Elaborada pela autor.

4 DESENVOLVIMENTO DO PROJETO PROPOSTO

Neste capítulo é abordado o desenvolvimento realizado para atingir os objetivos propostos nesta monografia, que é um estudo das funcionalidades e limitações do ARSTUDIO e uma análise da modelagem de objetos virtuais. O capítulo é dividido em duas subseções onde cada uma trata de um dos objetivos propostos nesta monografia.

4.1 ARSTUDIO

Nesta seção trata a geração de um conteúdo piloto e como consequência uma análise das funcionalidades do ARSTUDIO, dificuldades encontradas, um levantamento das limitações presentes e propostas de novas funcionalidades, visando a melhoria e abrangência dos recursos oferecidos pelo sistema, à luz do que é esperado de um estúdio virtual.

4.1.1 Geração de um conteúdo piloto utilizando o ARSTUDIO

Para poder analisar as funcionalidades reais, bem como suas limitações, propôs-se a geração de um conteúdo piloto que faria uso de todos os recursos apresentados pelo ARSTUDIO. Este conteúdo foi criado com auxílio de outros alunos do laboratório.

O conteúdo gerado foi denominado "Aula de Anatomia: Esqueleto Humano", que pode ser encontrado no *YouTube*¹¹. Este conteúdo trata-se de uma aula de anatomia do esqueleto humano ministrada por uma professora. A filmagem desta aula ocorreu em cinco cenas. Aqui está uma explicação de cada cena de forma que foi gravada, tanto pelo ponto de vista telespectador do conteúdo combinado, quanto pela visão da equipe de produção do conteúdo:

A. **Cena 1:** Caminhando até a escola.

- a. Telespectador: A professora caminha pela calçada de uma cidade praiana.
- b. Produção: A atriz caminha do lado esquerdo ao direito da área de captura da câmera (lado direito ao esquerdo do cenário real pela visão da atriz). Toda área de captura está coberta por um tecido ou uma parede verde. É então segmentado o *foreground*, a atriz, do *background*, restante da cena capturada,

¹¹ Disponível em: <<https://www.youtube.com/watch?v=gFDW5b6boM0>>. Acesso em: 10 fev. 2015

utilizando a cor verde com o método *color difference key*. Após a segmentação é realizado a composição por um background de uma praia que contém um espaço de calçada, mais próximo a câmera, onde a atriz estaria caminhando virtualmente. A Figura 22 ilustra esta cena com as duas visões.

Figura 22 - Conteúdo Piloto, Cena 1.



Fonte: Elaborada pela autor.

B. Cena 2: Chegando na escola.

- Telespectador: A professora caminha apressadamente, pois está atrasada, pela calçada, onde encontra-se a escola, e então entra pela porta da frente da mesma.
- Produção: A atriz está próxima a parede, do lado esquerdo do cenário, pela visão da câmera. Ela olha o horário, em seu relógio de pulso, e ao observar o atraso caminha mais rapidamente em direção ao lado direito do cenário, pela visão da câmera, porém em certo momento é necessário virar no sentido anti-horário para que fique de frente para a porta da escola. Novamente a atriz é extraída e o background é então composto por uma imagem de fachada de uma escola, com espaço no inferior da imagem de uma calçada onde a atriz estaria caminhando. A Figura 23 retrata esta cena com as duas visões.

Figura 23 - Conteúdo Piloto, Cena 2.



Fonte: Elaborada pela autor.

C. Cena 3: Susto no corredor.

- a. Telespectador: A professora caminha pelo corredor da escola, leva um susto ao ver um crânio de um esqueleto que está em um armário e continua caminhando.
- b. Produção: A atriz caminha desta vez em outro sentido, em relação a câmera. Ela inicia a trajetória ao lado da câmera e anda na diagonal para a esquerda até um certo momento em que ela se apresenta assustada e depois no continua seu caminho de forma diagonal para a direita. Desta vez o *background* composto é o corredor de uma escola. Neste corredor foi adicionados a cena dois objetos virtuais, um armário que contém um crânio de um esqueleto na prateleira superior e uma cabeça, digitalizada de um manequim real, que foi posicionada na prateleira inferior em relação ao crânio. A Figura 24 apresenta esta cena com as duas visões.

Figura 24 - Conteúdo Piloto, Cena 3.



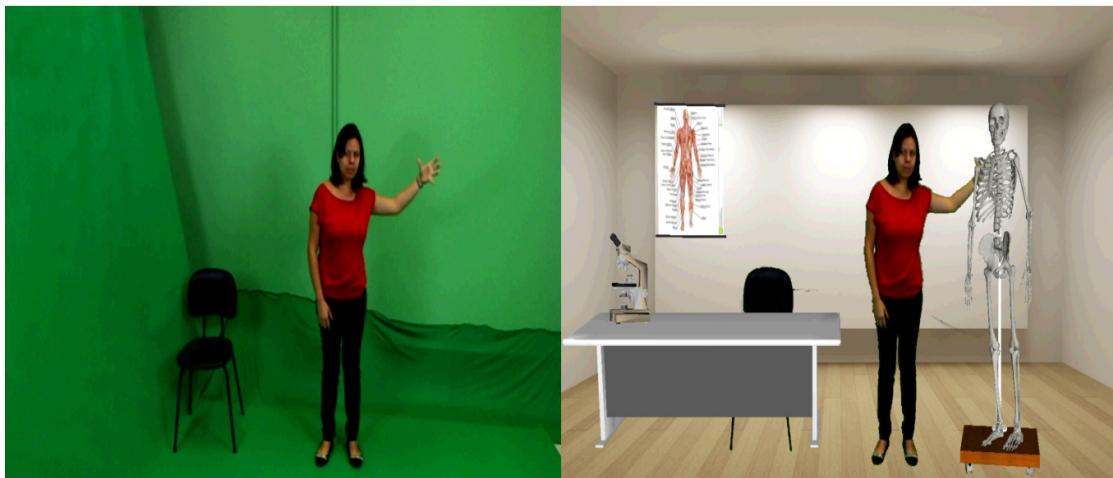
Fonte: Elaborada pela autor.

D. Cena 4: Aula de anatomia.

- Telespectador: Na sala de aula, é iniciado uma aula de anatomia. A professora, que está sentada atrás de sua mesa, se levanta em direção a um esqueleto que é usado para a explicação da aula. Em certo momento o crânio do esqueleto se volta para a professora e ela colocá-lo de volta no lugar. A explicação continua mas é interrompida desta vez pela movimentação do corpo do esqueleto, que a professora também coloca no lugar correto.
- Produção: A atriz está sentada em uma cadeira real, atrás de uma mesa virtual, próxima ao canto esquerdo da área de captura, cujo *background* é trocada por uma sala de aula. Sobre a mesa está um microscópio virtual e atrás da atriz está uma pôster virtual de anatomia humana. Ela levanta da cadeira e anda um pouco para a sua esquerda, lado direito em relação à câmera, em direção ao esqueleto virtual. Enquanto isso ela vai falando, o que foi anteriormente roteirizado, sobre a aula e iniciando a explicação da anatomia humana. Em certo momento ela transmite uma percepção falsa de que o crânio, virtual, do esqueleto está voltada para ela e então caminha mais próximo ao esqueleto para ajustar o posicionamento do crânio de volta a posição inicial. A fala da atriz prossegue até o momento que ela transmite uma percepção falsa de que o corpo do esqueleto está em movimento, ela então posiciona o corpo de volta a

posição correta e continua com a sua fala. A Figura 25 mostra esta cena com as duas visões.

Figura 25 - Conteúdo Piloto, Cena 4.



Fonte: Elaborada pela autor.

E. Cena 5: Continuação da aula de anatomia.

- Telespectador: A professora retorna à explicar sobre a anatomia, porém desta vez é interrompida ao perceber que o crânio está voando pela cena. A professora coloca o crânio de volta a posição correta e volta a explicação. O crânio novamente se move, mas desta vez em direção a professora que retorna a sua cadeira e desiste da aula.
- Produção: Nesta cena os objetos e a imagem do *background* continuam os mesmos em relação a cena 4. A atriz retorna a fala roteirizada que é interrompida ao transmitir uma falsa percepção de que o crânio está voando distante do tronco do esqueleto, no sentido a direita em relação a câmera. Desta vez o crânio virtual está fisicamente associado, em tempo real, a um marcador que está sobre um ator coberto de verde. É rosto do ator é então agarrado pela atriz e movido de volta ao local inicial, dando a impressão de que o crânio foi agarrado e retornado a posição inicial. A atriz prossegue então com a sua fala e ao dar a falsa impressão de percepção de que o crânio continua a voar, ela se move em direção a cadeira. O crânio, portanto o ator,

continua se movendo em direção a atriz se agacha para que crânio virtual fique atrás da mesa virtual. A Figura 26 ilustra esta cena com as duas visões.

Figura 26 - Conteúdo Piloto, Cena 5.



Fonte: Elaborada pela autor.

Para que esse conteúdo fosse gerado, houveram muitas tentativas e erros até que um resultado aceitável pudesse ser obtido. Falarei mais a respeito disso na próxima subseção.

Este mesmo conteúdo deu fruto a outros conteúdos, que também podem ser vistos no YouTube: Programa piloto "Aula de Anatomia: esqueleto humano", Versão II; Bastidores, Programa piloto “Aula de Anatomia” com realidade aumentada com o software ARSTUDIO; e Tutorial ARSTUDIO.

4.1.2. Levantamento das funcionalidades oferecidas

As funcionalidades do ARSTUDIO usadas na geração do conteúdo anteriormente descrito foram:

- **Inserção e retirada de objetos virtuais:** Pelo uso de realidade aumentada, neste caso com marcadores fiduciais ou multi-marcadores, é possível associar objetos 3D a estes marcadores. O formato destes objetos são: obj, definição geométrica no formato texto e desenvolvida pela Wavefront Technologies; 3ds, definição geométrica e de cena no formato binário, utilizada no 3ds Max e desenvolvida pela AUTODESK (MCHENRY, e BAJCSY, 2008); e o osg, definição geométrica e do grafo de cena do

OpenSceneGraph, com a possibilidade de criar uma animação e partículas predefinidas (WANG e QIAN, 2010). É também possível a retirada de objetos da cena;

- **Fixação de objetos:** Fixa o objeto na posição, em relação a câmera, em que o marcador foi encontrado pela última vez antes de ativar a fixação, o qual poderá ser realizado a qualquer momento, fazendo com que o marcador relacionado a este objeto não seja mais procurado na cena e atualizado a cada frame. Isso possibilita, além de uma estabilidade física ao modelo virtual e diminuição do processamento computacional, a retirada do marcador da cena. Podendo assim realizar a criação do cenário virtual em relação ao real e depois a retirada dos marcadores, o que gera a percepção de que os objetos virtuais são reais na cena. Há também a possibilidade de desfixar o objeto a qualquer momento, fazendo com que o processamento da imagem em busca pelo marcador, portanto a atualização da posição do objeto virtual na cena, seja realizado novamente a cada frame;
- **Manipulação de objetos:** Além da manipulação através de movimentação do marcador físico, é possível também manipular a escala do objeto virtual, bem como suas rotações e translações nos eixos X, Y e Z. Essa manipulação pode ser realizada tanto em objetos fixados quanto nos não fixados;
- **Oclusão Mútua:** Com informações de profundidade provenientes da Kinect v1, ou um marcador de plano que serve de plano de corte (SANCHES et al., 2012), é possível realizar a oclusão mútua entre objetos reais e virtuais;
- **Matting Digital:** Pelo uso dos métodos de *Chroma-key* e *Color Difference Key*, é possível realizar a segmentação do *foreground* e do *background*, bem como a extração do *foreground* e uma composição com um *background* novo, sendo que esse *background* pode ser tanto uma imagem, no formato jpg, quanto um vídeo, no formato avi;
- **Marcadores coloridos:** É possível a utilização de marcadores com espectros de cores que possibilitam a extração do mesmo da cena como se fosse a cor segmentada pelo *Matting* digital e ainda assim de modo que o marcador contém uma variação de cor suficiente de modo que é reconhecido pelo módulo de detecção do marcador possibilitando a inserção de objetos virtuais associados ao mesmo;

- **Visualização em tempo real:** É possível observar a cena aumentada, através da combinação da cena real capturada com objetos virtuais e substituições do *background*, em tempo real;
- **Exportação:** É possível exportar o vídeo não processado, proveniente da câmera de captura, o vídeo processado, vídeo composto com a cena aumentada, o áudio gravado da cena por meio de um microfone e um arquivo que armazena as associações dos objetos com os marcadores e as respectivas rotações e translações feitas nos mesmos objetos;
- **Importação:** É possível importar o arquivo das associações dos objetos com os marcadores e também é possível importar um vídeo, no formato avi, que é utilizado como se fosse um fluxo de vídeo proveniente de câmera em tempo real, possibilitando uma espécie de pós-produção.
- **Interface Gráfica:** O ambiente possui uma interface gráfica composta onde é realizado a interação com o sistema,

Um multi-marcador é quando um grupo de marcadores são associados com suas posições relativas. Quando pelo menos um desses marcadores são visíveis, pode ser computado a posição do marcador em relação a câmera, portanto o objeto pode continuar sendo apresentado se um ou mais, não todos, os marcadores forem obstruídos. Isso pode servir tanto para deixar o rastreamento mais preciso, quanto para utilizar marcadores de outros formatos, como um cubo em que cada lado representa uma visão, transformação, referente a aquele objeto.

4.1.2.2 Dificuldades na geração do conteúdo

As dificuldades encontradas durante a geração do conteúdo foram:

- **Matting Digital:** A cor utilizada para a segmentação sempre foi o verde, portanto não utilizávamos roupa ou objetos verdes na cena real para que não houvesse problema. A dificuldade porém foi de cobrir toda a área capturada pela câmera com verde e também com o fato de não criar sombras no cenário. Isso foi amenizado utilizando materiais verdes que eram fixados na cena real em diversos locais e também pelo uso de luzes para iluminar bem o cenário, porém a dificuldade com as sobras dos atores que estavam sempre em movimento. Outra forma de amenizar isso foi alterando o

limiar da cor, porém isso pode causar com que certos objetos reais são considerados como contendo aspectos verdes e causa uma segmentação errônea;

- **Retorno ao Ator:** Em um ambiente de estúdio virtual, com uma instalação de chroma-key, existe a dificuldade de prover aos atores um retorno, ou referência visual, da cena e dos objetos virtuais. É difícil, mesmo para atores treinados, interagir com objetos virtuais que eles não vêem. A posição desconhecida de outra pessoa ou objeto pode levar a pose ou o olhar incorreto do ator. Da mesma forma, a sincronização temporal com movimentos, eventos ou gestos é difícil de conseguir sem qualquer retorno (GRAU et al., 2012). Algumas dificuldade que os atores tinham em relação ao retorno de informação foram a respeito da noção do posicionamento dos objetos virtuais, o tamanho dos mesmos e onde acabava a área de captura pela câmera. Isso foi amenizado utilizando uma televisão que ficava de frente para o ator que retornava um vídeo composto, que era a duplicação da tela do operador, e também através de marcações pequenas no chão e nas paredes de modo que não interferissem com o *Matting* digital;
- **Animação de objetos:** Somente objetos osg podem conter animações predefinidas. Como utilizávamos objetos obj e 3ds, não foi possível a utilização de animações com os mesmos. Para amenizar essa dificuldade foi necessário que um ator que se cobrisse de verde, para que fosse retirado da cena composta, com a utilização do *Matting* digital, e carregasse consigo um marcador com um objeto virtual associado que era movido junto com a movimentação do mesmo ator;
- **Sincronização:** A sincronização do áudio e vídeo gerado, que será mais elaborado na próxima subseção, é um problema pertinente, tanto na filmagem com o Kinect ativado quanto com ele desativado. Essa falta de sincronização se agrava conforme é aumentado o tempo de gravação. Essa dificuldade foi contornada por três métodos, o primeiro foi utilizando gravações de curta durações, buscando ser menor do que um minuto e meio, que depois poderiam ser juntadas; na edição do vídeo, acelerando a taxa de frames, a um valor de 1.09, para que pudesse chegar mais próximo a velocidade da taxa de áudio; e alterando a posição inicial do áudio, para que no começo não houvesse uma sincronização mas apresentaria um pequeno atraso do áudio, para que no meio do vídeo houvesse uma sincronização e depois no final não

houvesse uma sincronização novamente, apresentando um áudio acelerado em relação ao vídeo;

- **Fixação de objetos:** A fixação de objetos, explicado anteriormente, fixa os objetos a localização em relação a câmera. Uma dificuldade que ocorre é quando se associa mais de um objeto ao mesmo marcador. Se há mais de um objeto associado ao mesmo marcador, para que os objetos ficam fixos, é necessário fixar todos os objetos associados aquele marcador;
- **Retirada de objetos:** A retirada de objetos é a retirada dos mesmos da cena e da associação com o marcador. Isso pode falhar, conforme explicada na próxima subseção. A maneira encontrada de contornar isto foi parando a execução do programa e executando novamente, o que causa problema se muitas configurações foram realizadas e portanto precisam ser feitas novamente. Caso a retirada ocorra com sucesso, se o objeto foi retirado sem antes fixar, os demais objetos associados ao mesmo marcador não conseguiram ser fixados;
- **Movimento da câmera:** A movimentação da câmera após a inserção de objetos virtuais não é possível, mais detalhes na próxima subseção. Um método de contornar isto é através de planejamento prévio das movimentações dos atores reais na cena no espaço capturado pela câmera, bem como o posicionamento de objetos virtuais.

4.1.2.3 Levantamento das limitações presentes

Com base na experiência de criação de conteúdo com o ARSTUDIO, podem ser destacadas as seguintes limitações:

- **Sincronização:** Conforme explicado anteriormente, o ARSTUDIO tem a capacidade de exportar os dois tipos de vídeos e o áudio. Isso traz três limitações. O primeiro sendo que os arquivos vêm de forma separados, portanto é necessário de um programa de edição para realizar a sincronização e junção do áudio e vídeo em um único arquivo. O segundo vem do fato de que quando é feito uma requisição de encerrar a gravação e exportar, não há sincronismo na hora de encerrar a captura do áudio e vídeo, o que gera arquivos de áudios com durações maiores do que os vídeos, necessitando cortar parte do áudio fora. O terceiro é a taxa de captura que é distinta para o vídeo e o áudio, o que causa problema na sincronização, causando com que o

áudio fique sempre a frente do vídeo, dando uma impressão de que o áudio está acelerado;

- **Retirada de objetos:** O ARSTUDIO apresenta a opção de retirar objetos que foram inseridos na cena, porém essa retirada não retira o nó do objeto no grafo de cena. Essa retirada é simplesmente visual e pode nem sempre funcionar visualmente, causando com que o objeto que foi requisito sua exclusão seja retirado da lista de objetos, na interface com o usuário, porém o mesmo continua sendo apresentado no fluxo de vídeo composto;
- **Movimentação da câmera:** Não é possível realizar a movimentação da câmera após a inserção de objetos virtuais seguida da remoção dos marcadores da cena, uma vez que o posicionamento dos objetos está atrelado ao posicionamento da câmera, portanto, alterando a posição da câmera alteraria a posição dos objetos virtuais.

4.1.3 Proposta de novas funcionalidades

Em vista dos aspectos levantados nas seções anteriores, propõe-se as seguintes novas funcionalidades ao ARSTUDIO:

- **Armazenamento de informações:** O armazenamento está limitado a aqueles aspectos discutidos anteriormente. Uma informação que está sendo utilizadas porém não armazenada é o mapa de profundidade a cada frame. A combinação deste mapa de profundidade com a posição dos objetos virtuais, em relação a real cena, possibilitaria uma pós produção com a oclusão mútua de objetos. Uma outra informação que poderia ser armazenado é o grafo de cena a cada frame, podendo assim, ser importado a um programa de pós produção com a cena devidamente criada a cada frame e a possibilidade de trocar os objetos virtuais por objetos de mais alta qualidade;
- **Retirada de objetos:** Conforme discutido anteriormente, existe a limitação da retirada do objeto de cena. Essa retirada poderia ser melhor implementada, retirando efetivamente o nó do grafo de cena;
- **Rastreamento de câmera:** O movimento da câmera no estúdio, bem como zoom, rotações no eixo fixo sobre o plano horizontal e inclinação, não são possíveis sem a captura e análise dos parâmetros posicionais da câmera real. Esta limitação é um resultado devido a desvinculação espacial do plano da frente com o plano de fundo

que são imagens sobrepostas no final em apenas um plano pelo compositor de imagens. Se uma alteração mais tarde viria a ser feito ao segmento de imagem, a referência espacial seria perdida e as perspectivas do plano da frente com o de fundo não seria mais corretas. Para superar estas desvantagens é necessário um travamento espacial do plano da frente com o do fundo. Isto é obtido através da captura dos parâmetros de posição da câmara de gravação, incluindo os parâmetros de ajuste da lente. (RATTHALER, 1996). Além deste aspecto em relação as imagens, há também a relação da movimentação da câmera em relação as posições dos objetos virtuais na cena;

- **Captura com diversas câmeras:** Seria a captura da cena em relação a outros pontos de vista e de modo sincronizado entre as câmeras, tendo assim a possibilidade de outras visões sobre a mesma cena com coerência no posicionamento e oclusão mútua dos objetos virtuais, que poderiam ser utilizados tanto em tempo real quanto na pós produção;
- **Retroalimentação:** Segundo Yamanouchi et al. (2002) é difícil gerar objetos de computação gráfica de alta qualidade, em tempo real, e isso faz com que seja difícil integrar o virtual e o real para criar uma ilusão de foto-realismo em tempo real. A maioria dos programas de TV são produzidos filmando cenas em um estúdio com modelos virtuais de baixa resolução. Em seguida, após a captura da cena real, é feito um trabalho demorado de renderizar os objetos virtuais e combinar esses elementos para criar uma imagem de mais alta qualidade. Conforme descrito anteriormente, há a possibilidade de importação de um fluxo de vídeo previamente armazenado. O ARSTUDIO poderia, nesta fase de pós produção, utilizar métodos mais computacionalmente complexos para o cálculo da segmentação, inserção de objetos com devidas profundidades e com uma entrada de qualidade de vídeo superior, levando em consideração as observações feitas nesta subseção em relação a funcionalidade proposta de armazenamento de informação. Tornando assim este sistema mais completo em termos de oferecer recursos para as fases de produção e pós produção;
- **Plano de Corte:** Atualmente o Kinect serve para realizar a oclusão mútua entre objetos reais e virtuais. Com o recurso de gerar um mapa de profundidade, seria possível também realizar um corte no plano utilizando esta profundidade, fazendo

assim uma segmentação do *foreground* e *background* sem a necessidade de utilizar fundos coloridos, fornecendo assim a opção de ter objetos e roupa de diversas cores e de não ter que se preocupar tanto com sombras. Isso possibilitaria que o cenário real atrás desse plano não aparecesse, podendo assim existir um cenário real, objetos e atores reais, marcadores fixos ou em movimento, que não apareceriam em cena. Estes marcadores em movimento possibilitaria a movimentação de objetos virtuais em tempo real sem a necessidade de animações predefinidas ou atores segurando os marcadores e vestidos da cor segmentada pelo *matting* digital. Isso também possibilitaria diversos planos de corte na mesma cena, de tamanhos e ângulos distintos;

- **Retorno ao ator:** Segundo Grau, Pullen e Thomas (2004), uma das formas de fornecer um retorno ao ator é através de projeções da cena ao redor do ator, levando em consideração a posição e orientação da cabeça e portanto da visão do ator sobre aquela cena virtual projetada. Alguns problemas como consequência da utilização desta técnica são o fato da projeção sobre o cenário físico interferir com o recorte do chroma-key e também pela projeção possivelmente incidir sobre os atores e objetos de cena, portanto alterando suas aparências. Este problema foi resolvido com um pano retro-reflexivo e câmeras equipado com um anel de LEDs azuis que são refletidas por um material do pano e capturados pela câmera como um azul saturado necessário para a segmentação com o chroma-key. Ao mesmo tempo, o ator pode observar imagens de um projetor de vídeo os quais são projetados sobre o pano, mas de forma que os níveis de luz não interferem com a tecnologia chroma-key;
- **Interação com objetos:** Atualmente a interação é realizada fisicamente manipulando os marcadores. Isto pode causar problemas como obstruções, sombras ou má iluminação sobre o marcador fazendo com que o objeto desapareça da cena. O Kinect possibilita a captura e reconhecimento de gestos. Essa entrada de dados poderia ser utilizada para interagir com objetos virtuais, não só pela alteração de suas matrizes de transformação mas também provocando animações realizadas por estes objetos.

4.2 Técnicas de geração de objetos tridimensionais

A criação de uma base de objetos tridimensionais é fundamental para a geração de conteúdo em estúdios virtuais. Conforme ilustrado na Figura 2 do capítulo 2, pode ser

observado que a base de dados 3D é utilizada em todas as etapas de produção. Com isso, pode-se constatar a importância do conteúdo 3D e subsequentemente de uma análise para a geração deste mesmo conteúdo.

4.2.1 Técnicas estudadas

Conforme descrito no capítulo 2, existem diversas formas de gerar um objeto 3D, por meio de uso de *softwares* modeladores, utilização de scanners 3D e baseado em imagens. Foi determinado um estudo utilizando estes últimos dois métodos de geração de conteúdo 3D por apresentar formas quantitativas de avaliar os resultados, pois o primeiro método envolve habilidade artística, não podendo assim ser sistematizado por uma metodologia rígida. Para análise e comparação dos métodos de geração foram utilizados 3 técnicas, duas provenientes das técnicas com scanner, câmera RGB-D (*Red, Green, Blue, Depth*) e scanner 3D, e uma a base de imagens, *Structure from Motion*:

- **Câmera RGB-D:** Foi utilizado o Kinect v2, uma câmera RGB-D da empresa *Microsoft* que captura a profundidade utilizando o mecanismo *ToF*, discutido no capítulo 2, junto com o programa *Kinect Fusion*. Este equipamento possui uma câmera RGB com resolução de 1920x1080 *pixels*, mapa de profundidade de 512x424 pixels e *Field of View (FOV)*, campo de visão, de 89x71 graus. (BREUER, BODENSTEINER, ARENS, 2014).
- **Scanner 3D:** Foi usado um *ranging* scanner, *Sense 3D Scanner*, da empresa *Cubify* que digitaliza com luz estruturada, conforme abordado no capítulo 2, junto com o programa *Sense*. Esse scanner possui uma câmera de RGB com resolução de 320x240 pixels, mapa de profundidade de 320x240 pixels e FOV de 45x57.5 graus. (CUBIFY, 2015)
- **Structure from Motion:** Foram utilizados um *tablet* da empresa Samsung, *Galaxy Note 10.1 2014 Edition*, e uma câmera, AVCCAM HD, da empresa *Panasonic*, junto com programas como o 123D *Catch* (123D Catch, 2014) e *ReCap 360* (ReCap 360, 2015) da *AUTODESK* e o *VisualSfM* (WU, 2011) com o CMPMVS (JANCOSEK e PAJDLA, 2012).

Para a análise das funcionalidades e limitações do ARSTUDIO foi proposta a geração de conteúdos 3D que fariam uso destas técnicas de geração de modelos 3D. Como uma

instância maior, após a geração dos modelos, foram realizados testes para verificar a qualidade e realismo, serão explicado depois o que estes parâmetros representam à luz destes experimentos, do modelo gerado utilizando o ARSTUDIO. Neste estúdio virtual foi realizado a associação dos modelos, gerados pelos métodos que serão abordados, a marcadores para que os modelos virtuais pudessem ser utilizados na interação com objetos 3D e na geração de conteúdo. Este conteúdo 3D foi criado com auxílio de outros alunos do laboratório, por causa que em certo métodos era preciso alguém operando um sistema enquanto outro iria realizando o método ou pelo fato do objeto a ser reconstruído em 3D era uma pessoa.

Estes conteúdos gerados, portanto utilizados nos experimentos, variaram desde objetos pequenos (7,5cm x 2,7cm x 3cm), objetos médios (30,5cm x 44cm x 1,75m) e o objetos grandes (uma sala de aula).

4.2.2 Experimentos com Câmeras RGB-D

Esta subseção relata experimentos feitos em relação a técnica de geração de conteúdo 3D utilizando câmeras RGB-D.

A princípio foi realizado um levantamento da teoria por trás da utilização da câmera de luz estruturada e *ToF* focado em seus sensores e especificações, conforme visto no capítulo 2. Após este levantamento, foi necessário um levantamento específico sobre a utilização destas câmeras na geração de conteúdo 3D. Subsequente ao levantamento foram realizados experimentos práticos utilizando o Kinect v2, pela qualidades apresentadas no sensor de profundidade e câmera RGB.

Foi utilizado o sistema de reconstrução *Kinect Fusion* que vem junto com o *SDK* do *Kinect v2*, *Kinect for Windows SDK 2.0*, para realizar os experimentos de digitalização dos objetos. Este sistema, em tempo real, recebe informações de profundidade e cria um modelo 3D geometricamente correto. O sistema também armazena a pose da câmera utilizando o método SLAM (*Simultaneous Localization and Mapping*), em 6 DOF (*degrees of freedom*), graus de liberdade, e combina novos pontos de vistas da cena em uma representação global da superfície do objeto. O processamento realizado para obter a pose da câmera e a reconstrução da superfície é feita através da utilização da GPU. O *Kinect Fusion*, diferente das outras técnicas que serão abordadas, não utiliza a detecção de características na cena, como foi visto em SfM no capítulo 2. O rastreio é feito pelas poses de câmera e também pelos mapas de

profundidade. O sistema também leva em considerações interações dinâmicas que podem ocorrer, como por exemplo a mão do usuário que aparece em cena, e tem a capacidade de desconsiderar essas movimentações baseando-se na diferença da geometria do modelo, comparado com as em capturas anteriores. (IZADI et al., 2011)

SLAM é um método voltado para o rastreio de usuários ou de robôs enquanto é criado um mapa das regiões físicas envolta do mesmo. (IZADI et al., 2011)

Este programa contém as seguintes funcionalidades:

- **Exportar:** O programa tem a opção de exportar para os formatos stl, com geometria mas sem textura; obj, com geometria e sem textura; e ply, com geometria e a textura é opcional;
- **Espelhamento:** O vídeo apresentado é visualizado de forma espelhada;
- **Volume:** É apresentado o volume de captura no vídeo para auxiliar no posicionamento do objeto a ser capturado;
- **Visão do Kinect:** Troca a visão padrão, focada no objeto a ser reconstruído de forma que o objeto parece estar em movimento, por uma visão real proveniente da câmera do Kinect que mostra a trajetória da câmera ao redor do objeto. Neste modo de visualização também é possível movimentar uma câmera virtual ao redor do objeto sendo reconstruído.
- **Limiar de profundidade:** É possível controlar o alcance mínimo e máximo da profundidade, alterando assim a distância mínima e máxima em que se encontra o objeto a ser digitalizado. Sendo o mínimo 0,5m e o máximo 8m.
- **Resolução:** Controle da resolução da textura a ser gerada sobre o objeto.
- **Controle do Volume:** É possível controlar as dimensões do volume do objeto a ser reconstruído.

Está técnica de geração baseado em uma câmera RGB-D foi estudada utilizando dois esquemas:

- **Objeto fixo e câmera em movimento:** Neste esquema o objeto a ser reconstruído digitalmente era fixo sobre uma mesa ou suporte e o Kinect v2 realizava uma trajetória circular ao redor do objeto.

- **Câmera fixa e objeto em movimento:** Esquema em que o Kinect v2 era posicionado de forma fixa sobre uma mesa ou tripé e o objeto era girado.

4.2.2.1 Objeto fixo e câmera em movimento

Primeiramente foram realizados experimentos em que o Kinect v2 está em movimento e o objeto está fixo. Estes experimentos foram realizados movimentando a câmera totalmente ao redor do objeto, 360 graus, e alterando os parâmetros de profundidade e resolução da área de captura. Para estes experimentos foram utilizadas pessoas. A Figura 27 retrata a área de experimentação com o Kinect v2, bem como um experimento em que uma pessoa está sendo capturada, abaixo é com a textura e a direita é somente a geometria superficial. A Figura 19 apresenta resultados de um experimento, sendo a primeira parte com textura, a segunda com uma superfície sólida e a terceira uma aproximação a nuvem de pontos. A Figura 29 ilustra o erro que é gerado com a perda da pose da câmera.

Figura 27 - Área de experimentação com o Kinect v2.



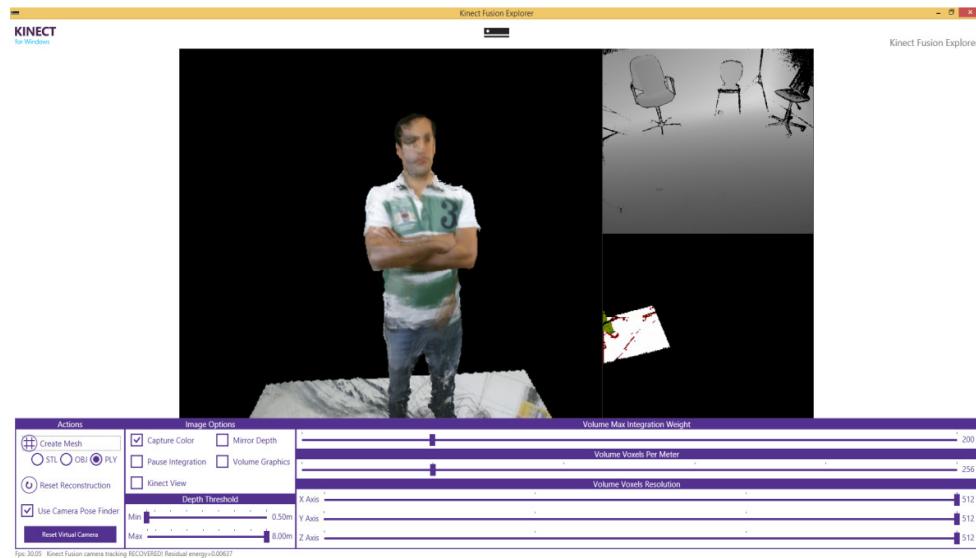
Fonte: elaborada pelo autor.

Figura 28 - Resultado com o Kinect v2.



Fonte: elaborada pelo autor.

Figura 29 - Erro com o Kinect v2.



Fonte: elaborada pelo autor.

A dificuldade encontrada foi a perda do posicionamento da câmera pelo sistema. Foi observado que a pose da câmera é perdida facilmente pelo sistema quando se realiza alguns movimentos, como por exemplo quando troca o peso de uma perna para a outra, o causam tremidas na câmera fazendo ela perder a pose. Essa perda da pose faz com que seja necessário retornar ao último ponto de vista para que possa encontrar onde a câmera está e continuar a reconstrução. Caso não seja possível retornar ao rastreio da posição da câmera, retornando ao último ponto de vista, o objeto precisa ser imediatamente exportado para que possa armazenar a reconstrução feita até o momento, senão a reconstrução será danificada. Caso seja possível retornar ao rastreio da câmera, muitas vezes a continuação da reconstrução pode ser danificada também porque o sistema ficará mais sensível a perda da pose após aquele momento. Para contornar esta dificuldade, foi necessário planejar o movimento ao redor do objeto, respeitando a distância mínima de 0,4m, retirando obstáculos presentes nesta trajetória, incluindo fios. Outro método utilizado para contornar foi ao invés de levantar as pernas para se movimentar, deslizar os pés, dessa forma tremia menos a câmera. Foi testado também a utilização de um tripé para sustentar a câmera mas como era necessário que o movimento da mesma ao redor do objeto, o problema continuava.

4.2.2.2 Câmera fixa e objeto em movimento

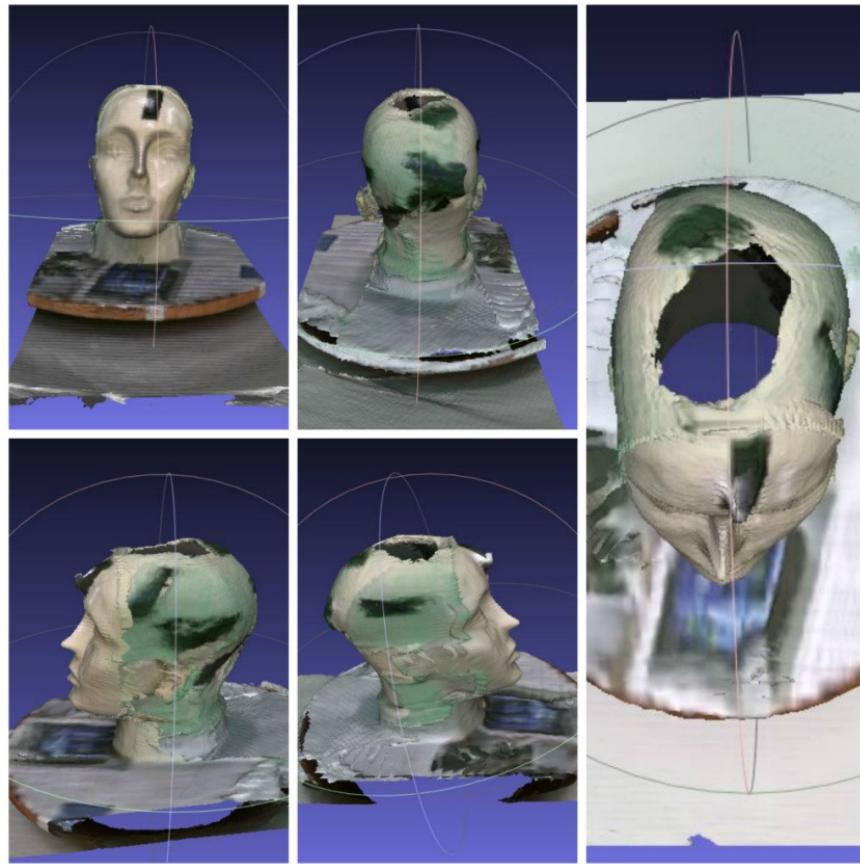
Também foram realizados experimentos em que o Kinect v2 está fixo e o objeto está girando em torno de um eixo. Para realizar a rotação do objeto, foi utilizado uma plataforma giratória que contém marcações especificando os graus. Estes experimentos foram realizados movimentando o objeto, no total de 360 graus, alterando a velocidade da rotação, tempo de espera e alterando os parâmetros de profundidade e resolução da área de captura. Os objetos que foram utilizados para estes experimentos foram uma cabeça de manequim e um dragão de brinquedo. A Figura 30 ilustra a área de experimentação em que o dragão está posicionado sobre a mesa giratória e o Kinect v2 está a sua frente. A Figura 31 retrata o resultado da cabeça.

Figura 30 - Experimentação com o Kinect v2 e uma plataforma giratória.



Fonte: elaborada pelo autor.

Figura 31 - Resultados do Kinect v2 com uma plataforma giratória.



Fonte: elaborada pelo autor.

Foram encontradas duas dificuldades principais: pontos característicos fixos na cena e parâmetros relacionados a velocidade de rotação.

A primeira dificuldade encontrada foi de retirar objetos e outras características que estão fixas enquanto o objeto a ser reconstruído está em movimento. O ato de girar o objeto e deixar a câmera fixa, como o sistema é baseado no movimento da câmera, precisa fazer com que o sistema obtenha a falsa impressão de que a câmera está em movimento ao redor da cena. Se a câmera observa somente o que está girando, o sistema entende que está girando ao redor do objeto, mas caso haja algum objeto ou característica dentro da visão da câmera que se mantém fixo, o sistema irá entender que a câmera está fixa e irá reconstruir o objeto a partir do mesmo ponto de vista, sobrescrevendo superfícies distintas no mesmo ponto. Para contornar esta dificuldade foi primeiro testado a aproximação do objeto a fundos de cor sólida

e posicionamento da mesa giratória sobre uma mesa branca. Quando se aproximava do fundo a cor verde do mesmo refletia-se no objeto causando problema na textura gerada sobre a superfície ao exportar o objeto ou na perda da pose. Foi então aproximado o objeto a fundos brancos, diminuído o alcance da profundidade e também diminuído o volume ao redor do objeto.

A segunda dificuldade encontrada foi em relação a velocidade em que a plataforma giratória era girada. Foi observado que a velocidade em que é girado o objeto, o ângulo girado e o tempo de pausa entre as rotações, influencia na qualidade do modelo gerado. Para contornar isto foram realizados 15 experimentos em que alterava estes parâmetros. Foi usado como base inicial rotações de 45 graus, este número é proveniente de um trabalho sobre reconstruções utilizando o Kinect v1 em que o usuário é reconstruído digitalmente sem a necessidade de outra pessoa para manipular o Kinect v1 (LI e WANG, 2014). O resultado destes experimentos foi que o ângulo ideal é a rotação de 45 graus, realizado 8 vezes para obter uma rotação completa de 360 graus. Esta rotação de 45 graus é feita em 4,5 segundos, portanto 10 graus a cada segundo, e o tempo de espera antes de girar novamente é de 4,5 segundos.

4.2.2.3 Análise dos resultados a partir da câmera RGB-D

Os resultados obtidos, uma vez que tanto a plataforma giratória quanto o Kinect foram manipulados de forma manual (sem qualquer mecanismo de automatização do movimento de rotação), não foram satisfatórios. Em ambos os casos foi observado que após um certo ponto na trajetória, seja girando em torno do objeto ou o objeto em rotação, uma área já gerada do objeto começa a ser sobreescrita por pontos novos causando resultados insatisfatórios. Também foi observado que a quantidade de voltas ao redor do objeto, ou de rotações completas do objeto sobre um eixo, não altera a qualidade do objeto gerado.

4.2.3 Scanner 3D

Esta subseção descreve os experimentos feitos em relação a técnica de geração de conteúdo 3D utilizando um Scanner 3D.

Inicialmente foi realizado um levantamento teórico a respeito dos diversos tipos de scanners disponíveis, o mecanismos de funcionamento dos mesmos e a utilização destes

scanners na digitalização de conteúdo 3D. Após esse levantamento foram realizados diversos experimentos com duas abordagens: a primeira com ênfase na reconstrução de modelos sem cor e a segunda, com cor. Em ambas as abordagens utilizou-se o scanner *Sense* 3D junto com o programa *Sense*.

Este programa possui as seguintes funcionalidades:

- **Tipo de objeto:** O programa apresenta opções para a escolha de tamanhos predefinidos do elemento a ser digitalizado, bem como se o elemento a ser digitalizado é uma pessoa ou um objeto.
- **Exportar:** Pode ser exportado o modelo gerado em três formatos, stl, ply e obj, sendo estes últimos com a possibilidade de exportar com cor.
- **Tratamento:** O sistema oferece a opção de tratar o objeto digitalizado antes de exportar. É possível realizar a exclusão de volumes geométricos; solidificar, tampa buracos encontrados no modelo; e a suavização de arestas;
- **Rastreamento:** Caso o rastreamento do objeto é perdido, há a opção de voltar ao último ponto de vista do scanner com o objeto, ou a opção de retornar ao ponto de vista no qual foi iniciada a digitalização.

4.2.3.1 Experimentos utilizando o Scanner 3D

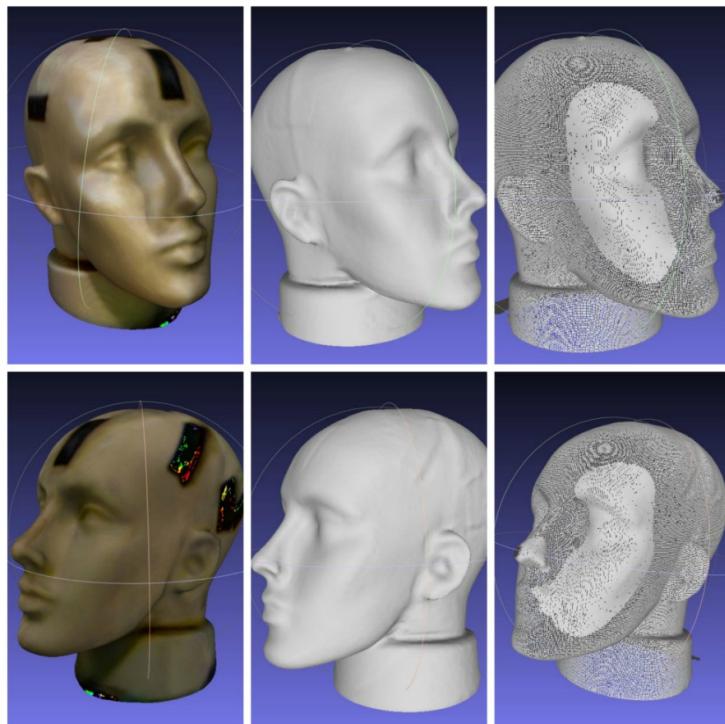
Alguns dos objetos utilizados nestes experimentos foram uma cabeça de manequim e uma garrafa metálica. A Figura 32 retrata a área de experimentação. A Figura 33 apresenta um a cabeça com cor, sem cor e a nuvem de pontos.

Figura 32 - Experimentação com o Scanner 3D.



Fonte: elaborada pelo autor.

Figura 33 - Reconstruções obtidas com o uso do Scanner 3D.



Fonte: elaborada pelo autor.

Observa-se a partir das imagens que os resultados foram satisfatórios no aspecto de geração de uma estrutura tridimensional coerente com a forma geométrica do objeto real, apresentando uma densa nuvem de pontos. Em relação aos resultados dos modelos com cor, não foi possível obter o mesmo nível de satisfação no aspecto de coerência estética em comparação ao objeto que foi digitalizado, por causa da baixa resolução proveniente de câmera RGB.

Este método apresenta duas dificuldades, o primeira é o alcance do cabo USB e o limiar de profundidade e a segunda, os requisitos de processamento para utilização do sistema.

A primeira dificuldade é o curto comprimento do cabo do scanner e pelo limiar de profundidade. O cabo tem 2,13m de comprimento e apresenta um limiar de 0,35m a 3m. Essas condições fazem com que seja inviável a digitalização de objetos médios, como pessoas, por não poder-se movimentar o scanner em torno dos objetos. Para contornar isto foi testado o método, como no caso da câmera RGB-D, de posicionar o objeto a ser reconstruído sobre uma plataforma giratória, mas pelo fato de características fixas estarem presentes na área de captura, o problema não foi resolvido, gerando resultados indesejáveis de sobrescrita no modelo.

A segunda dificuldade é a elevada demanda de poder computacional, tanto do CPU quanto da GPU, necessário para a geração de objetos 3D com este *software*. O computador necessita de, no mínimo, um processador de 2GHz, 2GB de RAM e resolução de vídeo de 1280x1024 pixels. Essa demanda inviabilizou experimentos com o scanner em um computador portátil, que seria o modo de contornar o primeiro problema.

4.2.3.2 Análise dos resultados a partir do uso do scanner

Devido ao fato do scanner utilizado ser um produto comercial voltado justamente para a função de digitalização de objetos 3D, foi esperando, antes mesmo do início dos experimentos, bons resultados. Assim como esperado, houveram resultados satisfatórios na reconstrução geométrica sem cor, porém não foram obtidos resultados semelhantes na reconstrução com cor. Este é um bom método de geração de conteúdo 3D onde se busca uma estrutura geométrica coerente com a estrutura real mas não busca fotorrealismo.

4.2.4 Structure from Motion

Esta subseção trata de experimentos, que buscam a geração de conteúdo 3D, efetuados pela utilização da técnica de *photogrammetry*, *Structure from Motion*, utilizando imagens. Para a captura das imagens foram utilizadas: um tablet Galaxy Note 10.1 2014 Edition, que gera imagens em arquivos *jpg* de resolução 3264x2448 *pixels*, e uma câmera AVCCAM HD que gera imagens em arquivos *jpg* de resolução 6016x3384 *pixels*.

Este estudo teve início pela realização de um levantamento teórico acerca das diversas técnicas de geração de conteúdo 3D baseado em imagens, com foco no *photogrammetry* e especificamente no *Structure from Motion*, as quais foram descritas em detalhe no capítulo 2. Posterior a esta busca na literatura, foi efetuado um investigaçāo das formas práticas de realização desta técnica e por fim experimentos utilizando os dois equipamentos anteriormente citados, três programas para a reconstrução e dois esquemas de captura. Os programas utilizados foram:

- **123D Catch:** Programa de geração de conteúdo 3D a base de fotos de livre acesso disponibilizado pela AUTODESK para o computador, *tablet* e *smartphone*, voltado para usuário comum. Neste programa pode-se enviar, no máximo, 70 imagens que serão processadas na nuvem da empresa. Neste sistema há a possibilidade de realizar algumas edições no modelo, como a exclusão de volumes geométricos. A escolha deste programa foi feita devida ao resultado final da reconstrução estar em forma de uma malha de triângulos, com texturas das imagens provenientes do *tablet* e da câmera, compatível com o ARSTUDIO;
- **VisualSfM com CMPMVS:** O VisualSfM é um programa criado por Changchang Wu, da Universidade de Washington, em Seattle (WU, 2013). Este programa utiliza o método SIFT, descrito no capítulo 2. O resultado é uma nuvem de pontos, uma lista de perspectivas de imagens e informações sobre os parâmetros intrínsecos e extrínsecos da câmera. O CMPMVS foi desenvolvido pelo Michal Jancosek, da Czech Technical University in Prague (JANCOSEK e PAJDLA, 2011). Este programa tem como parâmetros de entrada as informações geradas pelo VisualSfM. O resultado deste programa é uma malha de triângulos com textura, também compatível com o estúdio virtual. A escolha deste programa se deve ao fato de serem *software* livre e por executar localmente;

- **ReCap 360:** Este software é disponibilizado pela AUTODESK para o uso nos navegadores *web*. Trata-se de um programa, voltado para profissionais, que não gratuito mas foi obtido acesso pelo modo acadêmico da AUTODESK disponível para alunos e professores. Apresenta-se, até o momento desta escrita, um limite de 250 imagens, e sem limite na resolução das mesmas, que serão processadas na nuvem da empresa. A escolha deste programa foi feita devida ao resultado final da reconstrução estar em forma de uma malha de triângulos, com altas resoluções, 8192x8192 *pixels*, de texturas das imagens provenientes do *tablet* e da câmera, compatível com o ARSTUDIO.

Os esquemas de captura estudados foram:

- **Ambiente interno com características ao redor do objeto:** Experimento em ambiente com pontos característicos posicionados ao redor do objeto;
- **Ambiente interno com características no objeto:** Experimento em ambiente com pontos característicos posicionados no objeto.

4.2.4.1 Experimentos iniciais

Aqui está apresentado observações iniciais referentes ao três programas que serão importantes nas seções seguintes. A princípio os experimentos eram para conter análises envolvendo os três programas anteriormente mencionados. Após experimentos iniciais com os mesmos, foram observados os seguintes resultados do 123D Catch com o ReCap 360 e do VisualSfM e o CMPMVS com o Recap 360:

123D Catch: Por o 123D Catch apresentar um limite na resolução da imagem aceitável e um limite na quantidade de imagens, os resultados foram inferiores ao resultado quando comparado com o ReCap 360. O experimento foi feito utilizando 48 imagens de resolução 1536x2048 *pixels*. O resultado está demonstrado no

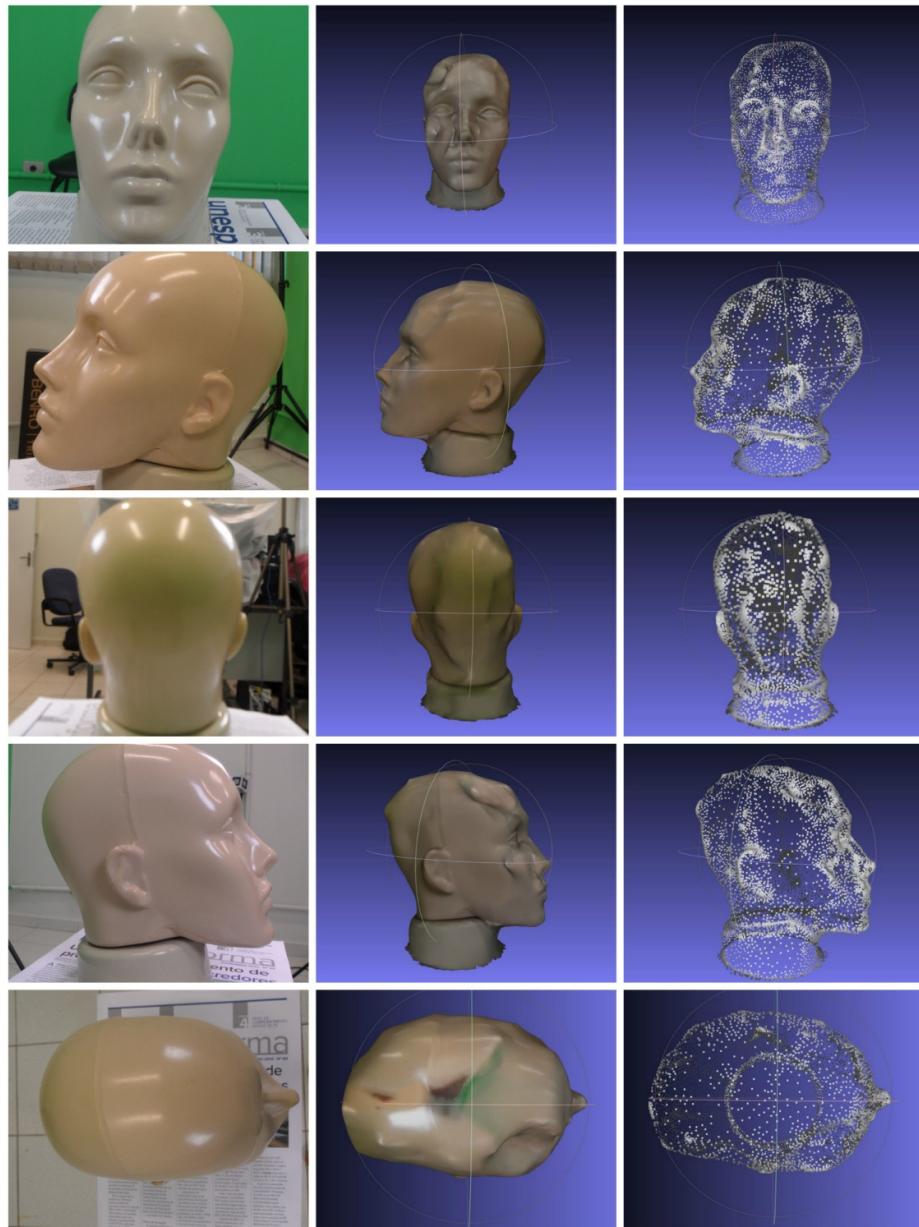
- Quadro 1. A Figura 34 ilustra: as imagens da cabeça no mundo real, modelo gerado com textura e a nuvem de pontos e nuvem, gerado a partir do 123D Catch. A Figura 35 apresenta: as imagens da cabeça no mundo real, modelo gerado com textura e a nuvem de pontos, gerado a partir do ReCap 360.

Quadro 1 - Resultados comparativos entre o 123D Catch e o ReCap 360.

Programa	Qtd. Imagens Entrada	Resolução Entrada (pixels)	Resolução Retornada (pixels)	Vértices
123D Catch	48	1536x2048	4096x4096	9257
ReCap 360	48	1536x2048	8192x8192	79810

Fonte: elaborada pelo autor.

Figura 34 - Resultados da cabeça de manequim com o 123D Catch.



Fonte: elaborada pelo autor.

Figura 35 - Resultados da cabeça de manequim com o ReCap 360.



Fonte: elaborada pelo autor.

- **VisualSfM com CMPMVS:** O software conseguiu combinar somente uma parcela das imagens na etapa de combinação de pontos característicos entre as imagens. Isso fez com que a reconstrução, nesse caso de um dragão, somente gerasse pedaços do

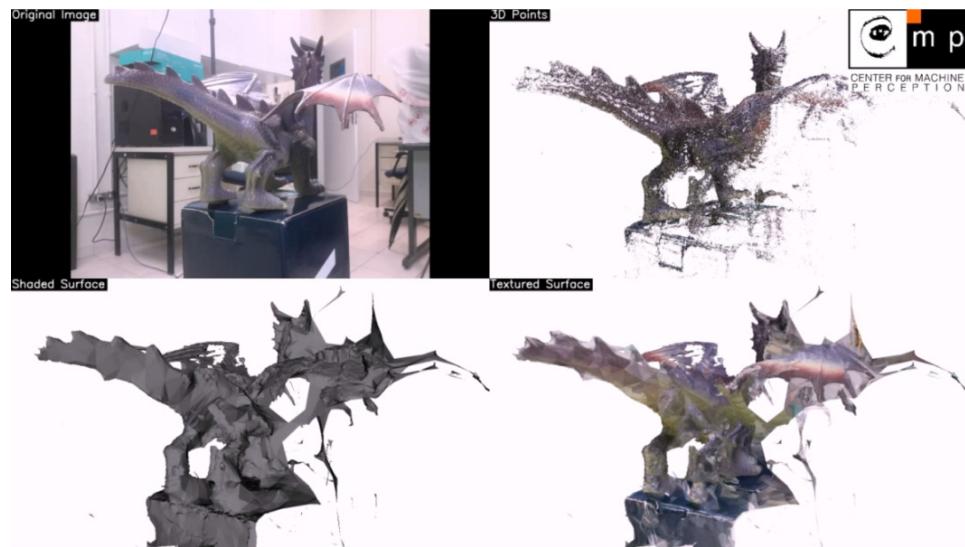
mesmo. O experimento foi feito utilizando 152 imagens de resolução 3264x2448 pixels. O resultado deste experimento está demonstrado no Quadro 2. A Figura 36 apresenta: a imagem do dragão real, superfície reconstruída sem textura, nuvem de pontos e a superfície reconstruída com textura do modelo gerado com o VisualSfM com o CMPMVS. A Figura 37 apresenta: as imagens do dragão real, modelo gerado com textura, nuvem de pontos e nuvem de pontos de uma visão mais próxima, gerado a partir do ReCap 360.

Quadro 2 - Resultados comparativos entre o VisualSfM com o CMPMVS e o ReCap 360

Programa	Qty. Imagens Entrada	Resolução Entrada (pixels)	Resolução Retornada (pixels)	Vértices	Faces	Topologia Retornada
VisualSfM + CMPMVS	152	3264x2448	4096x4096	813678	1626729	Não Coerente
ReCap 360	152	3264x2448	8192x8192	567967	1135642	Coerente

Fonte: elaborada pelo autor.

Figura 36 - Resultados do dragão pelo VisualSfM com o CMPMVS.



Fonte: elaborada pelo autor.

Figura 37 - Resultados do dragão com o ReCap 360.



Fonte: elaborada pelo autor.

O Quadro 3 apresenta as configurações destes programas.

Quadro 3 - Configurações dos programas de SfM.

Programa	Qtd. de Imagens de Entrada	Resolução Retornada (pixels)	Tempo Processamento	Topologia Retornada
123D Catch	70	4096x4096	30min	Não Coerente
ReCap 360	250	8192x8192	40min	Coerente
VisualSfM + CMPMVS	Suportada pela Memória RAM	4096x4096	2h 57min	Não Coerente

Fonte: elaborada pelo autor.

Com base nesses resultados foi desconsiderado o uso do 123D Catch e do VisualSfM com CMPMVS nos demais experimentos e utilizado somente o ReCap 360.

4.2.4.2 Ambiente interno com características ao redor do objeto

Primeiramente foram realizados experimentos em um ambiente interno com características ao redor do objeto. Estes experimentos foram realizados dentro do laboratório, anteriormente mencionado, ou em uma sala de aula da UNESP/Bauru. Foram escolhidos estes locais pelo controle de parâmetros como a iluminação, movimentações ao redor do objeto e na uniformidade de condições climáticas e pontos característicos no ambiente entre os experimentos.

Para aumentar a quantidade de pontos característicos, elementos foram colocados ao redor do alvo da reconstrução, que neste caso foi uma pessoa. A Figura 38 apresenta a área de experimentação com a digitalização de uma pessoa. A Figura 39 apresenta o modelo, com textura, gerado da pessoa deitada.

Figura 38 - Área de experimentação do SfM.



Fonte: elaborada pelo autor.

Figura 39 - Resultados de uma pessoa com o ReCap 360.



Fonte elaborada pelo autor.

A dificuldade encontrada foi a falta de pontos característicos ao redor do objeto, em especial quando aproximava-se o scanner do mesmo. Para contornar isso foi necessário planejar a trajetória que seria utilizada para digitalizar o objeto e, levando-se em consideração a presença de pontos característicos no enquadramento realizado. Conforme a Figura 29, elementos em volta da pessoa com o objetivo de aumentar o número de pontos característicos, como por exemplo, um jornal. Essa técnica apresentou bons resultados.

4.2.4.3 Ambiente interno com características no objeto

Uma outra abordagem foi a realização de experimentos em um ambiente interno com características posicionadas no objeto. Estes experimentos, realizados nos mesmos locais e pelas mesmas razões do esquema anterior, foram executados quando o objeto real apresenta simetria em sua forma geométrica, poucos pontos de reflexão ou uma ausência de pontos característicos. Estas condições podem provocar incertezas ou informações errôneas a respeito de características detectadas, pose da câmera ou textura do objeto, as quais prejudicam as etapas de detecção de pontos característicos das imagens, reconstrução projetiva e

reconstrução 3D de um objeto, conforme descrito no capítulo 2. Os objetos que foram utilizados para estes experimentos foram uma cabeça de manequim, um telefone e uma sala de aula. A Figura 40 apresenta o ambiente de experimento em uma sala de aula. A Figura 41 apresenta a cabeça do manequim com características. A Figura 42 apresenta: as imagens do telefone real, modelo gerado com textura, nuvem de pontos e nuvem de pontos de uma visão mais próxima. Neste telefone, pode-se observar que a frente tem muitos pontos característicos mas a parte de trás não tem, por isso a diferença de qualidade gerada desses dois lados.

Figura 40 - Ambiente de experimentação em uma sala da aula.



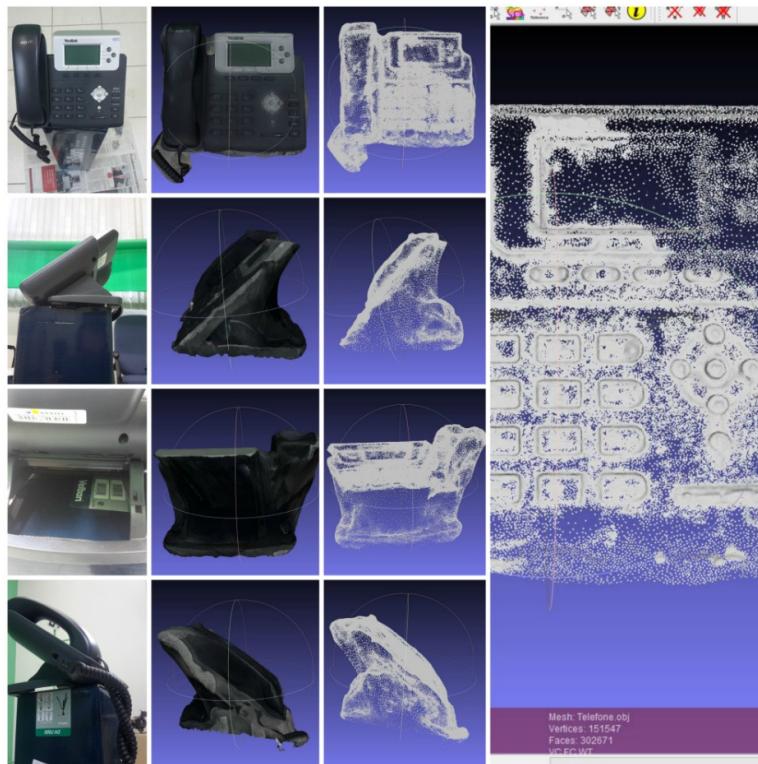
Fonte: elaborada pelo autor.

Figura 41 - Cabeça do manequim com características.



Fonte: elaborada pelo autor.

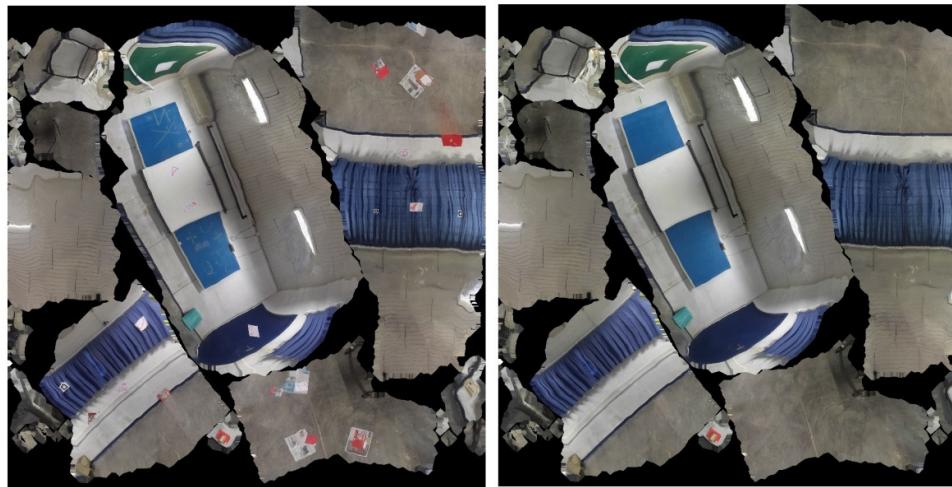
Figura 42 - Resultados do telefone com o ReCap 360.



Fonte: elaborada pelo autor.

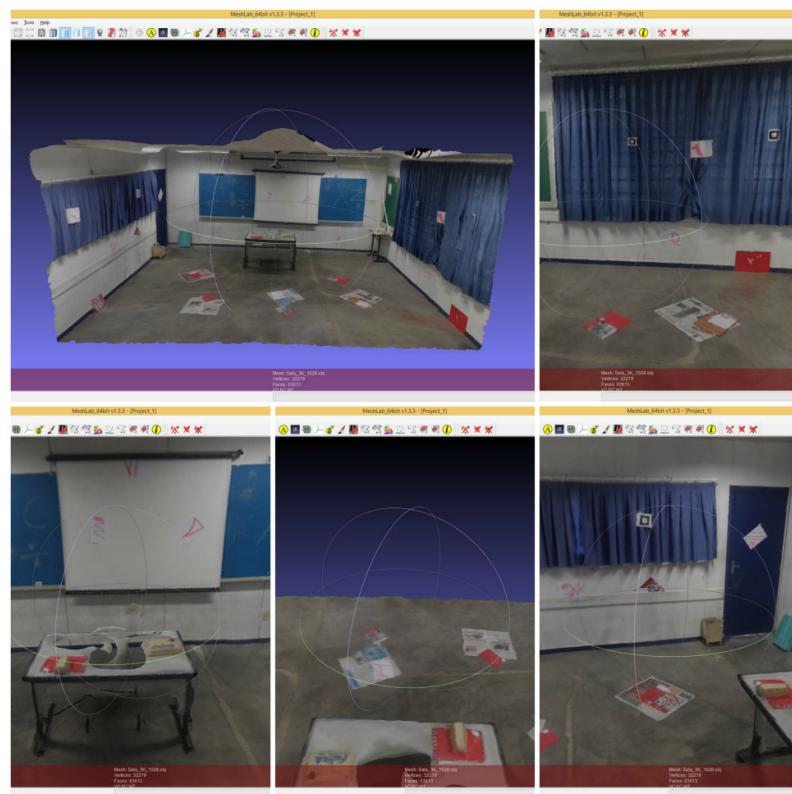
A dificuldade encontrada foi a retirada dos pontos característicos do arquivo jpg da textura do objeto após a geração do modelo 3D. Para realizar esta retirada é necessário que haja um tratamento na imagem, em um editor de imagem, e retire estes pontos. A técnica utilizada para esta retirada foi de copiar os pixels ao redor destes pontos sobre estes pontos a serem retirados. A Figura 43 apresenta a textura da sala antes e depois do tratamento em um editor de imagem. A Figura 44 apresenta o modelo da sala gerado, com a textura, antes deste tratamento e a Figura 45 apresenta o modelo da sala gerado após o tratamento.

Figura 43 - Textura da sala antes e após o tratamento.



Fonte: elaborada pelo autor.

Figura 44 - Modelo da sala gerada antes do tratamento de imagem.



Fonte: elaborada pelo autor.

Figura 45 - Modelo da sala gerada após o tratamento de imagem.



Fonte: elaborada pelo autor.

4.2.3.4 Análise dos resultados a partir do Structure from Motion

Como esta técnica utiliza a detecção de características nas imagens submetidas para processamento, o ato de inserir características no objeto ou na cena, para aqueles que já não contém características próprias, a ser digitalizado faz com que a quantidade de pontos característicos aumente e a combinação destes pontos com os pontos de outras imagens seja facilitado. Obteve-se bons resultados a partir destes experimentos com objetos de diversos tamanhos. Os melhores resultados foram aqueles em que são colocadas características no objeto, gerando assim mais pontos para referência e também sendo possível focar em detalhes do objeto, diferente de utilizar pontos ao redor do objeto em que é necessário capturar imagens com certa distância do objeto de modo que estas características ao redor possam ser capturadas.

4.2.4 Considerações Finais sobre os experimentos

A partir dos resultados destes experimentos, tem-se que a técnica de geração de conteúdo utilizando a câmera RGB-D não apresentou bons resultados visuais e mostrou-se uma técnica de difícil utilização por causa da constante perda da pose da câmera. A técnica a partir da utilização de um scanner 3D apresentou bons resultados com referência a geometria do objeto reconstruído, porém não apresentou bons resultados com relação a textura do objeto, pois a resolução RGB da câmera é baixa. Os resultados a partir da técnica *Structure from Motion* foram inferiores aqueles com o scanner em relação a geometria gerada, porém o aspecto visual é coerente com os objetos reais. Além desta coerência, é também possível capturar áreas maiores do que aqueles capturados com o scanner Sense 3D, como a captura de pessoas e de sala de aulas. A partir dos experimentos realizados, sugere-se o seguinte procedimento:

- Etapa 1 - Determinação do tipo de resultado desejado

Primeiramente é preciso fazer as seguintes perguntas:

- Qual é resultado desejado? Geometria ou fotorrealismo?
- Qual é o tamanho do objeto? Pequeno, médio ou grande?

Caso deseje-se obter fotorrealismo, é possível gerar objetos dos tamanhos pequenos, médios e grandes com a utilização da técnica *Structure from Motion*. No entanto, se o que deseja é obter um objeto com uma geometria de qualidade, é possível gerar objetos de tamanhos pequenos e médios utilizando a técnica com o scanner.

- Etapa 2 - Preparação da área de captura e parâmetros da técnica utilizada

Sabendo qual a técnica que será utilizada, é necessário preparar a área de captura, os seguintes itens precisam ser preparados:

- **Iluminação:** é necessário uma iluminação uniforme de modo que o objeto seja bem iluminado, que no decorrer do processo de captura o usuário não crie sombra sobre o objeto e que a luz não incida sobre a câmera de modo que crie capturas com alta luminosidade, perdendo assim características do objeto e da cena;
- **Características na cena:** é preciso planejar antes a área de captura de modo que seja possível capturar pontos característicos ao redor do objeto e sobre o objeto. Isso

ajudaria tanto na técnica por foto quanto na técnica utilizando o scanner. No caso do SfM, precisa ter certeza de que os ângulos irão capturar pontos característicos, especialmente quando tomado fotos debaixo do objeto, que pode capturar o teto da sala por exemplo, neste caso colocar o ponto sobre o objeto contornaria este problema;

- **Área de captura:** Antes de iniciar a captura, necessita-se o estudo do trajeto a ser percorrido e dos ângulos a serem utilizados para a captura. Precisa ter uma área vasta o suficiente para que seja possível a captura de pontos próximos e distantes, e nesta área não pode haver objetos físicos que podem barrar a captura por aquele ponto de vista naquele local ou ângulo;
- **Quantidade de fotos:** no caso do SfM, a quantidade de imagens capturadas influencia no resultado, pois com mais imagens, é possível capturar melhor o objeto a ser reconstruído, bem como pontos característicos nele e ao seu redor. Porém é necessário que haja uma distância suficiente entre as imagens para que existem tanto pontos característicos iguais como pontos distintos, e de modo que estes distintos serão iguais a outros pontos de outras imagens;
- **Resolução das fotos:** no caso do SfM, a resolução das imagens capturadas irá influenciar no resultado tanto na geometria reconstruída, quanto no fotorrealismo. Pois com uma resolução maior, a textura a ser gerada será mais realista, mesmo aproximando do objeto, e esta resolução maior auxiliará no reconhecimento de características pela cena, em especial nos pontos pequenos na imagem;

5 CONCLUSÃO

Durante o levantamento para esta monografia, pode-se constatar que o estúdio virtual é um recurso relativamente novo, tendo o seu começo no final da década de 80 (SHIMODA, 1989). O avanço da tecnologia vem gerando mudanças nesse ambiente a cada ano, com a inserção de novos recursos computacionais e dispositivos de entrada sendo utilizadas de formas inusitadas. Junto com esse avanço também está o crescimento da quantidade de conteúdo visual gerado e os meios de difundir estes conteúdos, seja pela televisão, cinema ou principalmente a internet. Estúdios virtuais podem servir como o sistema utilizado na criação dos conteúdos para estes meios, mas para que isso ocorra é preciso estudar as qualidades e as limitações que a utilização destes estúdios podem propor. Para que se possa utilizar este estúdios virtuais, é necessário que existem conteúdos para eles, como os conteúdos tridimensionais, portanto é necessário uma análise da modelagem de destes conteúdos.

Neste sentido, o primeiro objetivo desta monografia foi propor um estudo do estúdio virtual, ARSTUDIO que levasse em consideração os conceitos de estúdios virtuais e também a praticidade na sua utilização. Este estudo proporcionou o levantamento de suas funcionalidade e limitações. Verificou-se que é possível gerar um conteúdo piloto com o ambiente atual, porém existem aspectos que podem ser melhorados, tais como a sincronização entre áudio e vídeo e o salvamento das informações de rastreamento 3D para posterior utilização na fase de Pós-produção.

O segundo objetivo desta monografia foi de realizar uma análise quanto aos métodos de modelagem 3D disponíveis atualmente. Foram experimentadas diversas técnicas e situações de geração de conteúdo, utilizando não só diversos programas mas também diversos dispositivos. Verificou-se que o método de *Structure from Motion* gerou os melhores resultados, de acordo com os experimentos realizados. No entanto, dada a manipulação não automatizada dos equipamentos de digitalização (Scanner 3D e Kinect v1 e v2), tem-se que os resultados dos experimentos realizados com estes não podem ser considerados conclusivos.

Com relação aos trabalhos futuros, pretende-se aperfeiçoar o ARSTUDIO. Este aperfeiçoamento será a partir de uma análise da possibilidade de geração de um estúdio virtual utilizando bibliotecas para geração de objetos virtuais mais versáteis, como por exemplo, o UNITY 3D, um software voltado para geração de aplicações tridimensionais e que

contém diversas bibliotecas como as de física, partículas, realidade virtual, realidade aumentada, entre outras. Este ambiente pode ser utilizado não só em computadores, mas também em celulares e *tablets*, possibilitando novas formas de interação com o estúdio virtual e novos dispositivos de entrada. Um outro trabalho futuro seria a automatização, por exemplo, utilizando-se motores, com relação à plataforma giratória usada nos experimentos, bem como na movimentação do Kinect em torno do objeto a ser reconstruído.

REFERÊNCIAS

- AMENTA, N.; BERN, M.; KAMVYSELIS, M. A new Voronoi-based surface reconstruction algorithm. In: **Proceedings of the 25th annual conference on Computer graphics and interactive techniques**, 1998.
- AMENTA, N.; CHOI, S.; KOLLURI, R. The power crust, unions of balls, and the medial axis transform. In: **Journal Computational Geometry Theory Applications**, v. 19, n. 2-3, p. 127-153, 2001.
- AUTODESK. **123D Catch**. Disponível em: <<http://www.123dapp.com/catch>>. Acesso em: 20 dezembro 2014.
- AUTODESK. **ReCap 360**. Disponível em: <<http://www.AUTODESK.com/products/recap/overview>>. Acesso em: 5 março 2015.
- AZUMA, R. T. **A Survey of Augmented Reality**. Presense: Teleoperators and Virtual Environments, v. 6, n. 4, p. 355-385, 1997.
- BARBOSA, E. C. B. **Uma metodologia para a geração de conteúdos digitais baseado na utilização de estudos virtuais com realidade aumentada**. Dissertação (Mestrado em Televisão Digital), Faculdade de Arquitetura, Artes e Comunicação, Universidade Estadual Paulista “Júlio de Mesquita Filho”, Bauru, 2015.
- Bay, H.; Tuytelaars, T.; & Van Gool, L. Surf: Speeded up robust features. In: **Computer vision-ECCV**. p. 404-417, 2006.
- BELL, Max. **Everyday Mathematics Teacher's Reference Manual Grades 1-3**. McGraw-Hill Wright Group, 2007. 350 p.
- BERNARDINI, F.; MITTELMAN, J.; RUSHMEIER, H.; SILVA, C.; TAUBIN, G. The Ball-Pivoting Algorithm for Surface Reconstruction. In: **Journal IEEE Transactions on Visualization and Computer Graphics**. v. 5, n. 4, p. 349-359, 1999.
- BHAT, P.; BURKE, S. PhotoSpace: a vision based approach for digitizing props. **ACM SIGGRAPH 2011 Talks**. p. 1, 2011.
- 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.

BOEHLER, W.; MARBS, A. 3D scanning instruments. In: **Proceedings of the CIPA WG 6 International Workshop on Scanning for Cultural Heritage Recording.** 2002.

BOISSONNAT, J. D. Geometric structures for three-dimensional shape representation. In: **Journal ACM Transactions on Graphics.** v. 3, n. 4, p. 266-286, 1984.

BOUFAMA, B.; MOHR, R.; VEILLON, F. Euclidean Constraints for Uncalibrated Reconstruction. In: **Proceedings 4th International Conference on Computer Vision.** p. 466–470, 1993.

BOYER, K. L.; KAK, A. C. Color-Encoded Structured Light for Rapid Active Ranging. In: **Journal IEEE Transactions on Pattern Analysis and Machine Intelligence.** v. 9, n. 1, p. 14-28, 1987.

BRADSKI, G.; KAEHLER, A. **Learning OpenCV.** O'Reilly Media Inc., 2008.

BREGLER, C.; HERTZMANN, A.; BIERMANN, H. Recovering non-rigid 3D shape from image streams. In: **Proceedings IEEE Conference on Computer Vision and Pattern Recognition.** v.2, p. 690-696, 2000.

BREUER, T.; BODENSTEINER, C.; ARENS, M. Low-cost commodity depth sensor comparison and accuracy analysis. In: **Proceedings SPIE.** v. 9250, p. 1-10, 2014.

BUTZ, A.; HÖLLERER, T.; FEINER, S.; MACINTYRE, B.; BESHERS, 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.

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.

CHOU, C. H.; CHEN, Y. C. Moment-preserving pattern matching. **Journal Pattern**

Recognition. v. 23, n. 5, p. 461-474, 1990.

CHRONSITER, James. **Blender Basics Classroom Tutorial Book.** 2011 146 p. Disponível em: <<http://www.cdschools.org/Page/455>>. Acesso em: 10 janeiro 2015.

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.

CLARKE, T. A.; FRYER, J. G. The development of camera calibration methods and models. **The Photogrammetric Record.** v. 16, n. 91, p. 55-66, 1998.

CORRAL, Michael. **Vector Calculus.** Schoolcraft College, 2011. 220 p. Disponível em: <<http://www.mecmath.net/>>. Acesso em: 5 fevereiro 2015.

CUBIFY. **Sense 3D Scanner.** Disponível em: <<http://cubify.com/en/Products/SenseTechSpecs>>. Acesso em: 17 janeiro 2015.

DANA, K. J.; GINNEKEN, B.; NAYAR, S. K.; KOENDERINK, J. J. Reflectance and texture of real-world surfaces. In: **Journal ACM Transactions on Graphics.** v. 18, n. 1, p. 1-34, 1999.

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

EDELSBRUNNER, H.; MÜCKE, E. P.; Three-dimensional alpha shapes. In: **Journal ACM Transactions on Graphics.** v. 13, n. 1, p. 43-72, 1994.

EGELS, Yves; MICHEL, Kasser. **Digital Photogrammetry.** Bristol: Taylor & Francis, 2001. 376 p.

FARIN, Gerald. **Curves and surfaces for computer aided geometric design: a practical guide.** 5. ed. Morgan Kaufmann, 2001. 520 p.

FELDMAR, J.; AYACHE, N.; Betting, F. 3D-2D projective registration of free-form curves and surfaces. In: **Journal Computer Vision and Image Understanding**. v. 65, n. 3, p. 403-424, 1997.

FIRELIGHT TECHNOLOGIES. **FMOD**. Disponível em: <<http://www.fmod.org>>. Acesso em: 10 janeiro 2015.

FOSTER, Shaun; HALBSTEIN, David. **Integrating 3D Modeling, Photogrammetry and Design**. 1. ed. London: Springer-Verlag, 2014. 113 p.

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.

GOKTURK, S.B.; YALCIN, H.; BAMJI, C. A Time-Of-Flight Depth Sensor, System Description, Issues and Solutions. **Computer Vision and Pattern Recognition Workshop**, 2004. p. 1-9.

GOSHTASBY, A.; GAGE, S. H.; BARTHOLIC, J. F. A Two-Stage Cross Correlation Approach to Template Matching. **Journal IEEE Transactions on Pattern Analysis and Machine Intelligence**. v. 6, n. 3, p. 374-378, 1984.

GOVENDER, N. Evaluation of feature detection algorithms for *Structure from Motion*. **3rd Robotics and Mechatronics Symposium**. p. 1-4, 2009.

GRAU, O. A 3D production pipeline for special effects in TV and film. **IEEE Proc.– Vision, Image Signal Processing**. 2005.

GRAU, O.; BOYER, E.; HUANG, P.; KNOSSOW, D.; MAGGIO, E.; SCHNEIDER, D. Re@ct, Immersive production and delivery of interactive 3D content. **NEM Summit, Istanbul**. 2012.

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.

GROOVER, Mikell. **CAD/CAM: Computer-Aided Design and Manufacturing.** Prentice Hall, 1984. p. 512.

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.

HART, John. **The Art of the Storyboard:** A Filmmaker's Introduction. 2. ed. Focal Press, 2007. 224 p.

HARTLEY, R. I. Theory and Practice of Projective Rectification. **International Journal of Computer Vision.** v. 35, n. 2, p. 115-127, 1999.

HARTLEY, R.; ZISSELMAN, A. **Multiple View Geometry in Computer Vision.** 2. ed. Cambridge University Press, 2004. 670 p.

HAYASHI, M.; ENAMI, K.; Noguchi, H.; FUKUI, K.; YAGI, N.; INOUE, S.; SHIBATA, H.; YAMANOUCHI, Y.; ITOH, Y. Desktop virtual studio system. **IEEE Transactions on Broadcasting.** v. 42, n.3, p. 278-284, 1996.

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

HITL Washington. **ARToolKit.** Disponível em:
<http://www.hitl.washington.edu/artoolkit/documentation/tutorialmulti.htm>. Acesso em: 14 janeiro 2015.

HOPPE, H. Progressive meshes. In: **Proceedings of the 23rd annual conference on Computer graphics and interactive techniques.** p. 99–108, 1996.

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.

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

IZADI, S.; KIM, D.; HILLIGES, O.; MOLYNEAUX, D.; NEWCOMBE, R.; KOHLI, P.; SHOTTON, J.; HODGES, S.; FREEMAN, D.; DAVISON, A. KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera. **Proceedings of the 24th annual ACM symposium on User interface software and technology**. p. 559-568, 2011.

JANCOSEK, M.; PAJDLA, T. Multi-view reconstruction preserving weakly-supported surfaces. **IEEE Conference on Computer Vision and Pattern Recognition**. p. 3121-3128, 2011.

JANCOSEK, M.; PAJDLA, T. **CMPMVS**. Disponível em: <<http://ptak.felk.cvut.cz/SfMservice/webSfM.pl?menu=cmpmvs>>. 2012. Acesso em 17 janeiro 2015.

KATO, H.; BILLINGHURST, M.; POUPYREV, I. **ARToolKit user manual version 2.33**. 2000.

KATO, H. ARToolKit: library for Vision-Based augmented reality. **IEICE, PRMU**. v. 6, p. 79-86, 2002.

KAZHDAN, M.; BOLITHO, M.; HOPPE, H. Poisson surface reconstruction. In: **Proceedings of the fourth Eurographics symposium on Geometry processing**. (S.l.: s.n.), p. 61–70, 2006.

KOLLURI, R.; SHEWCHUK J. R.; O'BRIEN, J. F. Spectral Surface Reconstruction from Noisy Point Clouds. **Symposium on Geometry Processing**. Nice, 2004.

LI, Z.; WANG, J. Least squares image matching: A comparison of the performance of robust estimators. **ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences**. v. 2, p. 37-44, 2014.

LI, H.; QI, M.; WU, Y. A Real-Time Registration Method of Augmented Reality Based on Surf and Optical Flow. **Journal of Theoretical and Applied Information Technology**. v. 42, n. 2, p. 281-286, 2012.

LINSEN, L.; MÜLLER, K.; ROSENTHAL, P. Splat-based ray tracing of point clouds. In: **Journal of WSCG.** v. 15, n. 1-3, p. 51-58, 2007.

LOOP, C.; ZHANG, Z; Computing rectifying homographies for stereo vision. In: **IEEE Computer Society Conference on Computer Vision and Pattern Recognition.** v. 1, p. 125-131, 1999.

LOOSER, J.; GRASSET, R.; SEICHTER, H.; BILLINGHURST, M. OSGART – A pragmatic approach to MR. **International Symposium of Mixed and Augmented Reality (ISMAR).** p. 22-25, 2006.

LOWE, David;. Distinctive image features from scale-invariant keypoints. **International Journal of Computer Vision,** n. 60, p. 91-110. 2004.

LUEBKE, David; WATSON, Benjamin; COHEN, Jonathan D.; REDDY, Martin; VARSHNEY, Amitabh. **Level of Detail for 3D Graphics.** 1. ed. New York: Elsevier Science Inc, 2002. 432 p.

MALIK, A. S. Depth Map and 3D Imaging Applications: Algorithms and Technologies: Algorithms and Technologies. **IGI Global,** 2011. 648 p.

MCHENRY, K.; BAJCSY, P. An overview of 3d data content, file formats and viewers. **National Center for Supercomputing Applications.** 2008

MICROSOFT. **DirectX 6.0.** Disponível em:
<http://www.microsoft.com/msj/0199/direct3d/direct3d.aspx>. Acesso em: 15 janeiro 2015.

MIKKONEN, T.; TAIVALSAARI, A.; TERHO, M. Lively for Qt: A Platform for Mobile Web Applications. **Proceedings of the 6th International Conference on Mobile Technology, Application & Systems.** p. 24, 2009.

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

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.

OPEN KINECT. Open Kinect Project. Disponível em:
http://openkinect.org/wiki/Main_Page. Acesso em: 20 janeiro 2015.

ORRITE, C.; HERRERO, J. E. Shape matching of partially occluded curves invariant under projective transformation. In: **Computer Vision and Image Understanding**, v. 93, n. 1, p. 34-64, 2004.

PAPADOPOULOS, Apostolos N. **Nearest Neighbor Search: A Database Perspective**. Pittsburgh: Springer Science & Business Media, 2006. 178 p.

PAPADIMITRIOU, V.; DENNIS, T. J. Epipolar line estimation and rectification for stereo image pairs. In: **IEEE Transactions on Image Processing**. v. 5, n. 4, p. 672-676, 1996.

PIEGL, Les; TILLER, Wayne. **The NURBS Book**. 2. ed. Tampa: Springer, 1997. 649 p.

POLLEFEYS, Marc. **Visual 3D Modeling from Images**. Tutorial Notes, University of North Carolina. Chapel Hill, USA, 2002.

PROESMANS, M.; VAN GOOL, L. Reading between the lines, a method for extracting dynamic 3D with texture. In: **Proceedings of the ACM symposium on Virtual reality software and technology**. p. 95-102, 1997.

QUAN, Long. **Image-Based Modeling**. Kowloon: Springer US, 2010. 271 p.

RABAUD, V.; BELONGIE, S. Re-thinking non-rigid *Structure from Motion*. In: **IEEE Conference on Computer Vision and Pattern Recognition**. p. 1-8. 2008.

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

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

RICHMOND, Richard D.; CAIN, Stephen C. Direct-Detection LADAR Systems. **Bellingham: SPIE**, 2010. 154 p.

RUSINKIEWICZ, S.; HALL-HOLT, O.; LEVOY, M. Real-time 3D model acquisition. In: **ACM Transactions on Graphics Real-time 3D model acquisition.** v. 21, n. 3, p. 438-446, 2002.

RUSU, R. B.; COUSINS, S. 3d is here: Point cloud library (pcl). In: **IEEE International Conference on Robotics and Automation.** p. 1-4, 2011.

SANCHES, S. R. R.; TOKUNAGA, D. M.; SILVA, V. F.; SEMENTILLE, A. C.; TORI, R. Mutual occlusion between real and virtual elements in augmented reality based on fiducial markers. In: **2012 IEEE Workshop on Applications of Computer Vision.** p. 49-54, 2012.

SCHULTZ, C. Digital keying methods. In: **University of Bremen Center for Computing Technologies.** Tzi. v. 4, n. 2, 2006.

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.

SHAPIRO, L. G.; HARLICK, R. M. Structural descriptions and inexact matching. In: **IEEE Transactions on Pattern Analysis and Machine Intelligence.** v. 5, p. 504-519, 1981.

SHIMODA, S.; HAYASHI, M.; KATANTSUGU, Y. New chroma-key imagining technique with Hi-Vision background. In: **IEEE Transactions on Broadcasting.** v. 35, n.4, p. 357-361, 1989.

SILVA, Daniel Carneiro da. **Special Applications of Photogrammetry.** InTech, 2012. 146 p. Disponível em: <<http://www.intechopen.com/books/special-applications-of-photogrammetry>>. Acesso em: 10 janeiro 2015.

SNAVELY, N.; SEITZ, S. M.; SZELISKI, R. Photo tourism: exploring photo collections in 3D. In: **ACM transactions on graphics.** v. 25, n. 3, p. 835-846, 2006.

SURLYKKE, A.; PEDERSEN, S. B.; JAKOBSEN, L. Echolocating bats emit a highly directional sonar sound beam in the field. In: **Proceedings of the Royal Society B: Biological Sciences.** v. 276, n. 1658, p. 853-860, 2009.

TAN, T. N.; SULLIVAN, G. D.; BAKER, K. D. Recovery of Intrinsic and Extrinsic Camera Parameters using Perspective Views of Rectangles. In: **BMVC**. p. 1-10, 1995.

THELIN, Johan. **Foundations of Qt development**. Apress, 2007. 528 p.

THORMÄHLEN, T.; BROSZIO, H.; MIKULASTIK, P. Robust linear auto-calibration of a moving camera from image sequences. In: **Computer Vision–ACCV**. p. 71-80, 2006.

TORR, P. H.; MURRAY, D. W. The development and comparison of robust methods for estimating the fundamental matrix. In: **International journal of computer vision**. v. 24, n. 3, p. 271-300, 1997.

TORRESANI, L.; HERTZMANN, A.; BREGLER, C. Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. In: **IEEE Transactions on Pattern Analysis and Machine Intelligence**. v. 30, n. 5, p. 878-892, 2008.

TRIGGS, B. Autocalibration and the absolute quadric. In: **Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition**. p. 609-614. 1997.

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

VANDE H.; JOSHUA D. **A Novel Lidar Ceilometer**. Springer, 2015. 158 p.

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

WANG, Rui; QIAN, X. **OpenSceneGraph 3.0: Beginner's guide**. Packt Publishing Ltd, 2010. 412 p.

WETA. **Features**. Disponível em: <<http://www.wetafx.co.nz/features>>. Acesso em: 14 janeiro 2015.

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

WU, C. Towards linear-time incremental *Structure from Motion*. In: **International Conference on 3D Vision-3DV 2013**. p. 127-134, 2013.

WU, Changchang. **VisualSfM: A Visual Structure from Motion System**. Disponível em

<<http://ccwu.me/vSfM/>>. 2011. Acesso em: 20 janeiro 2015.

YAMANOUCHI, Y.; MITSUMINE, H.; FUKAYA, T.; KAWAKITA, M.; YAGI, N.; & INOUE, S. Real space-based virtual studio seamless synthesis of a real set image with a virtual set image. In: **Proceedings of the ACM symposium on Virtual reality software and technology**. p. 194-200, 2002.

ZHANG, Z. (1998). Determining the epipolar geometry and its uncertainty: A review. **International journal of computer vision**. v. 27, n. 2, p. 161-195, 1998.

ZHANG, Z. Camera calibration with one-dimensional objects. In: **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 26, n. 7, p. 892-899, 2004.

ZHANG, Z. Microsoft Kinect Sensor and Its Effect. **IEEE Multimedia**, v. 19, n. 2, p. 4-10, 2012.

ZHANG, Z.; DERICHE, R.; FAUGERAS, O.; LUONG, Q. T. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. In: **Artificial intelligence**. v. 78, n. 1, p. 87-119, 1995.

ARSTUDIO: A virtual studio system with Augmented Reality features

Tiago De Gaspari, Antonio Carlos Sementille, Daniel Zuniga Vielmas, Ivan Abdo Aguilar, João Fernando Marar
Computer Science Department
São Paulo State University (UNESP), Bauru, Brazil
{gaspari, semente, danielvielmas, ivan_aguilar, fermarar}@fc.unesp.br

Abstract

A system that benefits the production of content for digital TV and cinema is named *Virtual Studio*. A virtual studio system can use technologies such as augmented reality and digital matting in order to reduce production costs while providing the same resources of a conventional studio. Thus, it enables current studios, with low cost and using conventional devices, to create productions with greater image quality and effects, in real-time.

Based on this context, this paper presents the design and implementation of a virtual studio system called ARSTUDIO, which allows the inclusion of special effects in real time, different from current systems where the special effects are added during the post-production step of a pipeline. This approach proves to be quite interesting, flexible and innovative, since it can save time and possibly avoid rework by optimizing the production pipeline of audiovisual content.

CR Categories: H.5.1 [Information Systems]: Multimedia Information Systems—Artificial, augmented, and virtual realities I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual reality;

Keywords: Virtual Studio, Augmented Reality, Digital Matting

1 Introduction

In recent decades, the improvements in the technology of computer hardware caused drastic changes in the production of content for cinema and television, particularly with regard to the recording and post-production steps. With smaller, lighter and more flexible equipments, studios began to employ virtual images manipulation, thus using the full potential of computers to perform the video processing in real time.

That was how the concept of Virtual Studio had been created: a studio that allows the composition of real video with synthetic images. Also called *Virtual Reality in Third Person*, this composition technique allows people who watch this “mixed signal” to see other people (actors) and physical objects combined with a virtual environment [Gibbs et al. 1998].

Today, the use of computer graphics (CG) techniques allowed the creation of special effects and animation for movies and programs

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

VRCAI 2014, November 30 – December 02, 2014, Shenzhen, China.

Copyright © ACM 978-1-4503-3254-5/14/11 \$15.00

<http://dx.doi.org/10.1145/2670473.2670491>

for television and cinema. However, the production costs of softwares that involves CG are still relatively high. Considering the production pipeline currently used for digital television in particular, the risks are high, if any error occurs during the recording stage. For example, the impossibility of integrating a virtual character (computer generated), due to the fact that the camera angle or the position of the actors were wrong in the real images.

According to Grau [2000], usually the production of any television program or film which involves the use of CG, passes through three technical stages:

- *Planning Stage*: the conceptual ideas, usually represented in form of storyboard, are transformed into a screenplay, along with a list of scenes and technical instructions on how to obtain them;
- *On-set Stage*: in this stage, the movie is recorded, according to the script;
- *Post-production Stage*: the virtual content is integrated into the actual recording and scenes are edited into the final footage movie film.

In this workflow, the three-dimensional (3D) content is inserted only during the post-production stage, and thus the virtual content is visible only after the entire studio recording has been completed.

In a real studio, the director, actors and camera operators have rely on simple visual signals, such as a mark on the floor, to be guided, where the virtual characters or objects should appear. In this traditional approach, many problems often occur. A very common problem is the framing of the camera, that can be wrong at this stage. This makes the integration of the virtual objects into the scene, in post-production, difficult or impossible. The problem of “line of sight” is also very common. It happens when the actor is not looking at the virtual object, since it is not visible during the recording stage. Another problem is the bidirectional interaction between real actors and virtual elements. Considering the occurrence of these problems, the worst case is when the recording in the real studio has to be remade which reflects on high production costs [Grau 2005].

However, complex film productions can benefit from the techniques of “scenes combination” provided by Augmented Reality (AR), the application of these techniques to Virtual Studio systems can prove to be an approach to content generation, quite flexible and innovative. Therefore, the main goal of this project is to investigate and apply the techniques of AR, Virtual Reality (VR) and CG in a computational tool that allows creation of Virtual Interactive Studios. This research involves problems such as the registration of virtual objects, the digital matting for creating complex virtual scenes, mutual occlusion and three-dimensional reconstruction of actors and environments in order to obtain high quality content for digital television and internet, in real time. As a consequence, this work proposes a modification on the content production pipeline, reducing the time and cost of the post-production process.

2 Related Work

Full virtual scenarios, generated in real time, began to appear in Japan in 1991. The Japan Broadcast Company (NHK) used a prototype of a virtual studio system to produce a scientific documentary called “Nano Space”[Akiyama et al. 1993]. This prototype already presented the main aspects of a virtual studio system: background rendering with a tracking system of the camera in real time. However, the performance of the graphics hardware of that time were a hindrance to the development plans of NHK [Hayashi 1998]. In 1993, major improvements in the graphics hardware architecture made the first commercial virtual studios arise. Until then, the existing systems of virtual studio were for internal use of tv transmission companies [Gibbs et al. 1998].

The main features of the virtual studio systems are the focus of several studies. Some fields of interest of this area includes the camera tracking [Ward et al. 1992][Thomas et al. 1997]; the AR objects generation[Lalioti and Woolard 2003][Blonde et al. 1996]; the system feedback for the actor [Fukaya et al. 2002][Grau et al. 2004]; and the foreground extraction using chroma-key [Grau et al. 2004][Fukui et al. 1994], multiple cameras [Thomas et al. 2004] and time-of-flight of infra-red light[Iddan and Yahav 2001]. Or even combining some of these techniques for creating AR systems for content production [Bartczak et al. 2008].

Although Virtual Studios provides many advantages, it is rarely used today. Instead, directors still prefer to use storyboards drawn manually and cardboard models of film sets. To be attractive to the users in the production stage, this type of tool should be easy to use, having an interface with familiar terms and containing basic three-dimensional objects ready for use, to save time and to expedite the process as a whole [Thomas 2006]. Examples of works to produce tools specifically designed for pre-visualization is that of Higgins [1994] and the work of Ichikari et al. [2008] which includes the pre-visualization to the actors by a rehearsal system. More recently, commercial products began to appear on the market such as these two [Viz Virtual Studio] [VSET 3D].

Even if the pre-visualization does not necessarily need AR techniques, this is an area where computer graphics can help with the production of content for television and the film industry, making the process of content generation easier, faster and cheaper [Thomas 2006].

3 System Structure

The developed system was divided into several modules, due to the different features implemented. The modules, their functions and how the informations flow between them, are presented in this section.

The modules that constitute the system and the information flow are shown in Figure 1.

The functions of the modules shown in Figure 1 are described in the following:

- *Video Capture Module*: Responsible for capturing video from the camera and transmitting this information to other modules for processing;
- *User Interface Module*: Allows user to configure the scenes and interact with the whole system;
- *Marker Detection Module*: Using computer vision techniques, it receives the video from video capture module, identifies and tracks the markers present in the actual scene;

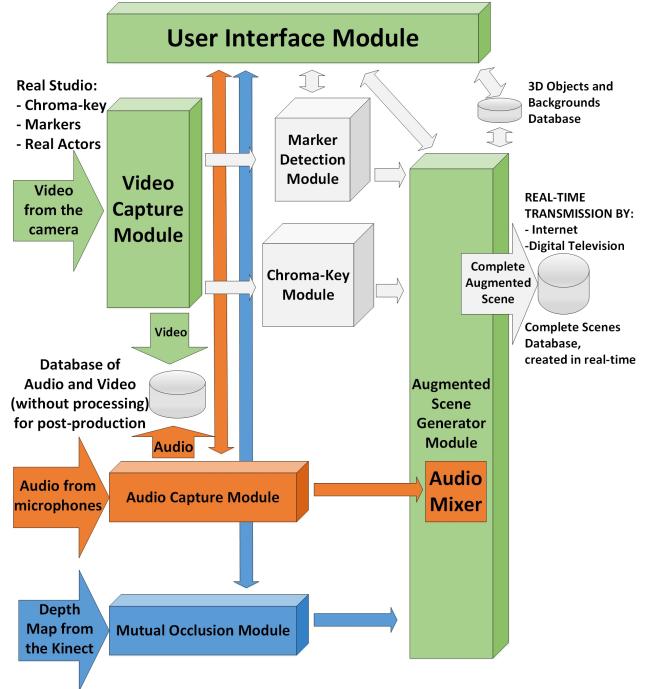


Figure 1: Modular structure of the system.

- *Chroma-key Module*: Performs digital matting techniques, for replacing the background image of the real scene, using methods based on chroma-key;
- *Audio Capture Module*: Responsible for capturing the audio from microphones in real scenario;
- *Mutual Occlusion Module*: Performs, with depth information, the mutual occlusion between virtual objects and the real actors in the scene;
- *Augmented Scene Generator Module*: From the information generated by all other modules, generates the final augmented scene, in real time, for transmission over the internet or digital TV, or saves the video stream locally.

The augmented scene generated by the system is structured as a scene graph, which allows to create complex virtual scenes incrementally, where several objects can be correlated, and change them in real time, via the *User Interface Module*.

In order to complement the modules and to test the whole system, we created a database of three-dimensional models to be included in the scenes, beyond a basic set of images and videos to use as arbitrary backgrounds.

To develop the system’s computer application, we chose to use free, multi-platform and opensource software libraries. Figure 2 shows the hierarchy of the libraries used for building of the system. In this hierarchy, the application layer is the highest level, based on the Qt development framework [Thelin 2007]. The OpenCV library [Bradski and Kaehler 2008] fits into an intermediate level between application and operating system, as well as the libfreenect library [OpenKinect Project] and the FMOD API [FMOD]. Below, VR (OpenSceneGraph [Wang and Qian 2010]) and AR (ARToolKit [Kato 2002]) toolkits are located, and the OsgART library [Loosser et al. 2006] presents one level that encompass the previous two. Toward the hardware, is the OpenGL graphics library, which supports the other libraries, and finally the operating system.

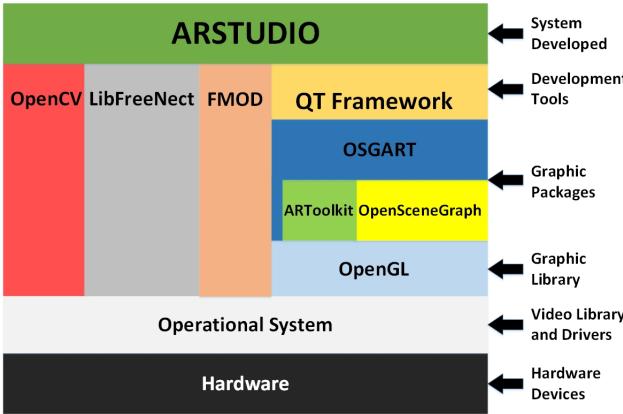


Figure 2: Library Hierarchy

4 System Development

The system was developed in a modular and incremental process, as described in Section 3.

Because it's a system that generates real-time content, the user interface was created so that the structure of the scene can be changed at runtime and the changes in the scene can be viewed by users. Furthermore, by allowing the use of different effects, which are described along this section, the interface allows them to be activated at any time by giving visual feedback to the user in real time.

Figure 3 shows the user interface, in which the camera image, that is captured by the camera and is already processed by the system, is presented on the left, while the controls of implemented functionalities are presented on the right side of the image.

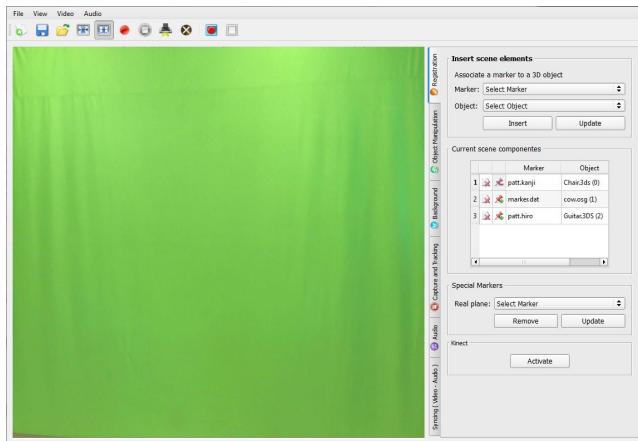


Figure 3: User Interface.

4.1 Chroma-Key and Digital Matting

One of the features that has been developed performs the segmentation of the video captured by the camera using chroma-keying methods in order to perform the digital matting, changing the background of the scene with a more appropriate static image or video.

Two different methods of digital matting, based on chroma-key, were implemented and incorporated into the system. One of them applies the algorithm proposed by van den Bergh and Laliotti, [1999] generates a binary matting, i.e. with only two levels of

transparency (total transparency or total opacity), a simple way to improve the quality of this method's results is using a second algorithm that applies various levels of transparency in the pixels (partial transparency).

The algorithm implemented for this purpose was the key color difference [Petro 1971]. This algorithm calculates the transparency based on the difference of the pixels color channels.

Another filter that is applied to improve the results of the matting algorithms is the blurring filter. This filter is used in two stages: an initial step to reduce the noise of the input image, i.e. reduce inaccurate color and brightness variations produced by the camera during image capture; and a final step to soften the edges of the extracted element by the matting algorithm [Wright 2013]. These initial and final stages are called pre-blurring and post-blurring respectively.

The application of the blurring filter on images brings a negative effect: the loss of image details [Wright 2013]. But the effects of applying the filter to digital matting are minimized by how it is applied: in the pre-blurring step, a copy of the input image is blurred and used to calculate the values of transparency of pixels, but the image used for compose the final result is the image of the original (not blurred) input; in post-blurring step, the filter is applied only in the alpha channel of the image (channel containing the opacity values), and not the image color channels, preserving the details of the pixel color.

In addition to choosing which algorithm will be used for the matting, the parameters of the blurring filter and the chosen algorithm can be manipulated by the user. Figure 4 shows the software interface, in which the algorithm, and its parameters, can be chosen. In Figure 4(a) the selected algorithm is chroma-key [van den Bergh and Laliotti 1999] and in Figure 4(b) the color key difference algorithm [Petro 1971]. Options such as different background colors and other parameters, allow the software to have good results in different environments, requiring only that the user configures the best parameters for each scene or environment.

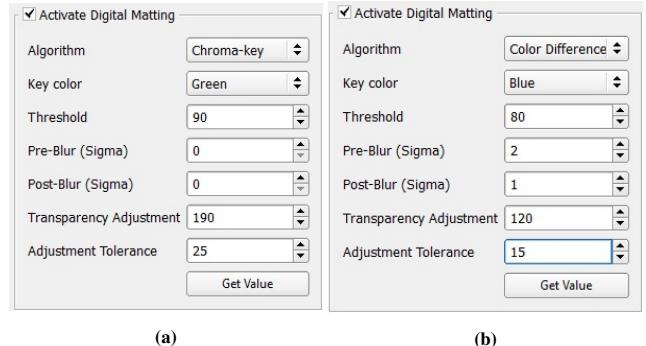


Figure 4: Digital Matting Algorithm Selection. (a) Interface with chroma-key algorithm selected; (b) Interface with color difference key algorithm selected.

4.2 Augmented Reality

One of the main features of the system presented in this paper is the insertion of virtual objects in real scene by means of AR techniques. For this purpose, the ARToolkit library [Kato 2002] performs a trace in each frame of the input video stream in search of markers registered in the system. When the pattern of a marker is recognized, its position and orientation, with respect to the camera, are calculated.

Then, the OSGART library [Looser et al. 2006] translates the position and orientation of the marker into a transformation matrix that will be inserted as a transformation node in the scene graph of OpenSceneGraph [Wang and Qian 2010]. If a 3D model (virtual object) is inserted as a child node of this transformation node, OSGART will apply the transformation matrix to the 3D model, so the model will be translated and rotated according to this matrix, which represents the marker in the virtual world. The virtual scene is then drawn by the virtual camera with the input image of the actual camera placed in the background of the scene; the visual impression given by the result is that the 3D model is in the same position as the marker in the real world, creating the effect of augmented reality.

Once the real camera is fixed in a position in the real world, the user can fix a virtual object in the scene in a determined position. Once the object position is fixed, the system halts the continuous update of the object projected position on the scene. The virtual object attached to the marker will always be visible and stationary, even if the marker is no longer visible by the camera in the real world.

To give the user control over the visible markers and virtual objects which are attached to these markers, the user interface was designed as is shown in Figure 5.

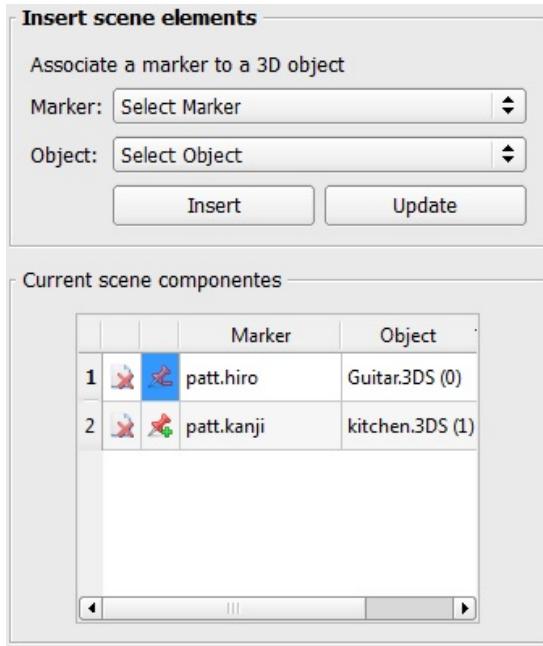


Figure 5: Interface for the association between the markers and virtual objects.

In the interface shown in Figure 5, the user can attach an object to a marker and insert it to the scene. After the object is inserted to the scene, a new row in the components of the current scene table is inserted, with the marker and object names. To fix the marker position, the user just needs to click the button with the pin icon. The marker position will be fixed and the icon will change, indicating that the procedure was completed. To unpin the marker, the user simply clicks the icon again and it will return to its normal state.

To improve the tracking of markers and prevent occasional occlusions, the concept of multi-marker was used. In this case, an object is represented by several different markers, associated in a predetermined way with each other. Thus, an object is visible in the scene if at least one of the markers is identified.

4.3 Registration of virtual objects using a depth sensor

This important feature of the system allows the registration of virtual objects in relation to the actors, enabling the generation of a scene with spatial coherence between real and virtual elements. For this feature, a Microsoft Kinect sensor was used to facilitate the registration.

4.3.1 Kinect Depth Map

The depth information obtained by Kinect is represented by a disparity map. According to Zhang et al. [2012] this map is calculated through a comparison between the standard Infrared (IR) projector known points (pattern memorized by the Kinect of a plan with a known depth) and the pattern of points captured by the IR camera in the environment. All the memorized points are compared with the captured points in search of a correlation, ie, the same point that is in one of the patterns must be found in the another pattern. After obtaining the correlation, the difference between the position of a point relative to the other is obtained: this is called disparity. Having known the depth of the stored pattern and the value of disparity it's possible to estimate the value of depth by triangulation. After applying the triangulation to all the values of the disparity map, the depth map is obtained [Zhang 2012]. In the depth map, each pixel represents the distance between the Kinect and the environment at that point. To find the depth of a point of a image captured by the RGB camera, it's necessary to find that point in the depth map. However, the relationship between the image of the RGB camera and depth map is not straightforward, due to the distance between the RGB camera and the IR camera, so the pictures are not aligned [Andersen et al. 2012]. Figure 6 shows the depth map overlayed on the RGB image. In Figure 6(a), no alignment was performed; In Figure 6(b), the two images are aligned.



Figure 6: Overlay of RGB image and depth map. (a) Overlay without alignment; (b) Overlay with alignment. [Andersen et al. 2012].

In the main libraries created to access the features of Kinect, this alignment is done automatically from known parameters of the device cameras. However, this alignment only works for the Kinect RGB camera, because the known parameters are only applicable to its camera. To use an external RGB camera, it's necessary to find the cameras parameters and perform the alignment between them. As the Kinect RGB camera features a low resolution of image (640x320 pixels at 30 fps), we chose to use the calibration process with an external camera. Thus, any video camera can be used, improving the quality of system-generated videos.

4.3.2 Calibrating the Kinect depth camera to an external RGB camera

The first step in the calibration process is to rigidly attach the external camera to the Kinect. In our tests we used a webcam (Logitech

HD Pro Webcam C910 model) attached to the Kinect, as shown in Figure 7.



Figure 7: External camera attached to Kinect.

To discover the intrinsic and extrinsic camera parameters, the algorithm developed by Herrera, Kannala and Heikkil [2012] in MATLAB (numerical computing environment) was used. By detecting the corners of a checkerboard, the algorithm can calculate the intrinsic parameters of focal length and the lens distortion coefficient of both cameras (RGB and IR camera), besides the rotation and translation matrices that should be applied on the depth map to align with the RGB image.

For better results, at least 20 pictures of the checkerboard in different positions, distances and angles to the camera should be captured by the RGB camera along with the disparity map corresponding to the RGB image, and provided to the algorithm to calculate the parameters [Herrera C. et al. 2012].

The detection of the checkerboard corners in RGB images is done automatically using methods of computer vision while the depth map corners must be selected manually, since they are not visible in the depth map. Figure 8 shows an example of a RGB image and a depth map used by the algorithm that computes the parameters.

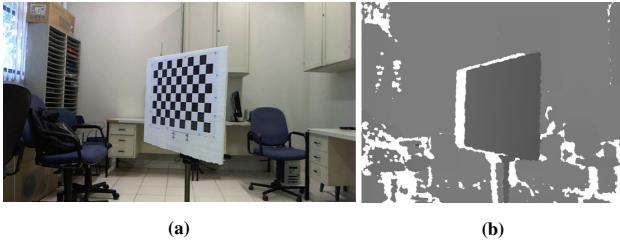


Figure 8: Checkerboard images for calibration. (a) RGB image of the external camera; (b) Depth map corresponding to the RGB image captured by Kinect.

The parameters obtained by the algorithm were stored in a file that is loaded when the ARSTUDIO boots. Having the intrinsic parameters of focal length, the lens distortion coefficient and the rotation and translation matrices, a method of calibration had to be applied to these parameters. In ARSTUDIO, the calibration method used was the method of Nicolas Burrus [2011].

With the depth value of the pixel calculated, it's possible to project

each pixel of the disparity map in a 3D space, transforming each pixel into a 3D point (with x, y and z coordinates).

After obtaining the 3D point of each pixel, rotation and translation, the alignment of the disparity map with the RGB image coordinates, is applied. However, the resolution of the disparity map provided by Kinect is 640x480 pixels, which is lower than the resolution typically used in RGB HD cameras. With this, the amount of 3D points is smaller than the amount of pixels of RGB image, creating a depth map with sparse pixels. To fill the empty spaces of the generated depth map, an interpolation of the map was necessary. With the depth map generated and calibrated on every frame of a video, the depth values of the pixels of the RGB image can be obtained and know if a pixel is ahead of or behind a rendered virtual object pixel.

4.3.3 Mutual Occlusion

After calibrating the devices, the next step in this development stage is to create a way of performing the occlusion of virtual objects that are behind the real-world objects, using the depth information.

To assign the value of depth for each pixel, a fragment shader was created. The fragment shader is compiled by the OpenGL and executed by the video card for each fragment of a virtual object, ie, the scheduled operations in the fragment shader are performed on each pixel of the object [Shreiner et al. 2013]. The properties that can be changed by the fragment shader are the color and the depth of the fragment. As the purpose of its use in ARSTUDIO is to assign the value of each pixel of the depth map, the property modified by the fragment shader created was the depth of the fragment.

The depth map obtained by the Kinect was then assigned to a cutting surface as a texture. As the cutting surface is transparent, the texture is not rendered, but serves as input to the fragment shader. Analyzing the pixels of the texture (depth map), the fragment shader converts the depth values, in meters, to depth units of the virtual world. After that, it calculates the depth value that will be sent to the depth buffer (z-buffer) of the video card by OpenGL. The depth values sent by OpenGL need to be calculated by the fragment shader, because the OpenGL uses normalized values, ie, those values do not represent the coordinates of the virtual world, but a value between 0 and 1 [Shreiner et al. 2013]. This normalization is made with relation to the view frustum of the virtual camera.

The viewing frustum of a virtual camera determines which objects should be rendered, dismissing the rendering process of objects that are out of it. This is done to save computer processing and increase its performance because, if the object doesn't appear on the screen, there is no need for all the mathematical operations usually needed for rendering. The viewing frustum of the virtual camera is determined by your angle of view, and two planes which determine the limits of the frustum: the near plane and the far plane. These planes determine the minimum and maximum depth that is viewed by the virtual camera. Figure 9 shows the viewing frustum, emphasized between near plane and far plane.

The normalization done by OpenGL considers the near plane and the far plane to determine the extreme values of depth. With the depth coordinates of these planes, a non-linear equation is applied to normalize the value of a given depth coordinate. This normalization is performed according to Equation 1 [Hoff III 1998], in which z' represents the depth value normalized, z represents the depth coordinate to be normalized, z_{near} represents the near plane depth coordinate and the z_{far} represents the far plane depth coordinate.

$$z' = \frac{\frac{1}{z_{near}} - \frac{1}{z}}{\frac{1}{z_{near}} - \frac{1}{z_{far}}} \quad (1)$$

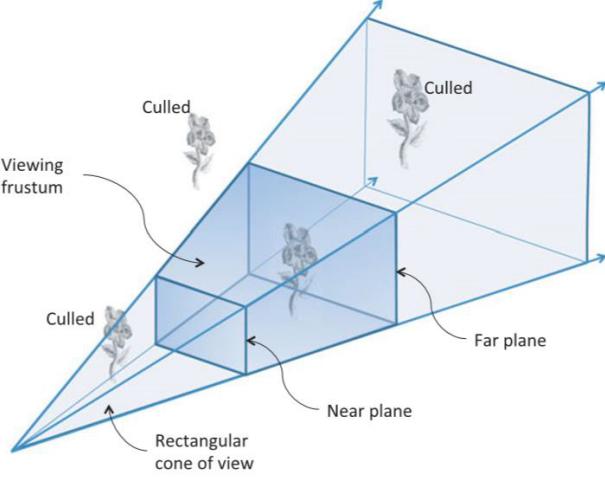


Figure 9: Viewing Frustum of a Virtual Camera [Shreiner et al. 2013].

Figure 10 shows the fragment shader code, in C++ language. The depth value in meters is obtained from the depth map and converted to depth coordinates of the virtual world (in the code, 1000 coordinate units of the virtual world are equivalent to 0.5 meters). If the depth value is unknown by Kinect (value 0), the coordinate assigned is the maximum, i.e. the depth coordinate of the far plane (z_{far}). After converting the value of the depth map, the depth values are limited by the coordinates of the near and far planes (a value can't be lower than z_{near} or higher than z_{far}), and then the depth value of the fragment is normalized.

```
uniform sampler2D depthTexture;
uniform float zNear;
uniform float zFar;
void main(void)
{
    vec4 depth = texture2D(depthTexture, gl_TexCoord[0].xy);
    float z;
    if (depth.r == 0)
        z = zFar;
    else
        z = depth.r * 1000 / 0.50;
    if (z < zNear)
        z = zNear;
    else
        if (z > zFar)
            z = zFar;
    gl_FragDepth = (1/zNear - 1/z) / (1/zNear - 1/zFar);
}
```

Figure 10: Fragment shader code.

With the normalization done in fragment shader, the depth buffer of the video card receives the depth values for each pixel and applies it in the rendering process, creating a surface that receives the values from the Kinect depth map. Thus, the surface becomes a virtual representation of the surface of the real world captured by the camera, occluding all virtual objects that are behind it, allowing the registration of the virtual objects in relation to real-world objects, including actors on stage.

4.4 Audio

The capture of the audio from the environment is done in a separate module, triggered by the user interface in the same way as the other

modules.

This module uses the FMOD API [FMOD] and only capture audio synchronized with the video of the scene from the microphones. The system allows the user to save the audio and video in a single file or in different files for the use in the post-production stage.

5 Results

As a result, the system is capable of generating complex scenes using AR, including the effect of digital matting, addition and manipulation of virtual objects in real scene and the mutual occlusion between these objects and the real actors. Moreover, it enables capturing video (both the raw video coming from the camera, and the video with the effects applied) and audio of the environment.

Figure 11 shows the result of a test, in which the hand of a person passes through a virtual object, associated with a real marker. In this case, the person hand is between the chessboard and its pieces, because of it, the hand just passed through the board but not the pieces.

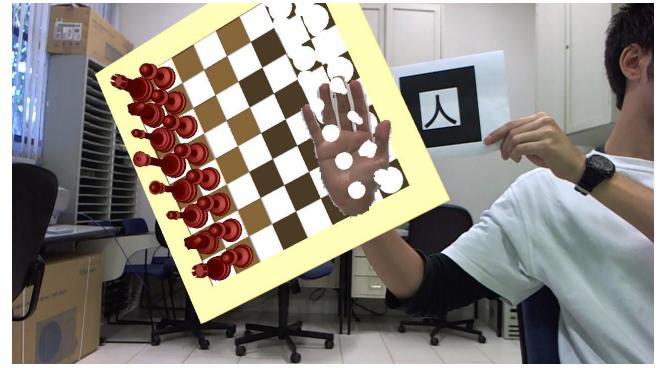


Figure 11: Occlusion effect when a real object passes through a virtual object.

Figure 12 shows an example of how mutual occlusion works between an actor and a virtual object at two different frames in the same scene. In it, it's possible to see how the actor can be located in front of a virtual object or behind it, depending on his movement in the scene.

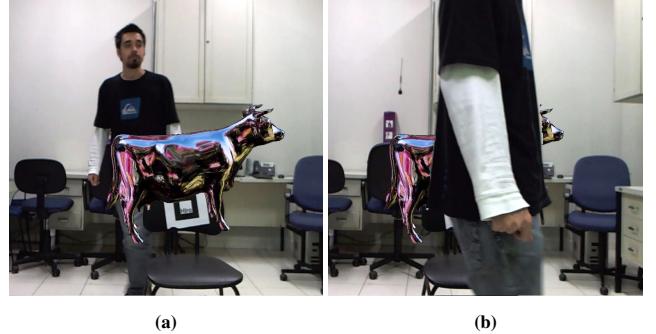


Figure 12: Example of the mutual occlusion effect. (a) The actor is behind the object; (b) the actor is in front of the object.

It is important to note in Figure 12 that only a part of the actor body hides the virtual object when it is behind the actor. This occlusion is performed pixel to pixel, ie, parts of the body of the actor can be in front of the object, while others are behind.

Figure 13 shows this occlusion effect. In this scene, the actor is behind the virtual object (in this case, an electric guitar), but with his hands in front of it. In addition, the background of the scene presented has been changed too, using the methods of chroma-key present in the system.



Figure 13: Scene background change and mutual occlusion between virtual object and actor.

With this it's possible that an actor can be in the middle of a completely virtual scenario, unlike what happens when only the background is replaced. In Figure 14 a scene is presented where the user is inside a virtual object (in this case, a virtual kitchen) with some parts of it behind the actor and others in front of him.

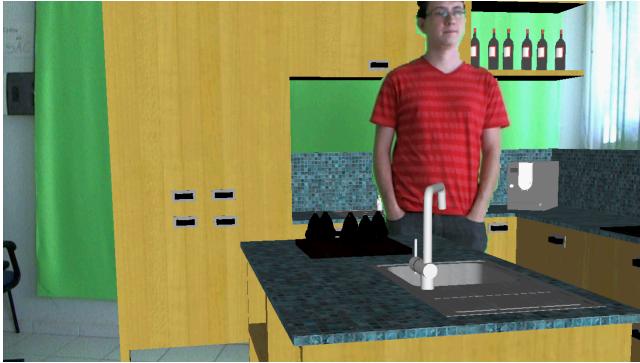


Figure 14: Mutual occlusion effect when the actor is inside a virtual object.

Observing the scene shown in Figure 14, however, it's possible to see that there remains a green border around the actor, since only the depth information is used in the segmentation. We intend, as a continuation of this work, conduct a more refined edge treatment in order to minimize this effect.

Figure 15 shows a complex scene built in ARSTUDIO. This scene presents the image captured by the video camera with the modified background using chroma-key and several virtual objects positioned in the scene. Furthermore, different levels in the scene show how mutual occlusion works: some virtual objects are behind or in front of the real actors depending on their positions (calculated using the depth sensor).



(a)



(b)

Figure 15: Scene built by the system; (a) Image captured by the camera without any processing; (b) Real-time rendered image displayed to the user.

6 Conclusion and Future Works

This paper presents a system for virtual studios called ARSTUDIO. With the results obtained, it's possible to observe how the addition of virtual elements and special effects, in real time, allow the user to have a preview of how the video will look like after going through the post-production, and correct problems at any stage of the generation of content, even during filming. The system innovation is optimizing the pipeline of digital content generation with virtual elements and special effects.

In this new structure, using the techniques of virtual and augmented reality, animations, objects and virtual scenarios can be viewed and manipulated by directors, actors and cameramen, in the course of production.

Currently, two video streams can be captured, transmitted and/or saved locally: one is the video with no effects applied, for use in post-production stage where special effects and more complex virtual objects can be inserted; and the video processed by the system, with special effects and virtual objects. As future work, we intend to, besides improved segmentation using depth information (quoted in section 5), capture the scene graph that composes the augmented scene, as well as the virtual objects and their positions in all the frames in the captured video and export this information to several softwares that are used traditionally in the post-production step such as Maya, 3ds Max and others.

Acknowledgements

ARSTUDIO is supported in part by the National Counsel of Technological and Scientific Development (CNPq) through a Master degree scholarship granted to Tiago De Gaspari and by Agência Unesp de Inovação (AUIN).

References

- AKIYAMA, T., HAYASHI, M., YAMANOUCHI, Y., KOBAYASHI, M., AND SATO, M. 1993. The new picture synthesis techniques of real time computer graphics and live camera picture on nhk special “nano space”. *ITEJ Technical Report* 17, 23, 13–18.
- ANDERSEN, M. R., JENSEN, T., LISOUSKI, P., MORTENSEN, A. K., HANSEN, M. K., GREGERSEN, T., AND AHRENDT, P. 2012. Kinect depth sensor evaluation for computer vision applications. Department of Engineering, Aarhus University. Denmark. 37 pp. - Technical report ECE-TR-6.
- BARTCZAK, B., SCHILLER, I., BEDER, C., AND KOCH, R. 2008. Integration of a time-of-flight camera into a mixed reality system for handling dynamic scenes, moving viewpoints and occlusions in real-time. In *Proceedings of the 3DPVT Workshop, Atlanta, GA, USA*.
- BLONDE, L., BUCK, M., GALLI, R., NIEM, W., PAKER, Y., SCHMIDT, W., AND THOMAS, G. 1996. A virtual studio for live broadcasting: the mona lisa project. *MultiMedia, IEEE* 3, 2 (Summer), 18–29.
- BRADSKI, G., AND KAEHLER, A. 2008. *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc.
- BURRUS, N., 2011. Kinect calibration. Retrieved from <http://nicolas.burrus.name/index.php/Research/KinectCalibration>.
- FMOD API. <http://www.fmod.org>.
- FUKAYA, T., FUJIKAKE, H., YAMANOUCHI, Y., MITSUMINE, H., YAGI, N., INOUE, S., AND KIKUCHI, H. 2002. An effective interaction tool for performance in the virtual studio-invisible light projection system. *Institution of Electrical Engineers (IEE), IBC 2002*, 389–396.
- FUKUI, K., HAYASHI, M., AND YAMANOUCHI, Y. 1994. A virtual studio system for tv program production. *SMPTE journal* 103, 6, 386–390.
- GIBBS, S., ARAPIS, C., BREITENEDER, C., LALIOTI, V., MOSTAFAWY, S., AND SPEIER, J. 1998. Virtual Studios: An Overview. *IEEE Multimedia* 5, 1, 18–35.
- GRAU, O., PRICE, M. C., AND THOMAS, G. A. 2000. Use of 3d techniques for virtual production. In *Proc. of SPIE, Conference of Videometrics and Optical Methods for 3D Shape Measurement*, vol. 4309, 40–50.
- GRAU, O., PULLEN, T., AND THOMAS, G. 2004. 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* 14, 3 (March), 370–380.
- GRAU, O. 2005. A 3d production pipeline for special effects in tv and film. In *Proceedings of Mirage 2005 Conference, Computer Vision/Computer Graphics Collaboration Techniques and Applications*.
- HAYASHI, M. 1998. Image compositing based on virtual cameras. *IEEE MultiMedia* 5, 1 (Jan), 36–48.
- HERRERA C., D., KANNALA, J., AND HEIKKIL, J. 2012. Joint depth and color camera calibration with distortion correction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34, 10 (Oct), 2058–2064.
- HIGGINS, S. C. 1994. *The moviemaker's workspace: towards a 3D environment for pre-visualization*. PhD thesis, Massachusetts Institute of Technology.
- HOFF III, K. E., 1998. Conversion between opengl depth-buffer z and actual screen-space depth. Retrieved from <http://www.cs.unc.edu/hoff/techrep/openglz.html>.
- ICHIKARI, R., TENMOKU, R., SHIBATA, F., OHSHIMA, T., AND TAMURA, H. 2008. Mixed reality pre-visualization for filmmaking: On-set camera-work authoring and action rehearsals. *Int. J. Virtual Reality* 7, 4, 25–32.
- IDDAN, G. J., AND YAHAV, G. 2001. Three-dimensional imaging in the studio and elsewhere. In *Proc. of SPIE, Conference of Videometrics and Optical Methods for 3D Shape Measurement*, vol. 4298, 48–55.
- KATO, H. 2002. Artoolkit: library for vision-based augmented reality. *IEICE, PRMU* 6, 79–86.
- LALIOTI, V., AND WOOLARD, A. 2003. Mixed reality productions of the future. In *IBC 2003 Conference, International Broadcasting Convention*, 11–15.
- LOOSER, J., GRASSET, R., SEICHTER, H., AND BILLINGHURST, M. 2006. OSGART-A pragmatic approach to MR. Santa Barbara, CA, USA. In *5th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 06): Industrial Workshop*, 22–25.
- OPENKINECT PROJECT. <http://openkinect.org>.
- PETRO, V. 1971. Electronic composite photography, July 27. US Patent 3,595,987.
- SHREINER, D., SELLERS, G., KESSENICH, J. M., AND LICEA-KANE, B. M. 2013. *OpenGL programming guide: The Official guide to learning OpenGL, version 4.3*. Addison-Wesley Professional.
- THELIN, J. 2007. *Foundations of Qt development*, vol. 7. Springer.
- THOMAS, G., JIN, J., NIBLETT, T., AND URQUHART, C. 1997. A versatile camera position measurement system for virtual reality tv production. In *International Broadcasting Convention, 1997*, 284–289.
- THOMAS, G., KOPPETZ, M., AND GRAU, O. 2004. New methods of image capture to support advanced post-production. *SMPTE motion imaging journal* 113, 5-6, 177–184.
- THOMAS, G. 2006. 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*, 31–36.
- VAN DEN BERGH, F., AND LALIOTI, V. 1999. Software chroma keying in an immersive virtual environment. *South African Computer Journal* 24, 50, 155–162.
- VIZ VIRTUAL STUDIO. <http://www.vizrt.com>.
- VSET 3D. <http://www.vset3d.com>.
- WANG, R., AND QIAN, X. 2010. *OpenSceneGraph 3.0: Beginner's guide*. Packt Publishing Ltd.

- WARD, M., AZUMA, R., BENNETT, R., GOTTSCHALK, S., AND FUCHS, H. 1992. A demonstrated optical tracker with scalable work area for head-mounted display systems. In *Proceedings of the 1992 Symposium on Interactive 3D Graphics*, ACM, I3D '92, 43–52.
- WRIGHT, S. 2013. *Digital compositing for film and video*. Taylor & Francis.
- ZHANG, Z. 2012. Microsoft kinect sensor and its effect. *MultiMedia, IEEE* 19, 2 (Feb), 4–10.