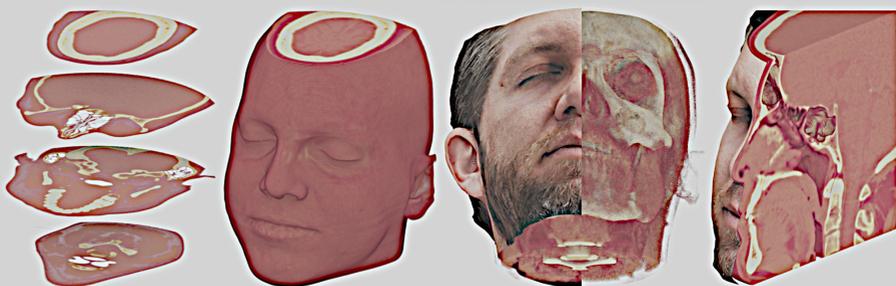


OrtoOnlineMag



Vol. 2

Cicero Moraes ● Everton da Rosa ● Rodrigo Dornelles

CICERO MORAES EVERTON DA ROSA RODRIGO DORNELLES

OrtogOnlineMag

VOLUME 2

SINOP-MT
Cicero André da Costa Moraes
2021

MORAES, Cicero
DA ROSA, Everton
DORNELLES, Rodrigo.
OrtogOnLineMag #2
-- 1. ed. -- Sinop-MT:
CICERO ANDRÉ DA COSTA MORAES, 2021.
114 p.: il.

ISBN 978-65-00-15884-7

DOI 10.6084/m9.figshare.14126951

Sumário

1	Preparo de Modelos Odontológicos para Impressão 3D com o OrtogOnBlender	1
1.1	Introdução	2
1.2	Preparo do Modelo para Impressão 3D	4
1.3	Impressão 3D	10
1.4	Conclusão	12
2	Confeccionando um Guia para Retirada de Cisto Dentário com o OrtogOnBlender	13
2.1	Introdução	14
2.2	Levantamento dos Dados	15
2.3	Confecção do Guia	16
2.4	Impressão 3D e Procedimento Cirúrgico	18
2.5	Conclusão	19
3	Fotogrametria 3D - Desempenho nos Sistemas Operacionais: Windows, Linux e Mac OS X	21
3.1	Descrição Técnica	22
3.2	Fotogrametria	23
3.3	Resultados	24
3.4	Conclusão	31
4	Reconstrução de Tomografias nos Sistemas Operacionais: Windows, Linux e Mac OS X	33
4.1	Descrição Técnica	34
4.2	Resultados	34
4.3	Conclusão	38
5	Sistema de Reconstrução de Tomografia Computadorizada Baseado no Slicer 3D e no DicomToMesh	39
5.1	Apresentação	40
5.2	Como Funciona	42
5.3	Tempo de Reconstrução em Sistemas Operacionais Diferentes	45
5.4	Resultados	46

5.5	Conclusão	49
5.6	Agradecimentos	49
6	Digitalização 3D a partir de Fotografias com o SMVS e o MVE no OrtogOnBlender	51
6.1	Apresentação	52
6.2	Testes e Resultados	54
6.3	Conclusão	59
7	Reconstrução de Voxel Data no OrtogOnBlender	61
7.1	Introdução	62
7.2	Como Funciona - Aspectos Gerais	63
7.3	Como Funciona na Prática	68
7.4	Testes e Resultados	70
7.5	Conclusão	74
7.6	Agradecimentos	74
8	Digitalização 3D de Faces a partir de Vídeos	75
8.1	Apresentação	76
8.2	Captura dos Vídeos	77
8.3	Ferramenta para a Conversão Direta de Vídeo em 3D	78
8.4	Resultados	79
	8.4.1 Protocolo Padrão para Ortogonática	79
	8.4.2 Protocolo Face Duplo 90° + Nariz Duplo 170°	81
	8.4.3 Protocolo Face Triplo 90°	82
8.5	Conclusão	84
9	Digitalização 3D a partir de Vídeo de Drones Disponíveis na Internet	85
9.1	Apresentação	86
9.2	Digitalização da Necrópole de Gizé	87
	9.2.1 Sobre a Necrópole	87
	9.2.2 Escolha do Material	89
	9.2.3 Digitalização da Cena	90
9.3	Outros Exemplos	96
	9.3.1 Cidade de Pedra de Uplistsikhe	96
	9.3.2 Stonhenge	96
9.4	Conclusão	97
10	Conversão de Malhas 3D em Arquivos DICOM com o OrtogOn-Blender	99

10.1	Apresentação	100
10.2	Como Funciona	100
10.2.1	Ferramenta 3D Object to CT-Scan	100
10.2.2	Como Configurar a Malha 3D para a Conversão em DICOM	103
10.3	A Tomografia <i>Fake</i> do Homem de Perak	104
10.4	Conclusão	107
Referências Bibliográficas		109

Preparo de Modelos Odontológicos para Impressão 3D com o OrtoGOnBlender

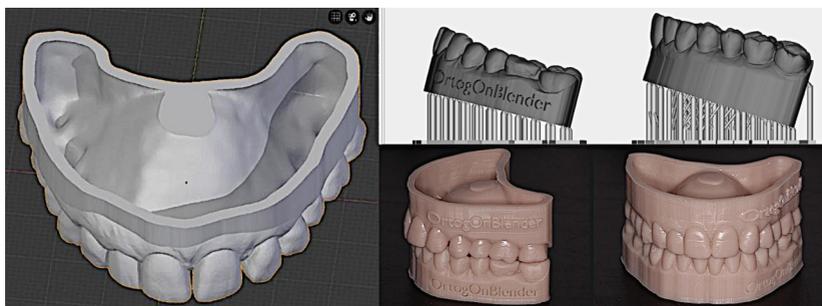
Sérgio Luís Rodrigues Gianvechio

Especialista em Endodontia APCD Araraquara (SP), Especialista em Ortodontia IPESP, Brasília-DF

Cicero Moraes

3D Designer, Arc-Team Brazil, Sinop-MT

DOI: <https://doi.org/10.6084/m9.figshare.13221266>



A odontologia digital vem experimentando uma grande popularização nos últimos anos e um dos principais fatores para o incremento de usuários e entusiastas é a possibilidade do trabalho conjunto com a impressão 3D. Everett M. Rogers no seu livro *Diffusion of Innovations* [Rog10] descreveu de forma gráfica como uma inovação é inte-

grada na sociedade e, de maneira bem sucinta, é possível dizer que quando cerca de 14% dos membros de um determinado grupo adotam uma inovação, ela passará a ser incorporada no seu cotidiano. Sabemos que somente cerca de 4% dos dentistas estão trabalhando atualmente com a odontologia digital (BUNEK, S. 2019) [Bun19] e um dos objetivos principais deste artigo é possibilitar um maior acesso de profissionais a inovações que cruzam computação gráfica com impressão 3D através do script OrtogOnBlender [MdRD20], sendo este completamente baseado em software livre e gratuito.

Atenção: Este material utiliza a seguinte licença Creative Commons: **Atribuição 4.0 Internacional (CC BY 4.0)**.

1.1 Introdução

Em 1987 Charles Hull desenvolveu um formato de arquivo voltado para a impressão 3d chamado de STL (Stereolithography). Trata-se de uma malha de superfície triangulada, ideal para geometria anatômica por conta de sua estrutura de arquivo simples e flexibilidade para corresponder a qualquer contorno desejado (KRAVITZ ET AL. 2008) [eal18]. Em face da sua estrutura relativamente simples o STL foi o formato de maior alcance entre os fluxos digitais em odontologia. A obtenção do arquivo pode se dar através da conversão do arquivo DICOM das tomografias, do escaneamento de modelos dentários ou moldagens e de escaneamentos intra orais. Após a sua obtenção o profissional irá trabalhar no design através de softwares específicos para cada especialidade.

A manufatura aditiva, comumente chamada de impressão 3D, vem sendo impulsionada devido a alguns fatores como a redução de preços, a produção de máquinas compactas ideais para consultórios, bem como a simplicidade de uso. Outro ponto é o aumento na disponibilidade de materiais pois contamos com opções específicas para cada especialidade como por exemplo resinas próprias para modelos dentários, resinas para placas miorelaxantes, resinas para provisórios, resinas para prótese total, resinas calcináveis, entre outras que estão chegando no mercado e finalmente, temos um aumento no acesso aos equipamentos de escaneamento, posto que atualmente

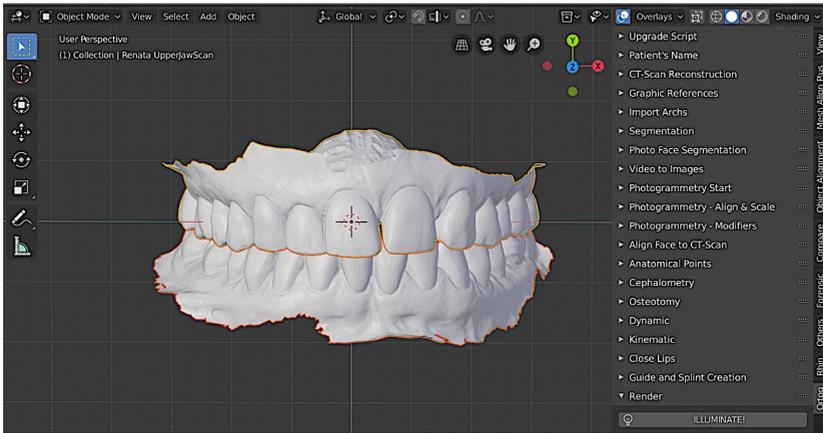
grande parte das clínicas de radiologia odontológica disponibilizam essa tecnologia para seus clientes. Também é importante abordar o aumento de marcas comerciais ao longo dos anos, incremento este que tem gerado concorrência e, naturalmente, preços mais acessíveis. Apesar de toda praticidade na obtenção destes modelos e de sua relativa simplicidade, alguns aspectos técnicos importantes devem ser destacados para que a impressão 3D seja eficiente. Antes de enviar o arquivo para a impressão devemos nos certificar que ele esteja limpo, regular e sem alterações na sua malha como furos ou falhas na conexão entre os triângulos que formam essa malha (RE-NATO PARKESIAN ET AL.) [MGF20].

KRAVITS ET AL., 2018 separaram estas etapas em limpar e reparar a malha (mesh). Limpar a malha envolve a eliminação de superfície estranha ou redundante (duplicada) das estruturas. O software permite que o operador isole uma região específica ou selecione um filtro para limpar automaticamente todo o modelo digital. Este processo inclui uma etapa chamada de “Mesh Decimation”. Alguns *scanners* produzem arquivos STL com um número extraordinariamente grande de triângulos e em algumas aplicações clínicas podemos simplificar esse número sem perder a qualidade da malha, ou seja, sem perder a forma desejada. Por exemplo, um modelo digital com 200.000 triângulos de superfície podem ser simplificados em 100.000 sem qualquer perda de detalhe impactante. A diferença na textura da superfície não seria visível para o olho nu, nem seria notado na saída de uma impressora 3D. Reparar a malha envolve usar o mesmo software para preencher pequenos vazios e reorientar triângulos invertidos. Reparar a malha é análogo a remendar espaços vazios em um modelo de gesso com resina. Uma terceira etapa, chamada retopologia, pode ser adicionada a este processo, caso o modelo original apresente muita irregularidade na superfície. No *OrtogOnBlender* é chamada de **Prepare to Printing** e consiste em fazer uma organização das estruturas de vértices/arestas/faces de um modelo 3D, ou seja, é o quanto você consegue organizar a malha para que a impressão seja mais eficiente possível.

Estando a malha apta para a impressão 3D passamos a um processo que chamaremos de “Preparo do Modelo para Impressão 3D”. Este processo consiste em deixar o modelo oco, para economizarmos resina e acelerar a impressão e ao mesmo tempo não permitir que a peça perca resistência. Para isso deixaremos as margens com uma

espessura (*offset*) entre 2 a 2,5mm. Finalmente colocaremos o nome da clínica ou do paciente e se for usado para fins comerciais ainda temos a possibilidade de inserir a logo de quem está produzindo os modelos ou para quem eles serão produzidos.

1.2 Preparo do Modelo para Impressão 3D



Figural: Importação dos modelos.

Ao abrirmos o software executaremos as etapas básicas de adicionar o nome do paciente e salvar para seguirmos no fluxo. A seguir iremos para a aba **Import Archs** onde temos a opção de importar os arquivos STLs. Neste momento quando importamos os arcos superior e inferior de um mesmo escaneamento eles já aparecerão em oclusão (Fig. 1 - atente-se ao gráfico dos eixos).

Agora é a hora de orientarmos os modelos para que as etapas futuras não sejam comprometidas. Para isso deixaremos a orientação transversal dos modelos no eixo X e a orientação anteroposterior no eixo Y (Fig. 2 - note a correção em relação aos eixos).

Estando os modelos orientados, iremos trabalhar com um deles por vez. Neste trabalho optamos por fazer o arco superior primeiro mas nada o impede de usar o fluxo para o arco que você desejar iniciar. Ocultaremos o arco inferior e iremos para a aba **Others** (Fig. 3).

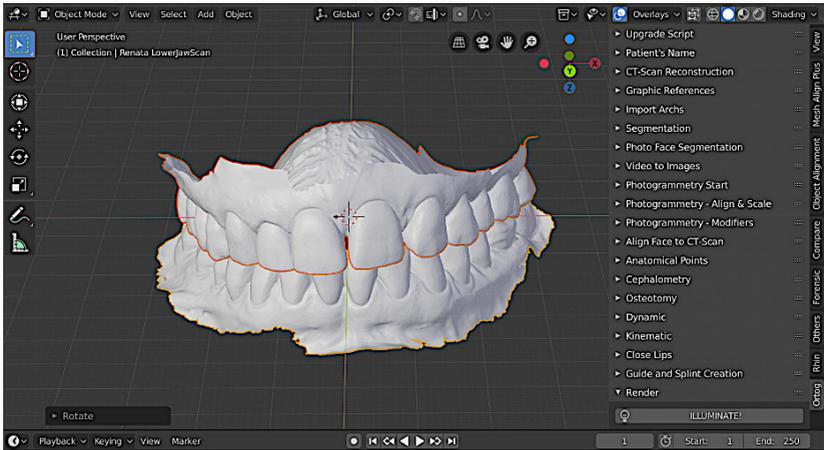


Figura2: Posicionamento dos modelos e relação aos eixos X e Y.

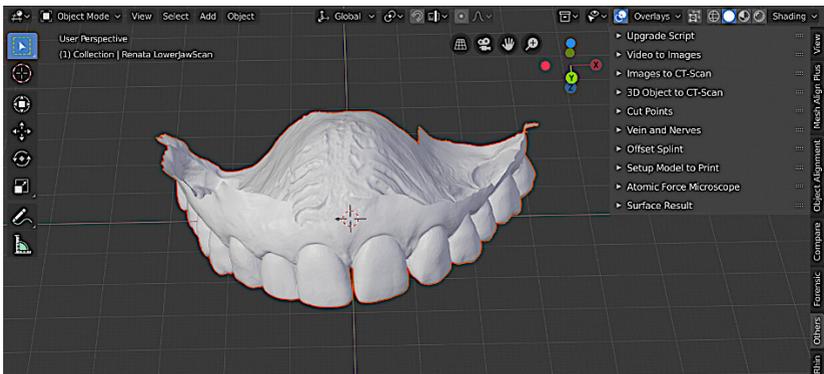


Figura3: Cena com o modelo inferior oculto.

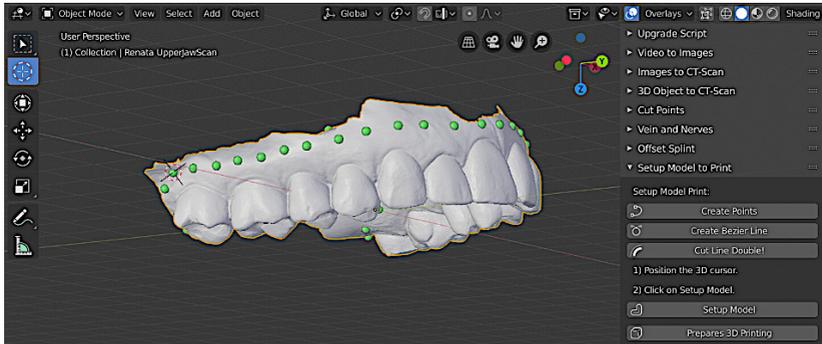


Figura4: Criando pontos de corte.

Agora teremos um novo menu e devemos ativar o **Setup Model to Print**. Neste menu encontraremos a sequência que já está programada para a execução do preparo do modelo. Ela se iniciará com a contorno dos limites do modelo e para isso ativaremos a opção **Create Points**. Com o lado esquerdo do mouse iremos clicando e desenhando nossa margem e ao terminá-la deveremos pressionar em ESC para finalizar a delimitação (Fig. 4).

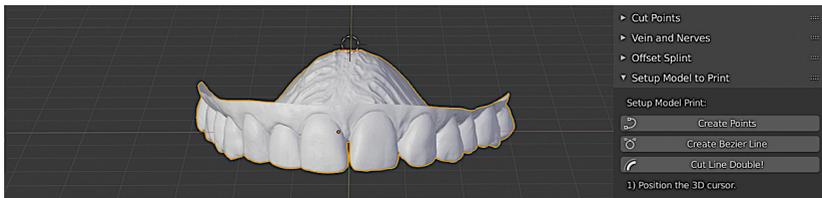


Figura5: Modelo segmentado.

Neste momento podemos fazer uma visualização orbitando o modelo e seguirmos para **Create Bezier Line**. Uma linha de corte será criada e você poderá novamente orbitar o modelo e conferir se o contorno que você desenhou está adequado para o preparo do mesmo. Estando tudo de acordo seguimos para o **Cut Line Double**. Agora a espessura do corte terá o diâmetro das esferas usadas para delinear o contorno do modelo. Feita a operação selecionamos a parte do modelo onde estão os dentes e então faremos a operação de inversão de seleção (Ctrl+I) e deletamos a parte que ficou fora da seleção.

De acordo com a sequência estabelecida pelo programa iremos agora

posicionar o cursor para indicarmos a altura e o *offset* (Fig. 5). A largura das margens do nosso modelo já foi descrita anteriormente como 2 a 2,5mm.

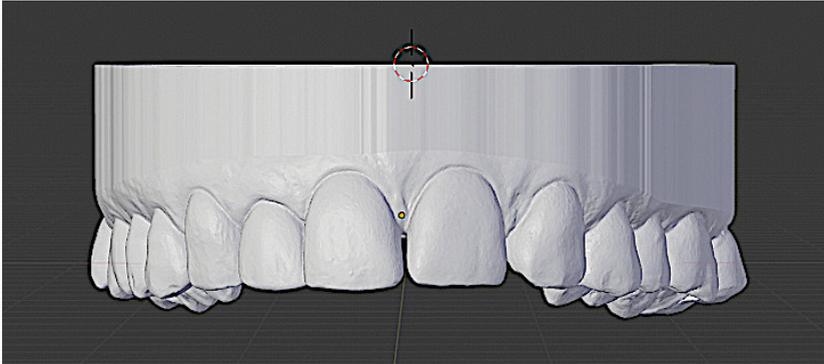


Figura6: Modelo preparado.

Para isso posicionaremos a peça voltando nossa vista à face vestibular dos incisivos centrais com a vista ortográfica. Posicionado o cursor ativamos o **Setup Model** e a programação do software finalizam o preparo da peça (Fig. 6).

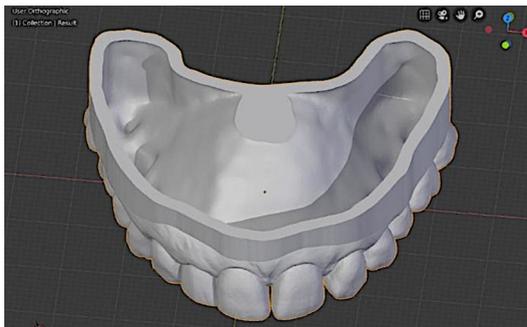


Figura7: Visualização dos detalhes superiores do modelo.

Finalizando as etapas do preparo (Fig. 7) partiremos para o arco inferior. A mesma sequência de trabalho deverá ser feita, lembrando apenas que o posicionamento do cursor deverá ficar para a face inferior do modelo para obtermos a extrusão e o *offset* sem interferir

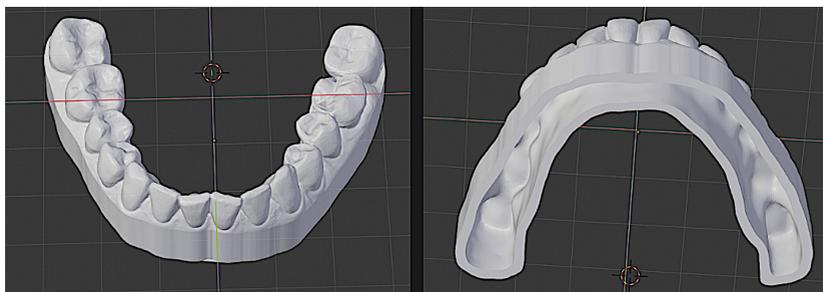


Figura8: Preparo efetuado na arcada inferior.

na região onde estarão os elementos dentários.

Realizada as etapas no arcos superior e inferior agora já contamos com o nosso modelo preparado, ou seja, já corrigimos as irregularidades das bordas, reparamos possíveis falhas na malha e podemos proceder com a etapa da topologia para melhorar a distribuição dos polígonos na malha que será impressa.

Entendemos que quanto mais fiel o modelo for em relação ao objeto escaneado a depender do tipo de trabalho a ser realizado devemos organizar os polígonos de modo a não alterar a sua forma, mesmo que de maneira mínima, pois vários procedimentos odontológicos são de extrema precisão. Também entendemos que modelos para diagnóstico poderão ter uma topologia mais simplificada que um modelo para a confecção de alinhadores ou para a confecção de provisorios dentários.



Figura9: Configuração do texto.

Como uma última etapa antes da exportação dos modelos procederemos com a sua identificação na seção **Project Text and Logo** (Fig. 9 à direita).

Nesta aba dispomos de uma sequência de trabalho que deverá ser feita a fim de identificar o modelo. Iniciamos selecionando o modelo a ser nomeado e clicamos em **Create Text**. Será criado um objeto de texto com a palavra digitada em Text. Para movimentarmos utilizaremos o comando *grab* (G) e o posicionamos de acordo com a nossa vista (Fig. 9 ao centro e à esquerda).

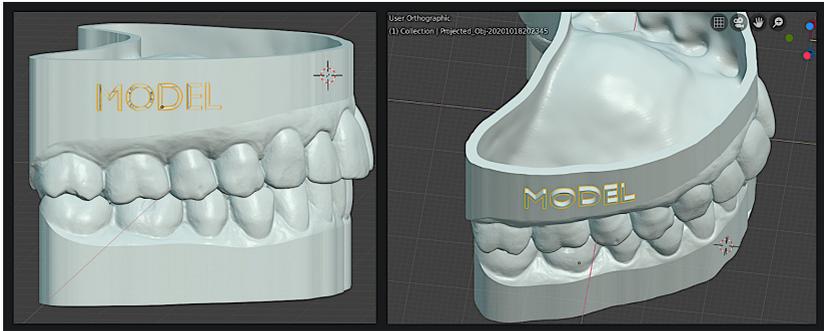


Figura10: Texto recebendo tratamento de espessura.

Seguindo a sequência já estabelecida pelo software projetaremos o texto sobre o modelo usando a ferramenta **Project to Mesh** e após isso clicamos em **Object Volume** para dar espessura ao objeto selecionado (Fig. 10).

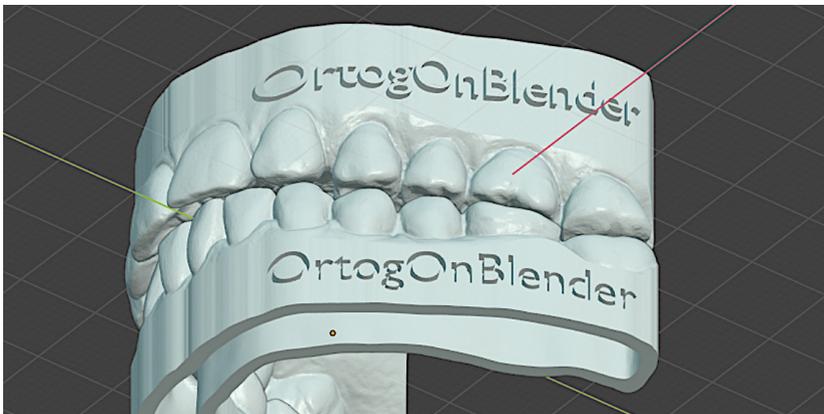


Figura11: Texto com booleanas Union e Difference.

Podemos ainda melhorar ou realçar o texto procedendo com uma operação para aumentar a espessura deste. Para tal ativamos o **Edit Mode** (Tab), mudamos o *set list* da operação de *transform* para **Normal**, ou seja, faremos um deslocamento em noventa graus do texto e agora com esta camada mais espessa nos é permitido fundi-lo ao modelo para realizar as operações booleanas, onde poderemos ter o texto em alto relevo (Union) ou em negativo (Difference) nos modelos. Desta forma finalizamos o preparo do nosso modelo com a exportação em STL de modo a deixá-lo apto a ser impresso em 3D.

1.3 Impressão 3D

O CAM (computer-aided manufacturing), grosso modo, materialização dos projetos, pode se dar de duas formas: através da fresagem de um bloco ou através da manufatura aditiva. A manufatura aditiva, também conhecida como impressão 3d tem sido a tecnologia de escolha para estes processos devido a rapidez e economia financeira em relação à fresagem. Na década de 1980, Charles Hull patenteou a tecnologia SLA (Estereolitografia) e a partir de então a impressão 3D vem evoluindo. Nos anos 2000 ocorreu o estabelecimento da impressão 3d como uma verdadeira indústria de manufatura e esta evolução a levou às áreas de saúde.

Basicamente em odontologia utilizamos 3 tipos de tecnologia para impressão 3D:

- FDM (impressoras de filamento);
- SLA (Laser Stereolitography);
- DLP (Digital Light Processing).

Cada uma tem suas aplicações e conseqüentemente seus prós e contras, e de maneira bem objetiva podemos considerar as de tecnologia DLP e as SLA adequadas para o uso em odontologia devido a qualidade de processos, velocidade e repetibilidade. Para nosso trabalho optamos pela impressão em uma maquina DLP, de marca MOONRAY. A seguir descreveremos o processo de impressão.

A primeira etapa deste processo é a importação dos modelos seguido da orientação destes na prancha de impressão. Seguimos para isso a

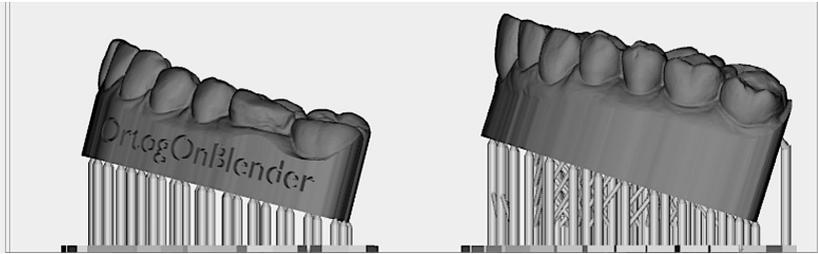


Figura12: Modelos preparados para impressão 3D, com suporte.

orientação das empresas que especificam que para uma menor superfície de descolamento que irá refletir em maior precisão de impressão e maior repetibilidade orientamos o nosso modelo de maneira inclinada na plataforma de impressão. Como desvantagem deste posicionamento, quando imprimimos em grande quantidade, temos que repetir o processo várias vezes pois os modelos ocupam uma grande área na prancha de impressão. Para resolver esse problema quando imprimimos um maior número de modelos os orientamos na vertical. Outro fator importante é a colocação dos suportes, que darão estabilidade à peça pois no processo de impressão precisamos manter a geometria da estrutura intacta e os suportes nos auxiliam nessa tarefa (Fig. 12).

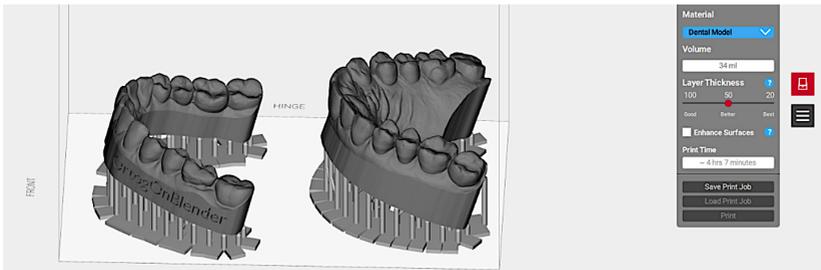


Figura13: Interface do software de preparo de impressão 3D.

A próxima etapa é a definição do material e da resolução da peça. Hoje temos para cada aplicação um determinado tipo de resina e para os modelos a resina *die and model* é a ideal. Neste trabalho usamos a resina SprintRay Die and Model, com o parâmetro de eixo Z em 50 microns. O software da impressora nos permite regular entre 100 , 50 e 25 microns. Para cada aplicação teremos uma regulação

neste parâmetro. No próprio *display* ainda contamos com a informação do tempo de impressão e do volume de resina que será gasto do processo (Fig. 13).



Figural4: Modelos impressos em 3D.

Seguindo com os trabalhos, acionamos a impressora e no tempo estabelecido retiraremos os modelos da prancha de impressão e partimos para o pós processamento. Este processo consiste na limpeza do modelo com álcool isopropílico para remover os resíduos de resina que ficam no modelo e depois fazer a pós cura para que as camadas superficiais da peça estejam totalmente polimerizadas. Finalizando estes processos o nosso modelo estará pronto para uso, seja para fins de diagnóstico ou para a produção de dispositivos (Fig. 14).

1.4 Conclusão

O OrtogOnBlender se mostrou uma ferramenta apta para a geração de modelos de arcadas dentárias ocas, com a opção de projetar textos ou logos em sua superfície, resultando em uma impressão 3D mais econômica e com visual personalizado.

Confeccionando um Guia para Retirada de Cisto Dentário com o OrtogOnBlender

Cicero Moraes

3D Designer, Arc-Team Brazil, Sinop-MT

Roosevelt Macedo

Cirurgião BMF, Clínica Statto, Sinop-MT

Gabriela Alonson

Cirurgiã BMF, Clínica Santa Izabel, Sinop-MT

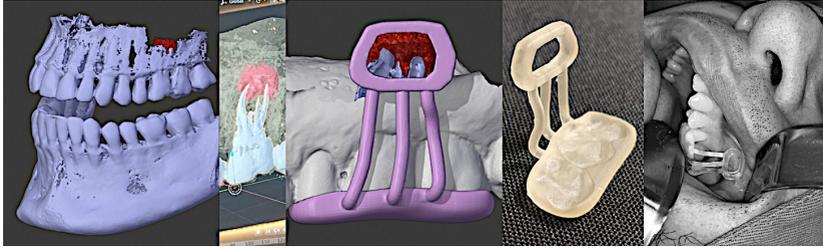
Carlos Augusto Abascal Shiguihara

Médico Veterinário, Clínica Santa Izabel, Sinop-MT

Everton da Rosa *Cirurgião BMF, Hospital de Base, Brasília-DF*

DOI: <https://doi.org/10.6084/m9.figshare.13256816>

O presente capítulo tem por objetivo apresentar um caso onde a ferramenta OrtogOnBlender permitiu a criação de um guia cirúrgico para extração de um cisto dentário, ampliando assim a precisão do procedimento bem como o incremento da sua segurança.



Atenção: Este material utiliza a seguinte licença Creative Commons: **Atribuição 4.0 Internacional (CC BY 4.0)**.

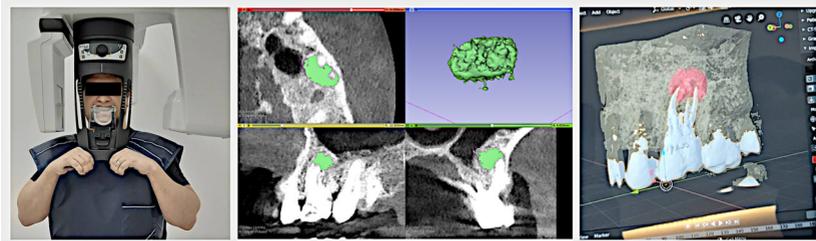
2.1 Introdução

O uso de novas tecnologias em odontologia tem auxiliado os cirurgiões-dentistas na obtenção de resultados melhores e mais confiáveis nos procedimentos desta especialidade. O OrtoGOnBlender (OOB) [MdRD20] tem se destacado como uma excelente ferramenta no ensino do planejamento de cirurgias ortognáticas, traumatologia maxilofacial, arqueologia entre outros. Neste capítulo será demonstrado o *workflow* para a utilização de um guia cirúrgico obtido por impressão 3D auxiliando o cirurgião na localização anatômica precisa de uma lesão dentária periapical (cisto radicular) em um dente molar superior, contribuindo para obtenção de um melhor resultado.

O cisto radicular é o mais comum dos cistos odontogênicos [Tom02]. Segundo Lindhe et al., 1999 sua prevalência corresponde à cerca de 60% dos cistos da maxila e da mandíbula [LKL99]. A terceira e a quarta décadas de vida apresentam uma destacada prevalência [dFReS00]. O cisto radicular consiste de uma cavidade patológica forrada por epitélio originário dos restos epiteliais de Malassez com líquido em seu interior, envolvendo o ápice de um dente presente na cavidade bucal e está relacionado com infecção da câmara pulpar e do canal radicular provocada pela cárie dentária [SMB87] [dAdA94]. Este cisto origina-se a partir de um granuloma periapical com epitélio preexistente, cronicamente inflamado, intra-ósseo, no ápice de um dente sem vitalidade [RS00] [SMB87]. Os cistos dos maxilares

geralmente são tratados cirurgicamente pelas técnicas de enucleação (remoção total da lesão) e/ou marsupialização (remoção parcial da lesão), realizando-se o tratamento endodôntico (tratamento de canal) prévio [LL98].

2.2 Levantamento dos Dados



Figural: Tomografia Cone Beam da região de interesse, segmentação do cisto e dos dentes.

Foi efetuada uma tomografia computadorizada Cone Beam (Fig. 1 à esquerda) na região de interesse que compreendia principalmente os dentes 25, 26 e 27. Uma vez digitalizada, a tomografia foi importada no software Slicer v. 4.81 [Sli20]. A segmentação do cisto foi efetuada pela ferramenta **GrowCutEffect** [Doc20a] presente no módulo **Editor**, bastando fornecer ao software informações sobre a área a ser segmentada (cisto) e a área a ser excluída (ossos e demais tecidos), traçando duas linhas com cores distintas nas regiões pretendidas (de reconstrução e de exclusão). O processo de pintura cria um *label*, ou seja um rótulo, uma identificação das áreas de interesse em cada *slice* (fatia) onde o cisto se fará presente, mas para converter essa sequência de pinturas em um volume é necessário recorrer ao módulo **Model Maker** [Doc20b], indicar o *label* e reconstruí-lo em um objeto 3D (Fig. 1 ao centro). No entanto, não basta reconstruir apenas o cisto, é necessário digitalizar mais estruturas, como as regiões mais densas dos dentes e canais (Fig. 1 à direita), bem como os ossos em geral, de modo a fornecer os dados necessários para a criação do guia de perfuração, que será abordado mais a frente.

A malha reconstruída a partir da tomografia computadorizada fornece muitas informações importantes, principalmente as internas

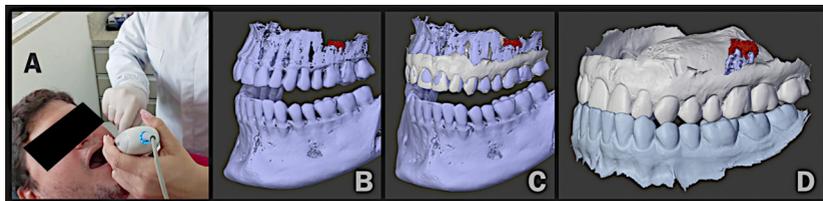


Figura2: Digitalização das arcadas e alinhamento com os modelos reconstruídos a partir da tomografia.

como as raízes dos dentes e o próprio cisto (Fig. 2, B). No entanto, é comum que na reconstrução dos dentes a malha apresente problemas como os artefatos ou outras irregularidades por conta do contraste na escala hounsfield. Para contornar esse problema e fornecer uma boa superfície para a criação do guia, foi efetuada uma digitalização 3D intraoral com luz estruturada nas arcadas do paciente (Fig. 2, A). A digitalização intraoral foi alinhada ao modelo da fotogrametria dentro do OrtogOnBlender (Fig. 2, C) resultando em um modelo mais simples e limpo, mas com os dados da raiz do dente 25 junto com o cisto, ambos segmentados a partir da tomografia (Fig. 2, D).

2.3 Confeção do Guia

A criação do guia apresentado neste capítulo consiste na formação de uma estrutura que permita a perfuração da região do cisto ao mesmo tempo que que se encaixe nos dentes, permitindo ao cirurgião que acesse a região de interesse de modo mais preciso e seguro.

Em linhas gerais é necessário identificar espacialmente onde se encontra o cisto (Fig. 3, A). Em seguida criar a estrutura que servirá para a perfuração do osso, de modo a acessar o cisto. É importante que a os canto do acesso forneçam um alinhamento que auxilie o cirurgião a mover o equipamento de modo objetivo (Fig. 3, B). A próxima estrutura a ser criada é aquela que encaixará o guia nos dentes e para que a mesma seja bem estável é imprescindível que se encaixe em mais de um dente, neste caso nos três, de modo a ficar bem fixa no momento do acesso ao cisto (Fig. 3, C). A etapa final é criar as hastes que unirão a estrutura de acesso à estrutura de fixação (Fig. 3, D).

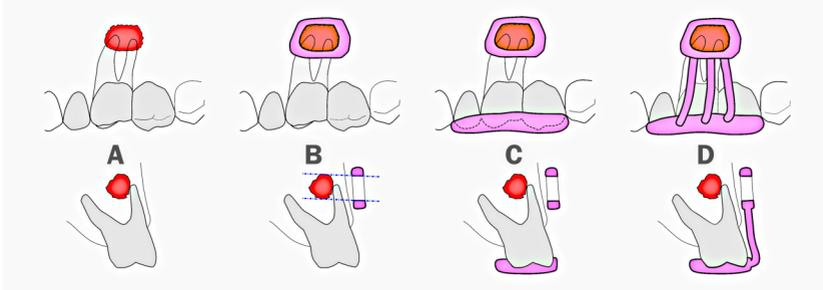


Figura3: Passo-a-passo esquemático da criação do guia em 3D.

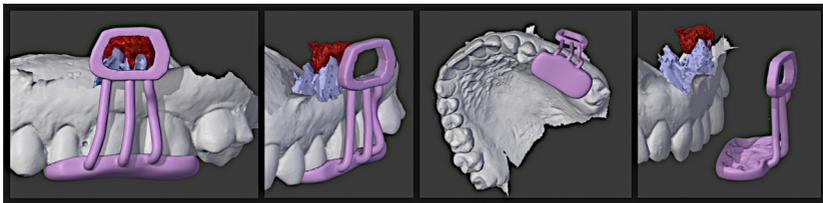


Figura4: Guia cirúrgico final.

O guia gerado permite um encaixe firme nos dentes e uma visão clara da região que será perfurada. A base da peça é lisa e o encaixe dos dentes é possível graças a subtração da intersecção da peça com os dentes através de cálculos booleanos (Fig. 4).

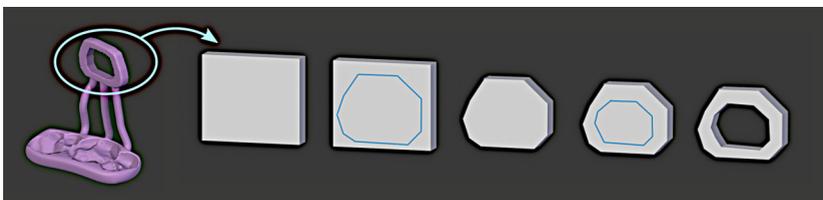


Figura5: Criação da estrutura de perfuração.

Cada parte do guia foi modelada de uma maneira. A estrutura de perfuração do osso (Fig. 5 da esquerda para a direita) foi modelada a partir de uma primitiva *box* achatada em um dos eixos. A estrutura foi posicionada em frente ao cisto e cortada de fora a fora por projeção de linhas com as ferramentas **Subtract IN** e **Subtract OUT**, até formar o volume final.

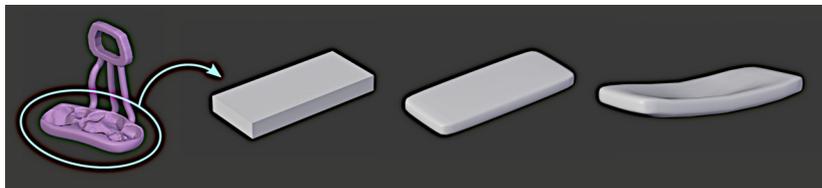


Figura6: Criação da estrutura de encaixe.

A estrutura de encaixe (Fig. 6 da esquerda para a direita) também foi modelada a partir de uma *box* deformada, mas no lugar do fatiamento fois ajustada com as ferramentas de escultura digital disponíveis no software de modelagem.

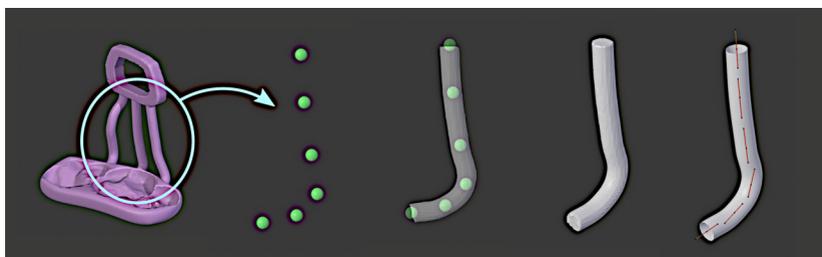


Figura7: Criação das hastes do guia.

A criação das hastes foi efetuada com a ferramenta **Draw Line Tube** (Fig. 7 da esquerda para a direita) pela inserção de pontos que se converteram em um tubo redimensionável e editável, de modo que o usuário possa alterar o raio do mesmo, bem como ajustar os pontos da linha bézier.

2.4 Impressão 3D e Procedimento Cirúrgico

O guia digital passa por uma preparação e é impresso finalmente em 3D, utilizando para tal uma impressora de resina. A peça resultante do processo (Fig. 8, A) é então testada no paciente antes da cirurgia para que haja tempo de efetuar eventuais correções (Fig. 8, B). Uma vez que tudo se mostre funcional, a peça recebe um tratamento de desinfecção e é preparada para a cirurgia. O procedimento foi efetuado e o guia cumpriu com o seu propósito de fornecer ao cirurgião

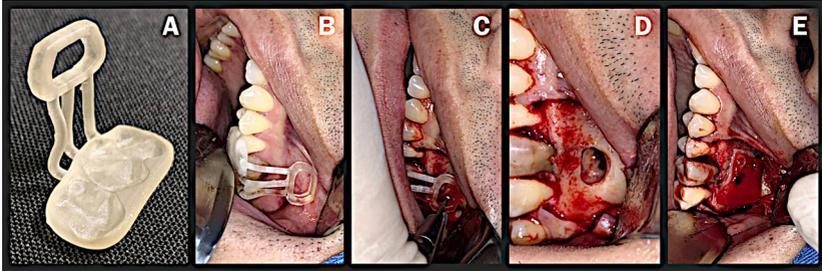


Figura8: Impressão do guia e procedimento cirúrgico.

informações espaciais para o acesso ao cisto (Fig. 8, C a D).

2.5 Conclusão

O OrtogOnBlender mostrou-se uma ferramenta apta para a criação de guia cirúrgico para a retirada de cisto dentário.

Fotogrametria 3D - Desempenho nos Sistemas Operacionais: Windows, Linux e Mac OS X

Cicero Moraes

3D Designer, Arc-Team Brazil, Sinop-MT

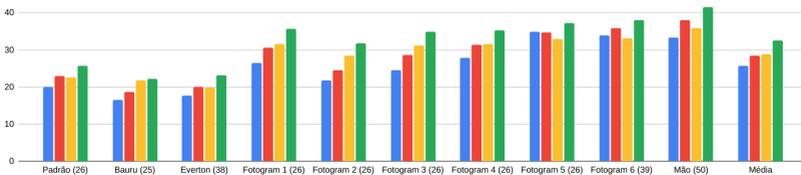
Rodrigo Dornelles

Cirurgião Plástico, Núcleo de Plástica Avançada - NPA, São Paulo-SP

Everton da Rosa

Cirurgião BME, Hospital de Base, Brasília-DF

DOI: <https://doi.org/10.6084/m9.figshare.13456442>



O mercado de sistemas operacionais reflete as necessidades, apreciações e até restrições pecuniárias dos seus utilizadores. Graças às ferramentas de desenvolvimento coletivo é muito comum que todas as plataformas tenham à sua disposição uma ampla gama de desenvolvedores focados na melhora do desempenho ao usuário. Com a popularização de compiladores multiplataformas é muito comum

que aplicações ofereçam opções para os três sistemas operacionais de computadores mais populares: Windows, Linux e Mac OS X. O presente capítulo explora o desempenho da fotogrametria 3D nos três sistemas rodando na mesma máquina, um notebook munido de um processador i7, 20 GB de memória RAM e com armazenamento SSD.

Atenção: Este material utiliza a seguinte licença Creative Commons: **Atribuição 4.0 Internacional (CC BY 4.0)**.

3.1 Descrição Técnica

Buscando a melhor abordagem possível para comparar o desempenho de uma mesma ferramenta de fotogrametria em três sistemas operacionais diferentes, os autores optaram por utilizar um mesmo notebook com os três sistemas instalados no mesmo dispositivo de armazenamento, buscando assim fornecer condições semelhantes para o cálculo do desempenho.

Notebook:

- Marca: Lenovo;
- Modelo: Ideapad S145;
- Processador: Intel Core i7-8565U 1.80GHz;
- Memória RAM: 20GB;
- GPU 1: UHD Graphics 620;
- GPU 2: NVIDIA GeForce MX110 2GB;
- Armazenamento 1: 480 GB SDSSDA SanDisk;
- Armazenamento 2: 512 GB SSD M.2 NVMe Intel;
- Armazenamento 3: 1.0 TB 5400 RPM WD10JPCX WD Blue.

Sistemas operacionais:

- Windows 10 Home;
- Windows Subsystem for Linux (WSL) com binários do Ubuntu 18.04;

- Ubuntu 20.04;
- Mac OS X 10.15 Catalina.

O Windows 10 foi baixado diretamente do site da Microsoft que fornece ISOs do sistema operacional para download [Mic] e o Windows Subsystem for Linux foi instalado diretamente pela Microsoft Store [MdRD20b].

O Linux utilizado foi uma versão mais nova do Linux 3DCS [MdRD20a] que tem como base a versão Netboot do Ubuntu 20.04 disponível para download no site do distribuidor [Ubu].

Para rodar o Mac OS X foi necessário uma prática conhecida como hackintosh [Gog] que permite a instalação do sistema operacional em máquinas não fornecidas pelo fabricante Apple. O uso do sistema foi feito em caráter excepcional para a testagem das ferramentas descritas neste capítulo. Ainda que possa configurar uma quebra de contrato, a instalação de um sistema operacional em uma máquina não fornecida pelo fabricante não representa crime no Brasil, posto que obrigar o uso de uma máquina específica para rodar determinado sistema operacional pode configurar venda casada, uma prática ilegal no país segundo o inciso I do Artigo 39 do Código de Defesa do Consumidor. [dR]

Aviso: Ainda que haja muita discussão sobre os limites legais de tal prática, os autores não recomendam o uso do hackintosh para ferramentas de produção, tampouco a compra de máquinas prontas com essas configurações para uso em produção. A situação abordada neste capítulo é puramente didática, experimental e sem fins lucrativos.

3.2 Fotogrametria

O processo de fotogrametria consiste em indicar um diretório onde há uma série de fotografias que seguem um protocolo de captura pré definido [MdRD20d] [MdRD20c]. Uma vez indicado o local das fotos o OpenMVG [MMPM16] encontra a localização tridimensional da câmera no momento em que a foto foi efetuada e cria uma nu-

vem esparsa de pontos correspondente ao objeto digitalizado. Em seguida os dados de posicionamento das câmeras são exportados ao OpenMVS [Cer20] que utiliza as informações para gerar uma nuvem densa de pontos. A nuvem densa de pontos é então reconstruída em uma malha 3D e a malha recebe os dados de texturização (cores) advindas das fotografias. Ao final do processo é gerado um modelo tridimensional devidamente colorizado.

Por padrão o OrtogOnBlender copia as imagens para um diretório temporário de modo a não alterar as fotografias originais. Se as imagens têm alguma incompatibilidade com o OpenMVG há uma correção automática, como por exemplo, a conversão do formato Heic para Jpg ou a inserção de dados de abertura da câmera. A cópia e eventual correção são ignoradas pelo cronômetro que estipula o Fator de Tempo por Fotografia (FTF) [MdRD20e], utilizado para a mensuração dos dados que leva em consideração apenas o tempo de processamento efetuado pelo OpenMVG e pelo OpenMVS.

Foram elencadas 10 sequências de fotografias (com o número de capturas entre parênteses): Padrão (26), Bauru (25), Everton (38), Fotogram 1 (26), Fotogram 2 (26), Fotogram 3 (26), Fotogram 4 (26), Fotogram 5 (26), Fotogram 6 (39) e Mão (50).

3.3 Resultados

Os dados foram organizados em uma tabela e ao final extraíu-se uma média de todos os tempos (Fig. 1). Para rodar os testes os autores optaram por fazê-los todos no SDSSDA SanDisk de 480 GB. O sistema operacional em questão foi instalado, os testes rodados, o dispositivo era então formatado e outro sistema instalado, repetindo o processo.

O resultado utilizando a mesma abordagem com todos os sistemas operacionais diferentes, colocou o Mac OS X em último lugar (Fig. 2) com pouco menos de 4 segundos de diferença do Windows e do Linux que vieram praticamente empatados na terceira e segunda posições. Curiosamente, a primeira posição foi ocupada pelo WSL. Embora muitos usuários o definam como um Linux que roda dentro do Windows, o que ele faz na verdade é traduzir os binários do Linux para que os mesmos possam ser executados diretamente pelo Win-

TEMPO (em segundos) SSD SATA					
Sequência	WSL	Windows	Linux	Mac	
Padrão (26)		20	22,92	22,58	25,75
Bauru (25)	16,61	18,67		21,88	22,21
Everton (38)	17,66	19,95		19,84	23,2
Fotogram 1 (26)	26,46	30,69		31,61	35,61
Fotogram 2 (26)	21,81	24,46		28,38	31,77
Fotogram 3 (26)	24,58	28,69		31,11	34,96
Fotogram 4 (26)	27,81	31,46		31,58	35,3
Fotogram 5 (26)	34,96	34,65		32,88	37,27
Fotogram 6 (39)	34	35,84		33,08	38,02
Mão (50)	33,34	38		35,92	41,48
Média	25,72	28,53		28,89	32,56

Figura1: Tabela com os resultados do fator de Fator de Tempo por Foto (FTF) em segundos.

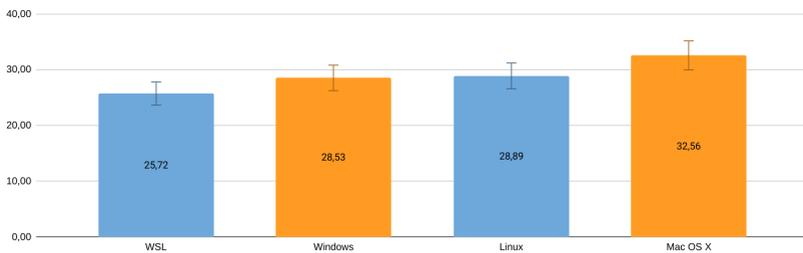


Figura2: Gráfico com os resultados da média geral do Fator de Tempo por Foto (FTF) em cada sistema operacional.

Padrão (26). O resultado do teste foi um gráfico, até o momento com 50 resultados, envolvendo diversos processadores e sistemas operacionais diferentes (Fig. 4).

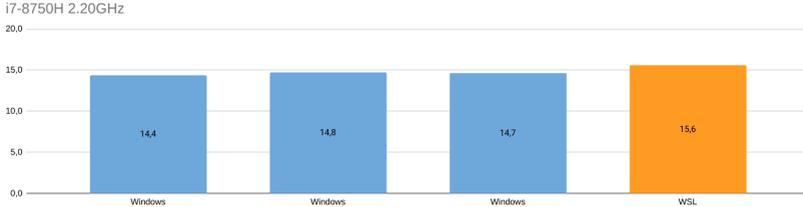


Figura5: Gráfico os resultados do FTF com um mesmo processador em computadores diferentes

Analisando os dados do gráfico de processadores e FTF é possível extrair uma série de resultados que envolvem o modelo i7-8750H de 2.20GHz em vários computadores diferentes. Neste caso o WSL também ficou em último, ainda que com uma pequena margem de diferença (Fig. 5).

Isso evidencia que o WSL não é efetivamente mais rápido em todas as situações, mas quanto ao Mac OS X, seria ele um sistema menos robusto que os demais? Não necessariamente.

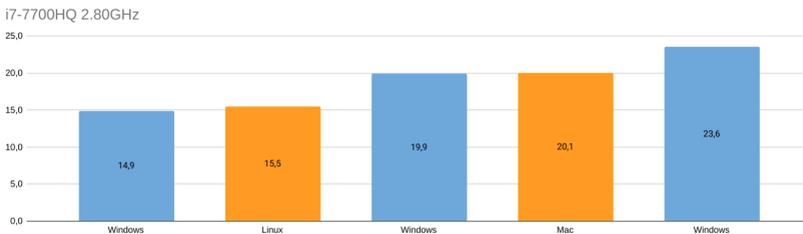


Figura6: Gráfico com os resultados o FTF em processadores e sistemas operacionais diferentes.

Foi possível isolar 5 resultados em computadores diferentes envolvendo o processador i7-7700HQ de 2.80GHz. Neste cenário um computador com o Windows está em primeiro e outro computador com o mesmo sistema está em último (Fig. 6). O Mac OS X está entre dois outros com um resultado significativamente menor que o último e

praticamente o mesmo do terceiro, ou seja, não é o menos robusto se forem levados em consideração todos os testes efetuados, posto que a média do Windows (19,73) ficou muito compatível com o resultado do Mac.

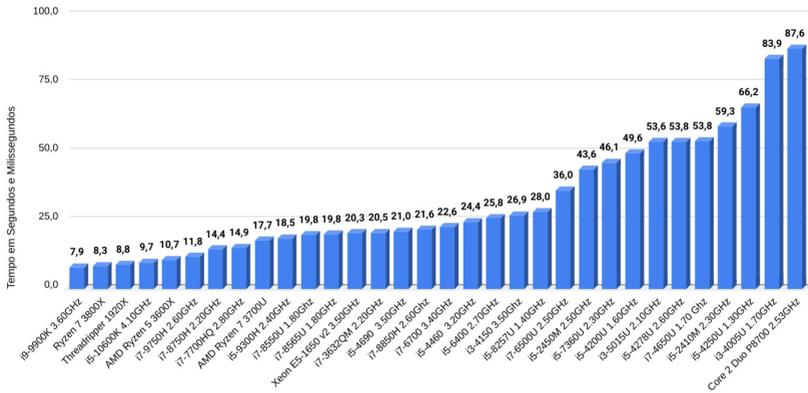


Figura7: Gráfico com os melhores resultados do FTF em processadores e sistemas operacionais diferentes.

Um outro gráfico foi gerado com os melhores resultados, reduzindo o número de IDs (processadores) de 50 para 32 (Fig. 7). Curiosamente o valor de 19,8 apresentado pelo processador i7-8565U de 1.80 GHz foi alcançado há alguns meses antes da edição deste capítulo, justamente pelo notebook aqui utilizado e rodando o sistema operacional Linux em uma versão anterior, a 18.04 (atualmente trata-se da 20.04). Em uma série de testes efetuados para o levantamento da margem de erro, o resultado do FTF foi de 22,58 até 23,25 segundos, ou seja, em um mesmo computador houve uma variação de 19,8 até 23,25 segundos por foto, o que dá uma margem de erro arredondada de 8% para mais e para menos.

Qualquer detalhe como uma compilação com parâmetros específicos, ou mesmo uma biblioteca do sistema que é chamada para executar determinada atividade no decorrer do processo, pode influenciar e alterar o tempo total da fotogrametria.

Uma situação que ajuda na compreensão do que acontece em processos complexos como o abordado aqui é outro teste feito durante os estudos que originaram este capítulo. Na ocasião foi levantado o

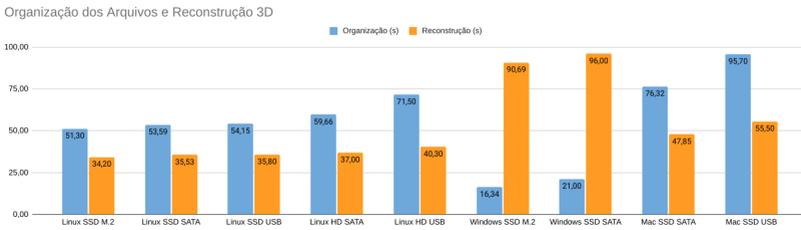


Figura8: Gráfico com o tempo levado para organizar e reconstruir a mesma tomografia computadorizada.

tempo necessário para a organização dos arquivos (Fig. 8 em azul) e também o tempo necessário para a reconstrução da tomografia em uma malha 3D (Fig. 8 em laranja).

É possível perceber que o Windows leva grande vantagem no processo de organização dos arquivos que consiste em identificar os dados do paciente e organizá-los em diretórios diferentes. No entanto, quando a demanda recai no processamento da reconstrução o Windows é claramente inferior aos demais sistemas operacionais.

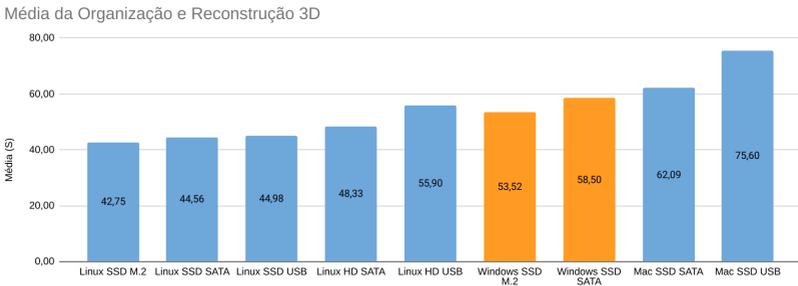


Figura9: Gráfico com o tempo médio dos processo conjuntos de organização e reconstrução 3D.

Contudo, quando se é levado em consideração o processo como um todo e a média é gerada, as diferenças podem se diluir a ponto de dar vantagem a um sistema ou outro, como no caso do Linux em primeiro seguido pelo Windows em segundo, que mesmo tendo um processamento de reconstrução inferior ao Mac OS X deixou-o em último lugar (Fig. 9).

Seria essa uma prova cabal de que o Mac OS X é de fato menos robusto que os demais? Não necessariamente.

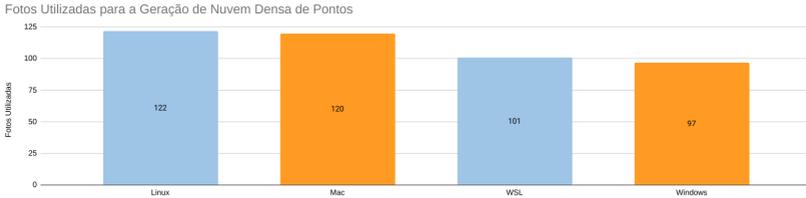


Figura10: Gráfico com o número de fotos utilizadas para a geração da nuvem densa de pontos.

Dentre os testes efetuados no processo, um deles foi descartado da contagem final por conta de uma diferença significativa de fotografias utilizadas para a geração da nuvem densa de pontos, etapa essencial para a reconstrução da malha tridimensional (3D mesh). Eram no total 122 fotografias de uma face humana, o Linux utilizou as 122 para os cálculos, seguido do Mac com 120, o que não seria um problema. A discrepância significativa foi justamente no número utilizado pelo WSL e o Windows, um com 101 e outro com 97 fotos (Fig. 10). Ainda que o WSL tenha efetuado o cálculo com menos imagens ele o fez na primeira tentativa, já o processamento no Windows teve que ser repetido várias vezes por conta de um erro não identificado que impossibilitava o cálculo final causando a interrupção do processo.

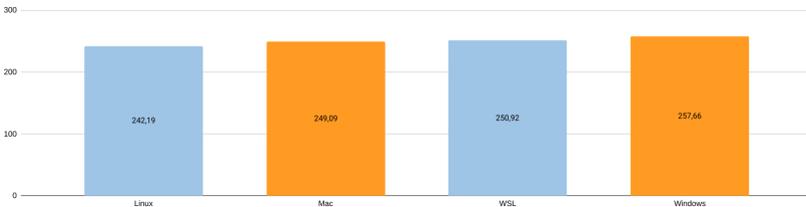


Figura11: Gráfico com o tamanho médio dos arquivos em MB.

Um fato a ser documentado é que não houve diferença significativa no tamanho médio dos arquivos gerados durante o processo (Fig. 11).

Foram efetuados testes em mídias e conexões diferentes para avaliar a influência das mesmas no resultado final do processo. O Linux e o

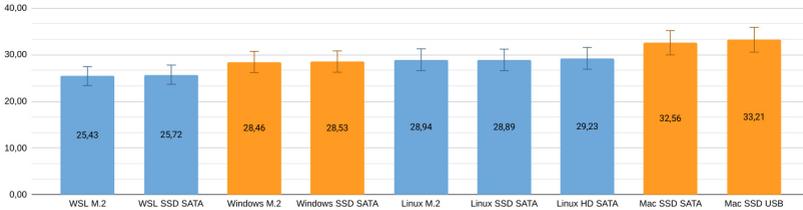


Figura12: Gráfico com o FTF em mídias e sistemas operacionais diferentes.

Windows foram instalados em um dispositivo SSD M.2 com uma taxa de leitura quase 4x maior do que o SSD SATA. O Mac foi testado com o um SSD SATA (conectado internamente) e o mesmo SSD conectado a uma entrada USB. O Linux também foi instalado e testado em um HD (disco) conectado internamente no SATA do notebook. Ao final, quando levada em consideração a mídia, no caso específico da fotogrametria não houve diferença significativa nos resultados dentro de um mesmo sistema operacional (Fig. 12).

3.4 Conclusão

Apesar de pequenas diferenças no tempo e no número de fotos que possam ser utilizadas nos cálculos envolvendo a fotogrametria no OrtogOnBlender, não houve grandes e inibidoras incompatibilidades entre os sistemas operacionais mais populares, com exceção do WSL, o que não garante que o mesmo seja robusto em todas as instalações, como demonstrado em testes externos ao computador utilizado aqui.

O fato é que a diferença não justifica a troca de um sistema no qual o usuário é familiarizado, por outro cujo os estudos muitas vezes terão que ser iniciados do zero, gastando um tempo precioso na adaptação do mesmo. O objetivo do OrtogOnBlender ao fornecer versões para o Windows, o Linux e o Mac é justamente contemplar os usuários de todos estes sistemas oferecendo um desempenho compatível, ainda que os processos que envolvem o funcionamento em mais baixo nível tenham diferenças significativas.

Felizmente a fotogrametria é uma pequena parte do processo de planejamento cirúrgico digital e ainda que uma sequência de fotos leve

de segundos a poucos minutos de diferença em relação a outro sistema, isso dificilmente impactará no tempo total do trabalho.

Reconstrução de Tomografias nos Sistemas Operacionais: Windows, Linux e Mac OS X

Cicero Moraes

3D Designer, Arc-Team Brazil, Sinop-MT

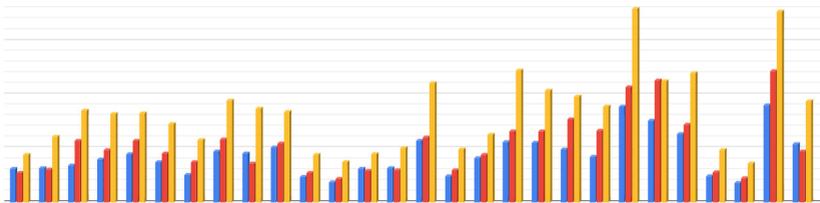
Rodrigo Dornelles

Cirurgião Plástico, Núcleo de Plástica Avançada - NPA, São Paulo-SP

Everton da Rosa

Cirurgião BME, Hospital de Base, Brasília-DF

DOI: <https://doi.org/10.6084/m9.figshare.13480326>



O presente capítulo tem por objetivo analisar o desempenho da reconstrução de tomografias computadorizadas executadas em um mesmo computador utilizando os sistemas operacionais: Windows, Linux e Mac OS X.

Atenção: Este material utiliza a seguinte licença Creative Commons: **Atribuição 4.0 Internacional (CC BY 4.0)**.

4.1 Descrição Técnica

Os autores selecionaram aleatoriamente 30 tomografias computadorizadas, sendo 16 médicas (CT-Scan) e 14 do tipo cone beam (CBCT), de modo a testar uma amostra heterogênea. A configuração do notebook e dos sistemas operacionais utilizados estão presentes no capítulo anterior intitulado **Fotogrametria 3D - Desempenho nos Sistemas Operacionais: Windows, Linux e Mac OS X**, também na *Descrição Técnica*.

Em linhas gerais trata-se de um notebook com processador i7 de 1.80 GHz, 20 GB de memória RAM com armazenamento SSD SATA de 480 GB.

Para simular uma situação que é de praxe para a maioria dos usuários, os arquivos DICOM foram gravados em HD externo e acessados via USB.

Aviso: Este capítulo é direcionado a usuários contumazes do OrtogOnBlender, de modo que não serão explanados conceitos básicos relacionados ao *add-on*, focando principalmente nos resultados do desempenho.

4.2 Resultados

Para permitir que os dados de tempo fossem capturados o add-on OrtogOnBlender recebeu uma programação específica que contava os segundos desde o início da reconstrução, ao usuário clicar o botão **“CT-Scan Reconstruction”** da seção DEFAULT até a importação e atualização da 3D View do Blender. Os resultados são apresentados ao usuário a partir da versão **2020-12-21**.

Das 30 tomografias selecionadas, duas apresentaram problema e não finalizaram a reconstrução no sistema operacional Windows 10, de modo que as mesmas foram descartadas ainda que tenham sido bem sucedidas no Linux e no Mac OS X. Ao final todos os sistemas reconstruíram 28 tomografias cada um (ver Tabela 1).

Uma média das 28 reconstruções em cada sistema foi extraída resultando em:

- Linux: 221 segundos (**3m41s**);
- Mac OS X: 266 segundos (**4m26s**);
- Windows: 418 segundos (**6m58s**).

O Linux e o Mac OS X ficaram praticamente empatados e em algumas situações o Mac foi até mais rápido do que o Linux. No entanto o Windows apresentou um desempenho significativamente inferior do que os dois outros sistemas ficando em terceiro em todos os resultados individuais. Os autores seguiram com outro teste que levantaria a margem de erro geral, o que permitiria aos usuários terem uma ideia da elasticidade do tempo. Uma mesma tomografia (ID 1) foi reconstruída 5 vezes em cada um dos sistemas operacionais

Teste	Linux	Mac OS	Windows
1	150	137	223
2	137	138	220
3	73	105	157
4	73	105	159
5	74	105	159

Observou-se um comportamento curioso e semelhante nos três sistemas operacionais, o primeiro e segundo resultados eram mais ou menos parecidos e do terceiro ao quinto o tempo reduzia drasticamente. No caso do Linux a redução foi significativamente maior do que os demais sistemas, dificultando o levantamento de uma margem de erro geral. Como em uma situação real o usuário geralmente faz apenas uma reconstrução, não faria sentido levantar a margem de erro, mantendo-se apenas os resultados apresentados no início do capítulo.

Um outro teste mais amplo foi efetuado visando isolar os dois fatores

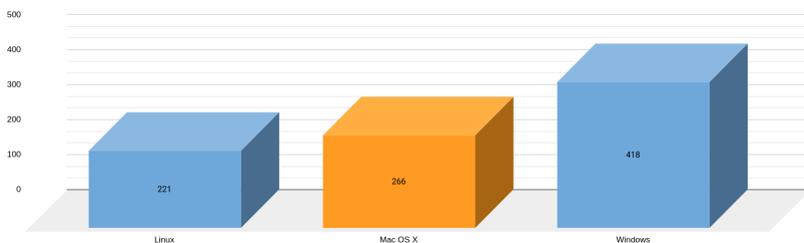


Figura1: Resultado final com a média de tempo em segundos da reconstrução em cada sistema operacional.

Tabela 1: Tempo da Reconstrução em Segundos

ID	Linux	Mac OS	Windows
1	156	137	223
2	158	153	307
3	172	287	428
4	201	244	410
5	226	286	415
6	189	227	364
7	127	187	289
8	239	294	473
9	229	180	437
10	256	276	420
11	118	137	222
12	95	111	187
13	157	146	224
14	161	150	254
15	286	302	555
16	123	151	247
17	206	221	315
18	281	330	613
19	277	330	521
20	248	386	492
21	211	333	445
22	447	536	900
23	379	568	563
24	319	362	601
25	122	141	245
26	90	113	182
27	452	609	887
28	271	237	471
Média	221	266	418

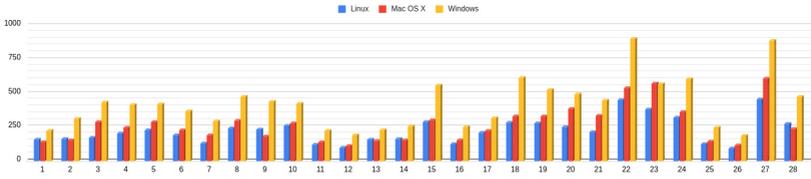


Figura2: Resultado final com todos os tempos individuais.

que mais influenciam na reconstrução de uma tomografia: a organização dos arquivos e o a reconstrução da malha 3D. Tomou-se como base os DICOMs disponíveis no diretório TOMOGRAFIA que acompanha a instalação do OrtogOnBlender. O teste consistiu em cronometrar o tempo necessário para que os arquivos fossem organizados e o tempo necessário para que a malha 3D fosse reconstruída a partir do diretório selecionado após a organização. Procedeu-se com os comando utilizando dispositivos diferentes (HD e SSD) ou com os mesmos dispositivos e conexões diferentes (SATA e USB).

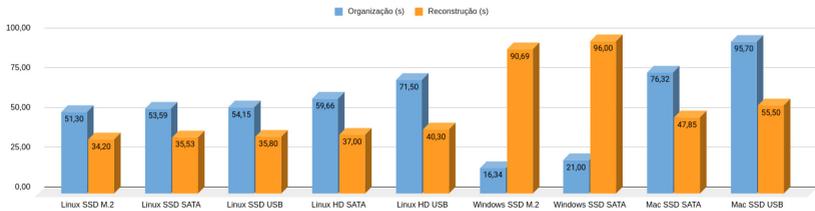


Figura3: Gáfcio de tempo em segundos da organização dos arquivos e reconstrução 3D.

Os resultados apresentam uma grande superioridade na organização de arquivos por parte do Windows. Todavia, o mesmo sistema que se mostrou notoriamente rápido na organização foi claramente o mais lento na reconstrução dos arquivos.

Quando os dados são diluídos em uma média, a ampla diferença não se mostrou um grande problema para o Windows, ao menos com o DICOM utilizado no exemplo, pois o mesmo posicionou-se em segundo lugar. Aparentemente o Linux e o Mac foram beneficiados pela abordagem da seção DEFAULT que foca mais na reconstrução do que na organização dos arquivos.

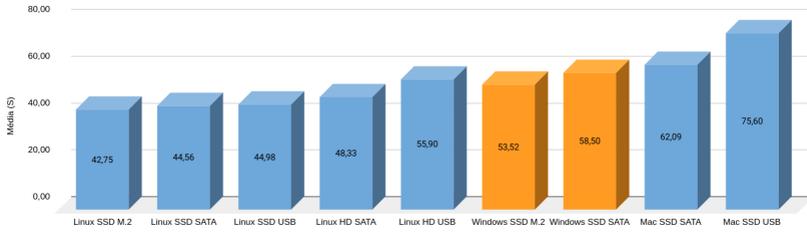


Figura4: Gráfico de tempo em segundos da organização dos arquivos e reconstrução 3D.

4.3 Conclusão

Ao que tudo indica o Linux e o Mac OS X levam uma acentuada vantagem em relação ao Windows na reconstrução de tomografias computadorizadas em malhas 3D. Como o objetivo desta abordagem é entender melhor as ferramentas do OrtogOnBlender e incrementar o seu desenvolvimento, evidenciou-se que melhorias precisam ser implementada para compatibilizar o desempenho dos binários do Windows com os outros sistemas, bem como solucionar os problemas de reconstrução a exemplo dos dois apresentados durante os testes com o Windows e que reduziu o grupo de 30 para 28 tomografias a serem avaliadas. No entanto, mesmo diante desta diferença significativa de tempo, isso pode não representar um grande problema aos usuários se for levado em consideração a familiaridade dos mesmos com o sistema e o tempo total do planejamento cirúrgico frente aos poucos minutos necessários para a reconstrução tridimensional de uma tomografia.

Sistema de Reconstrução de Tomografia Computadorizada Baseado no Slicer 3D e no DicomToMesh

Cicero Moraes

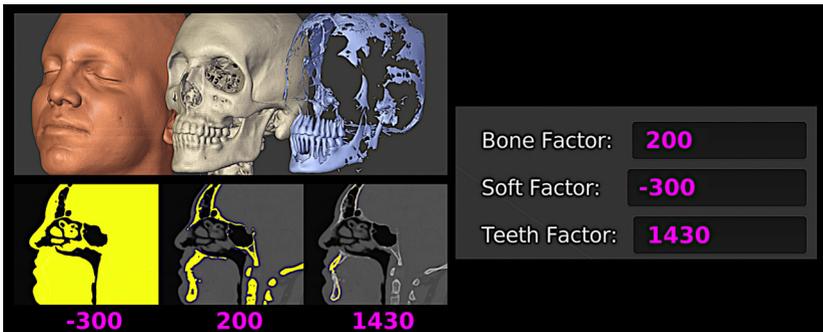
3D Designer, Arc-Team Brazil, Sinop-MT

Rodrigo Dornelles

Cirurgião Plástico, Núcleo de Plástica Avançada - NPA, São Paulo-SP

Everton da Rosa *Cirurgião BMF, Hospital de Base, Brasília-DF*

DOI: <https://doi.org/10.6084/m9.figshare.13513890>



O presente capítulo tem por objetivo apresentar a nova ferramenta de reconstrução de tomografias computadorizadas do OrtogOnBlender, bem como analisar o desempenho da reconstrução de tomografias executadas em um mesmo computador utilizando os sistemas operacionais: Windows, Linux e Mac OS X.

Atenção: Este material utiliza a seguinte licença Creative Commons: **Atribuição 4.0 Internacional (CC BY 4.0)**.

Aviso: Este capítulo é direcionado a usuários contumazes do OrtogOnBlender, de modo que não serão explanados conceitos básicos relacionados ao *add-on*, focando principalmente nos resultados do desempenho.

5.1 Apresentação

Altamente popular no meio médico, o formato de arquivo DICOM (Digital Imaging and Communications in Medicine) permite ao usuário visualizar não apenas um volume em forma de fatias, mas também utilizar estas fatias para reconstruir o conjunto em três dimensões, seja pela visualização de grades de pixel em 3D (voxel) ou em malha tridimensional (3D mesh).

Em face da popularização da impressão 3D, os arquivos DICOM têm sido amplamente reconstruídos em malhas 3D e exportados em formatos como STL, OBJ, PLY e afins.

A maior dificuldade dos usuários que iniciam os trabalhos de reconstrução 3D é justamente entender os parâmetros que envolvem toda essa abordagem e justamente por isso o OrtogOnBlender conta com uma série de ferramentas que automatizam o processo, tornando-o mais simples e acessível.

Ainda que forneça várias formas de visualizar e reconstruir uma tomografia computadorizada, cada uma dela tem os seus prós e contras. No decorrer dos últimos dois anos, uma série de abordagens foram testadas e na maioria dos casos o que os usuários buscam é reconstruir o tecido mole, os ossos e os dentes sem os ruídos gerados pelos artefatos.

Houve uma grande evolução até a criação da ferramenta DEFAULT, a mais utilizada hoje em dia e que foca em três dados para a reconstrução, extraídos de modo automático da tomografia: o modelo do tomógrafo o Convolution Kernel e o Series Number do grupo de fatias.

Ainda que funcione em grande parte dos casos a abordagem (DEFAULT) apresenta alguns problemas, como por exemplo, a não filtragem de estudos pela data, a presença de dois Series Number iguais em uma mesma tomografia e o erro causado pela ausência do tomógrafo ou do Convolution Kernel no grupo de fatias. Além disso, é muito comum que um ou outro arquivo DICOM não seja compatível com o reconstrutor 3D, o que acaba por impossibilitar o processo gerando uma série de erros.

A ferramenta SLICER, como o próprio nome indica é baseada no poderoso software de visualização e edição de arquivos DICOM, o 3D Slicer [Sli]. O mesmo utiliza uma série de bibliotecas baseadas em software livre, oferecendo um incontável número de módulos compatíveis com vários campos de conhecimento que envolvem as ciências da saúde como a odontologia, a cardiologia, a neurologia e afins.

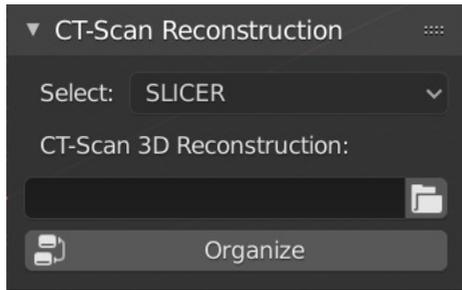
Ao explorar o poder de leitura, edição e exportação do Slicer é possível exportar DICOMs totalmente compatíveis com o DicomToMesh [Sch], um pequeno e poderoso software livre para reconstrução de tomografias computadorizadas.

Juntando as duas ferramentas em uma única interface, é possível fornecer ao usuário a possibilidade de trabalhar com uma série de aplicativos que simplificam o processo de reconstrução 3D de uma tomografia computadorizada ao passo que permite o uso de um grande número de arquivos de fontes diferentes, sem gerar os erros comuns nas outras abordagens oferecidas pelo OrtogOnBlender.

A proposta da ferramenta apresentada neste capítulo é dar um passo a mais na robustez da reconstrução automática, organizando de

modo mais claro as fatias, tornando a tomografia compatível com a biblioteca utilizada pelo DicomToMesh sem deixar de oferecer ao usuário algumas opções de correção manual, o que pode ser muito útil dependendo da situação e do modelo do tomógrafo.

5.2 Como Funciona



Figural: Ferramenta SLICER.

A ferramenta está localizada na seção **CT-Scan Reconstruction**, para acessá-la é necessário ir no menu **Select** e selecionar a opção **SLICER** (Fig. 1).

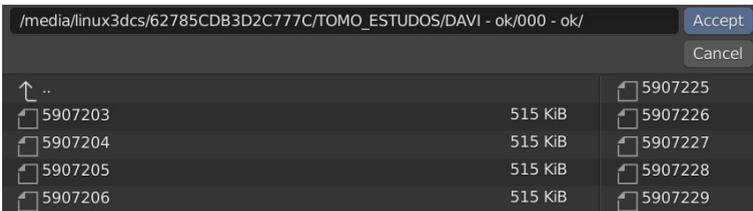


Figura2: Seleção do diretório.

Em seguida entrar no diretório onde a tomografia foi salva e clicar em **Accept** (Fig. 2).

Assim que o **Accept** é clicado a interface retorna ao estado anterior e na caixa do endereço é possível atestar a indicação do diretório. É importante notar que no campo **Tomograph** lê-se a indicação **NO INFO!!!** mostrando que por ora não há qualquer informação sobre o

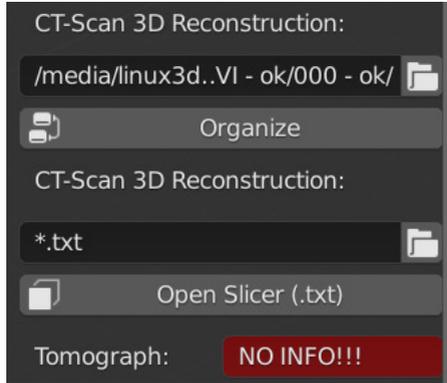


Figura3: Diretório selecionado.

tomógrafo em questão e logo abaixo também não é informado nenhum arquivo de texto contendo os dados das fatias, apenas uma indicação (*.txt) sobre o formato de arquivo (Fig. 3).

Isso acontece porque o sistema ainda não leu os dados presentes no diretório de tomografias, o processo é efetuado quando o usuário clica no botão **Organize**.

Slices	Patient Name	Date	Description	File
228	DIVT JAMES NORMAL	20170307	Sft Tissue 1.0	2.txt
228	DIVT JAMES NORMAL	20170307	Bone 1.0	1.txt
170	DIVT JAMES NORMAL	20170307	Bone 1.0 Corona	6.txt
153	DIVT JAMES NORMAL	20170307	Bone 1.0 Sagitt	3.txt
130	DIVT JAMES NORMAL	20170307	Sft Tissue 0.4	5.txt
130	DIVT JAMES NORMAL	20170307	Sft Tissue 0.4	4.txt
12	DIVT JAMES NORMAL	20170307	Sft Tissue VR.	7.txt
1	DIVT JAMES NORMAL	20170307	ANAMNESE	8.txt
1	DIVT JAMES NORMAL	20170307	2.0	0.txt

Figura4: Dados apresentados no Terminal.

Se tudo der certo, alguns segundos depois (ou minutos, dependendo da mídia e do número de arquivos) os dados relacionados a tomografia serão apresentados na saída do Terminal (Fig. 4).

A primeira coluna (Slices) informa o número de fatias presentes em cada grupo listando inicialmente, trata-se do grupo com o maior número de slices. A segunda coluna informa o nome e sobrenome do

paciente, a terceira a data em que a tomografia foi efetuada, a quarta informa a descrição do grupo de fatia e a última coluna informa o nome do arquivo.

Diferentemente do DEFAULT, o modo SLICER utiliza como elemento de organização de grupos o parâmetro **Series Instance UID** das fatias DICOM e essa abordagem contorna uma série de problemas como, por exemplo, o que foi supracitado envolvendo **Series Number** duplicado.

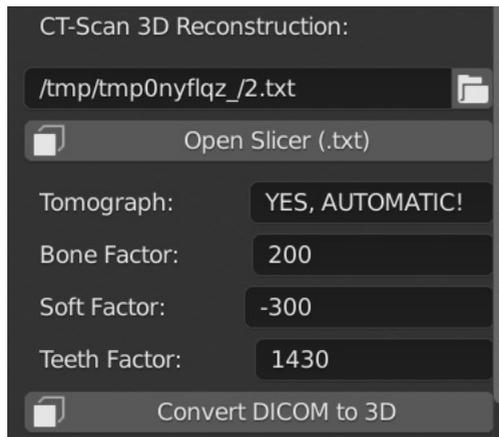


Figura5: Fatias organizadas.

Uma vez que os grupos foram organizados o OrtoGOnBlender testa se o diretório com mais fatias tem o tomógrafo cadastrado no banco de dados do *add-on*, se não tiver a mensagem **NO INFO!!!** permanece, se tiver o tomógrafo cadastrado o fundo vermelho é substituído pela cor cinza escuro e a mensagem que aparece será **YES, AUTOMATIC!**

Isso significa que os valores dos fatores foram ajustados para o grupo de fatias em questão, neste caso não houve alteração posto que os valores 200, -300 e 1430 são muito comuns em tomografias com o foco no tecido mole. Basta observar a imagem relacionada aos dados da tomografia e atestar que o **Description** do arquivo 2.txt, que foi selecionado automaticamente, tem a seguinte informação: **Sft Tissue 1.0**, ou seja, trata-se do tecido mole.

Basta então clicar em **Convert DICOM to 3D** (Fig. 5) para a tomografia

ser reconstruída.

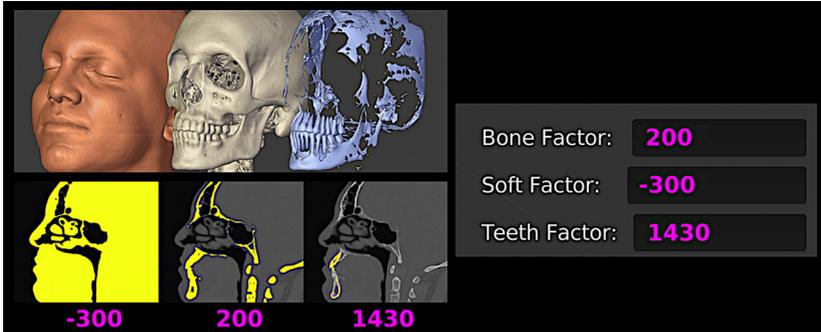


Figura6: Treshold indicado.

O sistema seta automaticamente os valores, mas caso haja algum problema na reconstrução por conta do Treshold, é possível abrir o grupo de fatias utilizando o Slicer e testar qual é o valor mais compatível com o trabalho em questão. Basta indicar o arquivo .txt desejado e em seguida clicar em **Open Slicer (.txt)**.

5.3 Tempo de Reconstrução em Sistemas Operacionais Diferentes

Os autores selecionaram aleatoriamente 30 tomografias computadorizadas, de modo a testar uma amostra heterogênea e apresentar uma média geral do tempo necessário para a reconstrução em três sistemas operacionais diferentes. A configuração do notebook e dos sistemas operacionais utilizados estão presentes no capítulo intitulado **Fotogrametria 3D - Desempenho nos Sistemas Operacionais: Windows, Linux e Mac OS X**, mais precisamente na *Descrição Técnica*.

Em linhas gerais trata-se de um notebook com processador i7 de 1.80 GHz, 20 GB de memória RAM com armazenamento SSD SATA de 480 GB.

Para simular uma situação que é de praxe para a maioria dos usuários, os arquivos DICOM foram gravados em um HD externo e aces-

sados via USB.

5.4 Resultados

Para permitir que os dados de tempo fossem capturados o add-on OrtogOnBlender recebeu uma programação específica que contava os segundos desde o início da reconstrução, ao usuário clicar o botão “**CT-Scan Reconstruction**” da seção SLICER até a importação e atualização da 3D View do Blender. Os resultados são apresentados ao usuário a partir da versão **2020-12-27**.

Das 30 tomografias selecionadas, uma apresentou problema e não finalizou a reconstrução nos três sistemas operacionais, de modo que a mesma foi descartada ainda que tenham sido bem sucedidas em outro computador com processador i9 rodando o Linux. Ao final todos os sistemas instalados no notebook reconstruíram 29 tomografias cada um (ver Tabela 1).

Tabela 1: Tempo da Reconstrução em Segundos

ID	Linux	Mac OS X	Windows
1	126	169	251
2	386	507	864
3	178	237	384
4	154	200	305
5	92	123	169
6	87	127	194
7	132	180	296
8	189	264	405
9	151	207	344
10	98	109	196
11	74	109	158
12	245	318	519
13	61	88	130
14	182	243	416
15	469	589	1034
16	174	230	386
17	108	150	229

continues on next page

Tabela 1 – continuação da página anterior

ID	Linux	Mac OS X	Windows
18	113	148	240
19	103	166	297
20	310	473	559
21	93	125	186
22	120	215	244
23	313	436	706
24	123	166	263
25	178	256	385
26	139	188	292
27	139	188	292
28	193	263	420
29	241	311	527
Média	173	237	372

Dentre as tomografias havia uma de DICOM único, ou seja, um único arquivo com mais de 1 GB de informação. O novo sistema, diferente dos demais (MANUAL, DEFAULT, BDENTAL, etc.), permite a reconstrução deste tipo de arquivo, aumentando ainda mais a praticidade e facilidade do sistema, ainda que apresente um tempo maior de reconstrução, resultado da conversão do arquivo único em várias fatias.

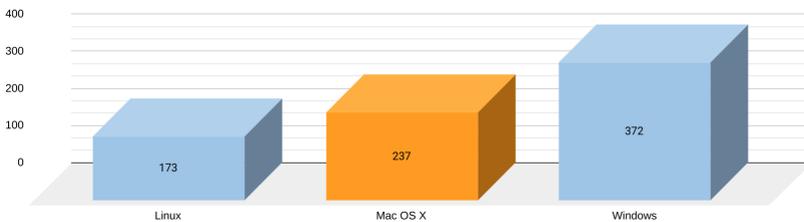


Figura7: Resultado final com a média de tempo em segundos da reconstrução em cada sistema operacional.

Uma média das 28 reconstruções (Fig. 7) em cada sistema foi extraída resultando em:

- Linux: **173** segundos (**2m53s**);
- Mac OS X: **237** segundos (**3m57s**);

- Windows: 372 segundos (6m12s).

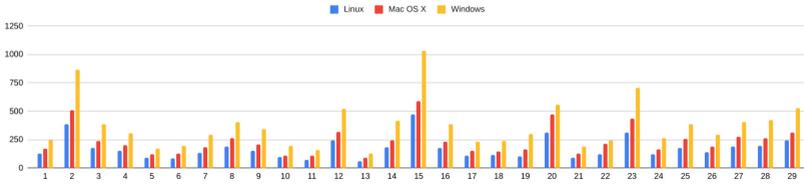


Figura8: Gráfico com o tempo (vertical) e ID (horizontal) apresentando as diferenças entre os sistemas operacionais.

É possível verificar pelo gráfico que houve um padrão de comportamento no tempo das reconstruções, sempre com o Linux em primeiro, o Mac OS X em segundo e o Windows em terceiro, geralmente com um pouco mais do dobro do tempo levado pelo Linux (Fig. 8).

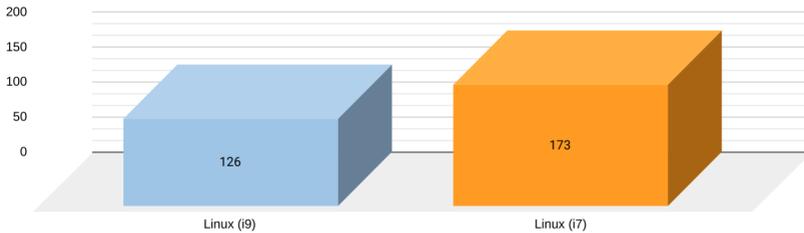


Figura9: Gráfico com a comparação do tempo entre dois processadores distintos: i9 e i7.

Não houve grande diferença no tempo de reconstrução médio entre um computador com processador i9 (9900K 3.60GHz) e o notebook com o processador i7 (8565U 1.80GHz), resultando em 126 segundos no primeiro e 173 segundos no segundo. Há no entanto uma informação a ser repassada, a de que o computador com o processador i9 reconstruiu as 30 tomografias, diferente do notebook com o processador i7 que reconstruiu uma a menos nos 3 sistemas operacionais testados na mesma máquina. O número não é significativo para evidenciar que o motivo da falha possa estar ligado ao processador e a potência geral da máquina, estudos complementares serão efetuados no futuro buscando a solução do problema, já que, apesar de falhar algumas tomos das 30 utilizadas o sistema de reconstrução

DEFAULT reconstruiu a tomografia que falhou no SLICER. Daí a importância de manter várias ferramentas disponíveis, posto que uma tomografia que não roda em uma, pode vir a funcionar em outra.

5.5 Conclusão

O novo sistema de reconstrução de tomografias a partir de arquivos DICOM mostrou-se funcional, prático e amplamente aplicável. Assim como toda ferramenta em desenvolvimento, há a possibilidade de uma ou outra incompatibilidade, mas é questão de tempo para que o problema seja identificado e solucionado.

Chamar a ferramenta de “nova” é em certa medida uma incompatibilidade com a real situação, posto que o que se vê na verdade é uma ferramenta evoluída a partir de outras, ainda em desenvolvimento, mas sempre focada no objetivo central de todo este trabalho, que é fornecer ao usuário a possibilidade de utilizar ferramentas livres e gratuitas de modo a oferecer aos seus pacientes mais segurança e previsibilidade graças a um robusto e centrado planejamento cirúrgico.

5.6 Agradecimentos

Ao Dr. Davi Sandes Sobral por ceder a tomografia utilizada neste capítulo e ao Dr. Bruno Aragão por esclarecimentos relacionados ao formato de arquivo DICOM, essenciais para a composição do sistema automático de reconstrução 3D utilizado pelo OrtogOnBlender.

Digitalização 3D a partir de Fotografias com o SMVS e o MVE no OrtoGOnBlender

Cicero Moraes

3D Designer, Arc-Team Brazil, Sinop-MT

Rodrigo Dornelles

Cirurgião Plástico, Núcleo de Plástica Avançada - NPA, São Paulo-SP

Everton da Rosa

Cirurgião BMF, Hospital de Base, Brasília-DF

DOI: <https://doi.org/10.6084/m9.figshare.13549595>



O presente capítulo tem por objetivo apresentar as ferramentas de digitalização 3D por fotogrametria SMVS e MVE no OrtoGOnBlender, bem como analisar o desempenho do processo em um mesmo com-

putador utilizando os sistemas operacionais: Windows, Linux e Mac OS X. Ao final do processo foram implementadas melhorias que por um lado tornaram a ferramenta mais rápida e por outro permitiram que os usuários reconstruíssem os modelos com mais qualidade.

Atenção: Este material utiliza a seguinte licença Creative Commons: **Atribuição 4.0 Internacional (CC BY 4.0)**.

Aviso: Este capítulo é direcionado a usuários contumazes do OrtoGOnBlender, de modo que não serão explanados conceitos básicos relacionados a *add-on*, focando principalmente nos resultados do desempenho.

6.1 Apresentação

A fotogrametria é um processo de digitalização de objetos a partir de fotografias que geralmente é composto pelas seguintes etapas:

- **Etapa 1:** Utilizando como referência as regiões compatíveis entre as fotografias o sistema encontra a localização da câmera em cada tomada posicionando-a nos eixos X,Y e Z, bem como a rotação no momento da captura. Ao final é gerada uma nuvem esparsa de pontos que apesar de reduzida já dá indicações de como é o objeto digitalizado.
- **Etapa 2:** Com os dados levantados na etapa anterior é gerada uma nuvem densa de pontos que lembra muito o objeto, posto que a mesma também já se encontra colorizada.
- **Etapa 3:** O sistema usa os dados da nuvem densa de pontos e reconstrói um modelo formado por malha tridimensional com a forma final do objeto, mas ainda sem coloração ou com a coloração efetuada por vertex color (geralmente com qualidade inferior a texturização por imagem)
- **Etapa 4:** Utilizando os dados das fotografias é projetada uma

textura (JPG) no objeto gerando o modelo final cujo a saída geralmente é no formato OBJ ou PLY.

O OrtogOnBlender oferece ao usuário várias formas de trabalhar com fotogrametrias, uma delas é a **SMVS+MVE**. Atualmente o sistema é composto por três aplicativos de código aberto: Multi-View Environment (MVE) [dD], Shading-aware Multi-view Stereo (SMVS) [LSHG16] e MVS-Texturing [WMG14].

Aviso: A versão anterior recebeu o nome de SMVS+Meshlab, mas sofreu alteração para SMVS+MVE depois que o MeshLab foi retirado do protocolo, sendo substituído pelo MVE na reconstrução da malha 3D.

O sistema descrito neste capítulo segue a lógica supracitada, mas precisamente com as seguintes etapas:

- **Makescene** (MVE): Cria a estrutura para os cálculos serem efetuados, organizando as cenas baseado nas fotografias.
- **SfmRecon** (MVE): Organiza a posição das câmeras no momento da captura.
- **SMVSRecon** (SMVS): Gera a nuvem densa de pontos, já colorizada.
- **FssRecon** (MVE): Reconstrói uma malha 3D a partir da nuvem densa de pontos.
- **MeshClean** (MVE): Limpa a malha 3D reconstruída retirando parte significativa dos ruídos.
- **TexRecon** (MVS-Texturing): Gera o UV map e projeta a texturização, baseada em uma série de arquivos de imagens.
- **OrtogOnBlender**: O OrtogOnBlender importa o modelo, simplifica o mesmo quando necessário (Decimate), gera um novo UV map e funde todas as texturas em apenas uma de modo a permitir o uso no modificador Displacement em conjunto com os modificadores MultiRes e Smooth, todos devidamente atribuídos após os melhoramentos efetuados já dentro do Blender.

6.2 Testes e Resultados

Foram selecionadas 10 seqüências de fotografias salvas em diretórios com nomes distintos. Essas seqüências são as mesmas apresentadas no capítulo *Fotogrametria 3D - Desempenho nos Sistemas Operacionais: Windows, Linux e Mac OS X*, com excessão do primeiro grupo substituído pela seqüência de 12 fotografias presentes no diretório de arquivos didáticos do OrtogOnBlender. Outro aspecto divergente do capítulo citado é o fato dos demais diretório terem sido salvos em um HD externo, no entanto, como visto nos capítulos anteriores, como se tratam de poucos arquivos a presença dos mesmos em dispositivos diferentes não altera significativamente o tempo da digitalização 3D.

A configuração do notebook e dos sistemas operacionais utilizados estão presentes na *Descrição Técnica* do capítulo supracitado. Em linhas gerais trata-se de um notebook com processador i7 de 1.80 GHz, 20 GB de memória RAM com armazenamento SSD SATA de 480 GB.

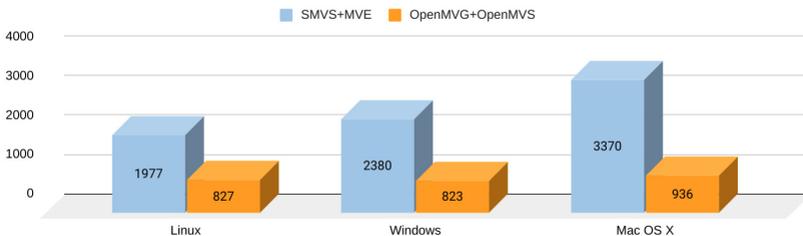


Figura1: Comparação do tempo médio em segundos.

Foram rodados as 10 seqüências de fotos nos 3 sistemas operacionais e extraída uma média de cada um deles (Fig. 1), essa média (SMVS+MVE) foi comparada com o tempo de digitalização do OpenMVG+OpenMVS resultando em uma diferença significativa sendo o SMVS+MVE em relação ao OpenMVG+OpenMVS:

- 2,39x mais lento no Linux;
- 2,89x mais lento no Windows;
- 3,60x mais lento no Mac OS X.

Um amplo estudo comparativo de ferramentas de fotogrametria foi

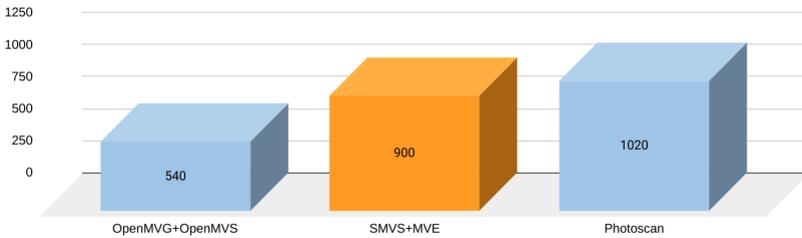


Figura2: Comparação do tempo médio em segundos em testes anteriores.

efetuado há algum tempo [MdRD20], fornecendo a base para o desenvolvimento das ferramentas presentes no OrtogOnBlender. Na ocasião digitalizaram-se a face de 13 indivíduos de idades e ancestralidades diferentes, resultando em um tempo médio muito compatível com a diferença encontrada neste capítulo (Fig. 2). Ainda que o SMVS+MVE tenha levado quase o dobro do tempo do OpenMVG+OpenMVS a ferramenta foi mais rápida que o Photoscan (hoje [Metashape](#)³). No entanto, o fato de ser mais rápido do que uma ferramenta paga e específica não implica em se adequar ao nível de praticidade esperado pelo usuário, ainda mais quando, em linhas gerais o funcionamento do OpenMVG+OpenMVS é superior ao SMVS+MVE, tanto na quantidade de superfície quanto na qualidade da textura.

Algumas mudanças foram implementadas no add-on de modo a tornar a digitalização mais rápida ao passo que permita ao usuário optar por uma reconstrução mais detalhada e demorada.

Em linha gerais a aparência da interface (Fig. 3) segue compatível com a versão anterior:

1. O usuário expande a seção **Photogrammetry Start** e seleciona a opção **SMVS+MVE**;
2. É necessária a seleção do diretório contendo as fotos;
3. Assim que setado o diretório basta clicar em **Alternative Photogrammetry I** para iniciar o processo.

A diferença em relação a versão anterior é que agora o usuário tem à sua disposição a opção **Decrease scanning time!** que reduz o tempo

³ <https://www.agisoft.com/downloads/installer/>

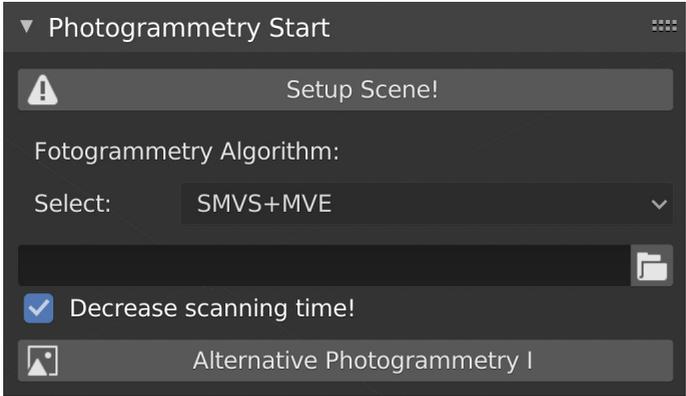


Figura3: Interface do SMVS+MVE no OrtoGOnBlender.

da digitalização e vem ativada por padrão. Caso seja desativada a fotogrametria será efetuada de modo mais detalhado e demorado.



Figura4: SMVS+MVE default, SMVE+MVE detalhado, OpenMVG+OpenMVS default, OpenMVG+OpenMVS detalhado.

Ao serem alinhados os resultados da fotogrametria efetuado com as ferramentas SMVS+MVE e OpenMVG+OpenMVS apresentam modelos compatíveis mas com pequenas diferenças. Há uma tendência do SMVS+MVE digitalizar mais detalhes (Fig. 4 à esquerda) do que a outra ferramenta (Fig. 4 à direita).

O SMVS+MVE costuma gerar texturas com qualidade inferior a outra ferramenta e isso fica mais evidente em capturas que não mantém a mesma exposição entre as fotografias, gerando uma imagem bastante poluída, sem homogeneidade nas cores. No caso do teste



Figura5: SMVS+MVE default, SMVE+MVE detalhado, OpenMVG+OpenMVS default, OpenMVG+OpenMVS detalhado. Modelos texturizados.

apresentado aqui (Fig. 5) a textura foi gerada de modo bem sucedido graças à uma cena com iluminação difusa e constante.



Figura6: SMVS+MVE default, SMVE+MVE detalhado, OpenMVG+OpenMVS default, OpenMVG+OpenMVS detalhado. Displacement ativo.

Os modelos passam a ter a aparência mais compatível uns com os outros a partir do momento em que o Displacement é ativado (Fig. 6). Essa opção utiliza a textura para criar detalhes no relevo da superfície e é muito utilizada para a confecção de próteses faciais.

Os detalhes da superfície ficam ainda mais aguçados caso o modificador Multires seja ativado (Fig. 7). Ao observar a base do nariz de todos os modelos é possível analisar a qualidade da reconstrução posto que os modelos mais detalhados foram mais bem sucedidos ao digitalizarem aquela região. Nos modelos detalhados as bases dos narizes são iguais, já nos modelos simplificados (default) é possível



Figura7: SMVS+MVE default, SMVE+MVE detalhado, OpenMVG+OpenMVS default, OpenMVG+OpenMVS detalhado. Multires e Displacement ativos.

identificar pequenas diferenças com superfície positiva (para fora) e negativa (para dentro). Fora essa região, as demais que receberam ampla cobertura fotográfica foram reconstruídas de modo muito semelhante.

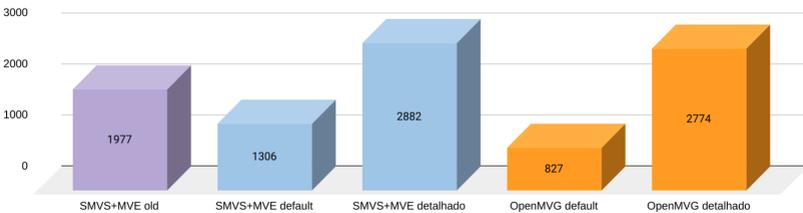


Figura8: Média do tempo em todas as ferramentas.

Ao se comparar o tempo médio de digitalização de cada ferramenta (Fig. 8) é possível atestar que o tempo da nova configuração default da SMVS+MVE é significativamente menor do que a configuração clássica (old). Ainda que o default do SMVS+MVE demore geralmente mais do que o tempo do default do OpenMVG+OpenMVS, a configuração detalhada das duas leva quase o mesmo tempo para ser efetuada (2882 vs. 2774 segundos).

Inicialmente a diferença de tempo entre as ferramentas era de 2,39x, agora caiu para 1,58x no Linux e pode servir de base para a projeção nos outros sistemas operacionais.

6.3 Conclusão

O SMVS+MVE não é a ferramenta de fotogrametria padrão do Orto-
gOnBlender, mas dependendo da situação pode ser uma alternativa
ao OpenMVG+OpenMVS e até suplantá-lo em qualidade. Uma das
vantagens do Orto-
gOnBlender é justamente oferecer várias aborda-
gens possíveis para a solução de um mesmo problema focando sem-
pre na possibilidade de efetuar uma tarefa necessária ao usuário.
Ainda que um comando possa levar mais tempo do que outro para
fazer a mesma tarefa, invariavelmente o usuário se deparará com
uma situação em que a ferramenta padrão não atenderá às suas ne-
cessidades e é justamente aí que as alternativas como o SMVS+MVE
vêm a calhar.

Reconstrução de Voxel Data no OrtogOnBlender

Cicero Moraes

3D Designer, Arc-Team Brazil, Sinop-MT

Marcos Graf

Cirurgião-Dentista, Bombinhas-SC

Rodrigo Dornelles

Cirurgião Plástico, Núcleo de Plástica Avançada - NPA, São Paulo-SP

Everton da Rosa

Cirurgião BMF, Hospital de Base, Brasília-DF

DOI: <https://doi.org/10.6084/m9.figshare.13670134>



O presente capítulo tem por objetivo apresentar a reconstrução de tomografias por voxel data a partir de arquivos DICOM via *add-on* OrtogOnBlender.

Atenção: Este material utiliza a seguinte licença Creative Commons: **Atribuição 4.0 Internacional (CC BY 4.0).**

7.1 Introdução

Os arquivos DICOM fornecem ao usuário a possibilidade de visualizar estruturas internas com uma ampla gama de níveis permitindo em muitos casos, como na saúde humana, segmentar ou definir tecidos específicos a serem reconstruídos em 3D, como por exemplo, ossos, vias aéreas e tecido mole em geral.

O uso do formato DICOM no OrtogOnBlender se dá majoritariamente na reconstrução de malhas 3D que passarão por osteotomia (*boolean*) e simulação de deformação de tecido mole (*rig*).

O problema da reconstrução de malhas 3D mora justamente na limitação da região de interesse a ser reconstruída. Muitos usuários não compreendem como uma tomografia que lhes é apresentada com detalhes de vasos sanguíneos, raízes de dentes bem definidas e até cabelos não resultem em uma malha igualmente limpa e bem definida quando efetuado o processo de reconstrução.



Figural: Comparação entre uma reconstrução de malha 3D (à esquerda) e o voxel (à direita).

Ao se comparar as reconstruções (Fig. 1) é possível, em um primeiro momento, atestar que a reconstrução de uma malha 3D (à esquerda) resulta em um modelo simplificado com a estrutura geral do tecido de interesse (mole, ossos e dentes) ao passo que o voxel (à direita) conta com um grande detalhamento dos tecidos, permitindo

ao usuário visualizar detalhes dos músculos e da massa encefálica, por exemplo.

Grosso modo, comparar uma malha 3D a um voxel é como comparar uma impressão 3D monocromática a um holograma. A primeira é uma representação simplificada e sólida do original, o segundo tem mais detalhes, mas não é sólido, apesar de tridimensional.

O fato é que cada abordagem oferece vantagens e o usuário pode explorá-las da melhor forma possível.

A exemplo do que já acontece com o crânio composto, onde uma digitalização de arcada em alta resolução substitui a estrutura em mais baixa resolução advinda da tomografia, o usuário pode complementar o planejamento com os dados do voxel data em uma mesma cena ampliando as potencialidades do mesmo. Com o voxel é possível identificar detalhes de uma estrutura que podem não ser reconstruídas em uma malha 3D, como um vaso ou uma variação de densidade de tecido. Ao ter todo o conjunto de dados disponíveis o planejamento pode ser elevado a outro patamar de confiabilidade.

7.2 Como Funciona - Aspectos Gerais

O Blender 3D não oferece suporte nativo ao formato de arquivo DICOM, mas graças a flexibilidade no desenvolvimento, o OrtogOnBlender resolveu a limitação e passou a oferecer suporte tanto a visualização, quanto a reconstrução por malha 3D e voxel data [MdrD20].

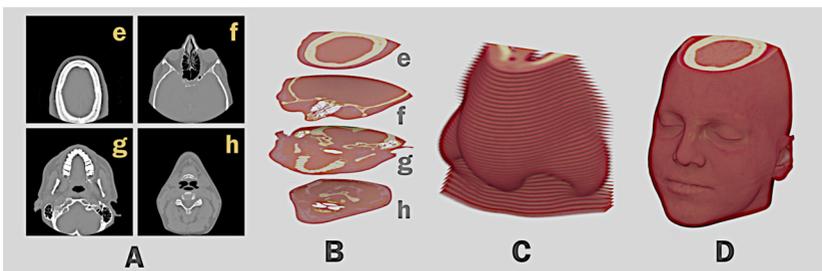


Figura2: Voxel Data importado pelo OrtogOnBlender.

O funcionamento segue um princípio bem simples (Fig. 2), as fatias

de um arquivo DICOM são convertidas em imagens (A), utilizando os dados contidos em cada um dos arquivos originais o algoritmo extrai a localização de cada uma das fatias posicionando-as no espaço tridimensional, além de tratar as cores aplicando transparência (B). Por ser uma imagem cada fatia tem a espessura 0 (zero), mas pela sobreposição das várias camadas cria-se a ideia de volume (C), resultando na região de interesse (D).

Importante: O número de fatias é bastante elevado, geralmente na casa das centenas. No exemplo acima descrito foram utilizadas apenas algumas fatias (A, B), ocultando-se outras que se encontravam entre as mesma para evitar poluição visual.



Figura3: Fatias axiais.

Inicialmente a ferramenta de voxel fornecida pelo OrtogOnBlender permitia apenas a reconstrução das fatias do plano axial (Fig 3). Se o volume for visto pelo topo é possível visualizar a estrutura da peça (Top), o problema é que, por serem imagens sem espessura nada se vê se observado pela frente e pela direita (Front e Right). Para o exemplo aqui presente, foi necessário ativar o *wireframe* para que as fatias fossem visualizadas pela frente e lateral, caso contrário nada se veria na cena. No entanto, mesmo se tratando de um volume formado por planos sem espessura, ao se orbitar a peça ainda que levemente, é possível visualizar a estrutura tomografada (Ortho).

Para evitar a falta de informação em qualquer um dos planos de visualização foi implementado no algoritmo do OrtogOnBlender a geração dos demais planos, ou seja, o coronal e o sagital (Fig. 4). Se observados isoladamente cada um dos planos apresenta as mesmas falhas em pontos diferentes, mas quando todos eles são agrupados

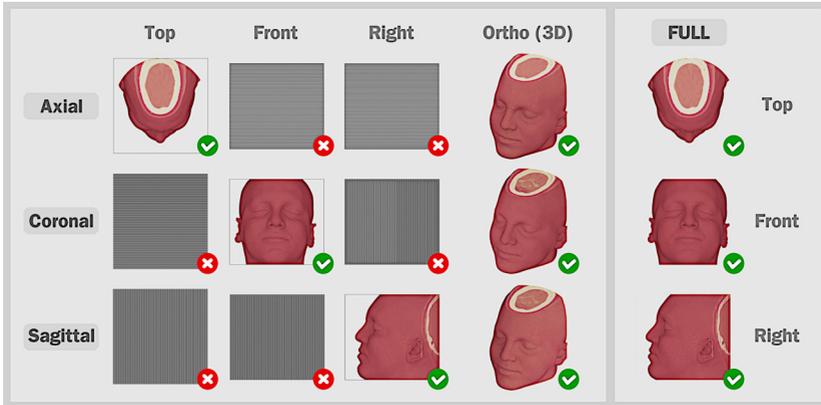


Figura4: Fatias completas nos planos axial, coronal e sagital.

no mesmo espaço a peça reconstruída é visualizada completamente (FULL).

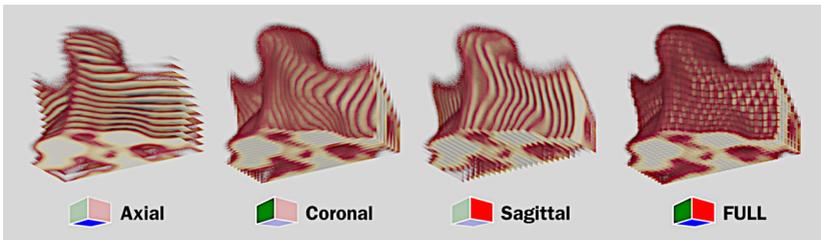


Figura5: Exemplo de renderização nos planos.

Diferentemente do Slicer [Sli] e outros softwares de visualização de arquivos DICOM, que criam pequenos cubos efetivamente volumétricos para a formação do voxel, o OrtogOnBlender gera o modelo pela intersecção das fatias. O resultado é suficientemente bom, de modo que a intersecção das fatias só é vista com o uso de um zoom de alto valor (Fig. 5).

Até a presente versão (2021-01-28) o usuário tem à sua disposição quatro configurações de visualização (Fig. 6):

- **Default:** com todos os tecidos visíveis;
- **Bone:** com apenas os ossos visíveis;

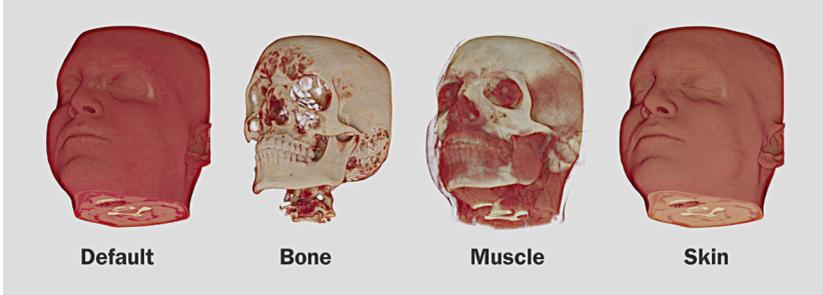


Figura6: Modos de visualização do voxel.

- **Muscle:** com os ossos e os músculos visíveis
- **Skin:** com pele mais densa e clara.

Importante: Os modos de visualização costumam funcionar muito bem em tomografias médicas (CT-Scan) e geralmente funcionam também em tomografias cone beam (CBCT), mas é possível que em alguns casos a apresentação não seja bem sucedida, principalmente nas tomografias cone beam, sendo necessária uma edição manual do *shader* [Foub]. Mais estudos serão necessários para a adequação do algoritmo ao modelo do tomógrafo.

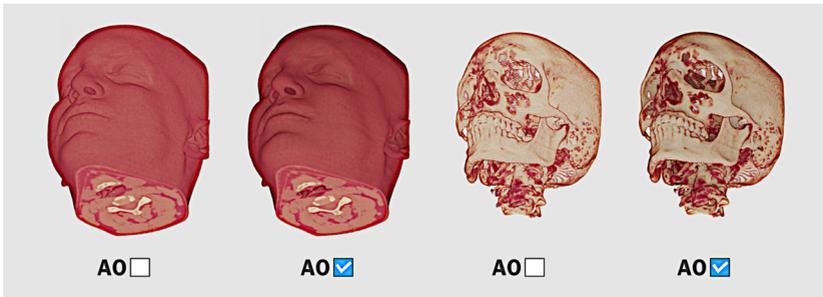


Figura7: Voxel Data com e sem Ambient Occlusion (AO).

Há ainda a opção de ativar o **Ambient Occlusion (AO)** [Foua] para evidenciar a profundidade na visualização da malha (Fig. 7). Apesar de melhorar a questão da profundidade, nem sempre o AO é a melhor opção, seja pela confusão que pode causar em modelos com transpe-

rência, seja pela lentidão que pode ocorrer em alguns computadores com GPUs menos potentes.

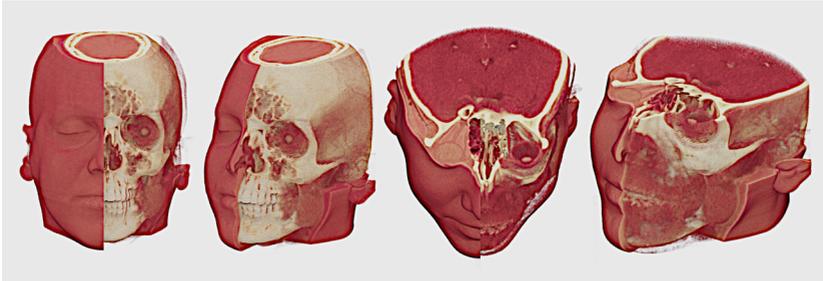


Figura8: Composição de voxel data com voxel data.

Por se tratar de um conjunto de planos com *shaders* próprios, é possível fazer composição de voxel data com voxel data utilizando configurações de material diferentes, cruzando com segmentação por cálculo booleano (Fig. 8).

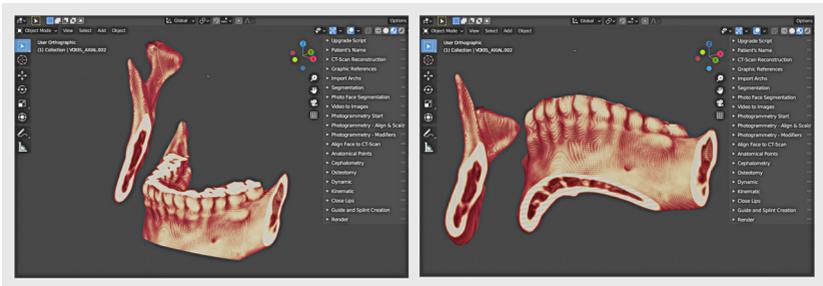


Figura9: Osteotomia efetuada por *boolean*.

É permitido ao usuário explorar as vantagens do voxel e da malha 3D (mesh) em uma mesma cena, um exemplo possível é isolar peças anatômicas como uma mandíbula (Fig. 9) e proceder com a osteotomia digital da mesma. A mandíbula do voxel data conta com mais dados estruturais do que a reconstruída em malha 3D, o que pode ser particularmente útil para identificar regiões críticas como aquelas compostas por densidades diferentes ou mesmo a identificação de vasos e outras partes. No caso da imagem de exemplo a malha em questão está oculta e serviu apenas para proceder com a booleana de

intersecção.

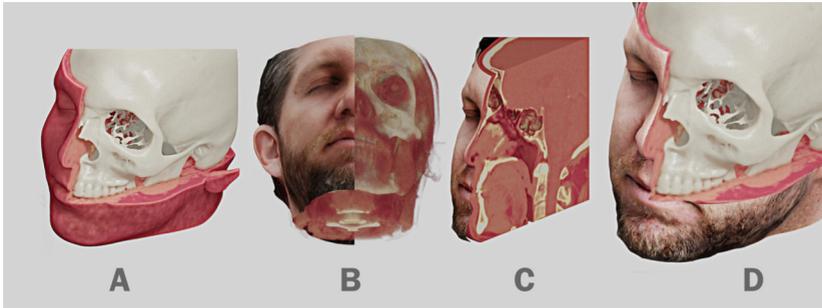


Figura10: Exemplos de composição com o voxel data.

Outrossim, composições de voxel e malha são possíveis e podem gerar modelos úteis e didáticos, seja a junção da malha reconstruída com o voxel (Fig. 10, A) ou o voxel com fotogrametria (B, C) ou ainda uma junção da malha reconstruída a partir da tomografia com o voxel e uma fotogrametria (D). Se for bem estruturado o modelo final oferece ao usuário uma ampla gama de informações extraindo o melhor de cada abordagem.

A fotogrametria, ou seja, a digitalização de uma face em 3D a partir de fotografias é outro exemplo de abordagem com a vantagem da texturização ou coloração da estrutura, ainda que não conte com elementos internos como a tomografia. No entanto, é uma abordagem que vem a somar no planejamento ao passo que fornece um rosto mais compatível com a realidade, posto que as tomografias usadas no meio médico ainda não fornecem a textura (coloração) da pele, seja por reconstrução de malha ou por voxel data.

7.3 Como Funciona na Prática

1. Na seção **CT-Scan Reconstruction** do **OrtogOnBlender** o usuário selecionará a opção **VOXEL_FULL SLICER!** (Fig. 11);
2. Em **CT-Scan Preparing** é necessário clicar no ícone do diretório e escolher a pasta onde se encontram os arquivos DICOM. Em seguida clicar em **Accept**;



Figura11: Interface do VOXEL-FULL SLICER.

3. Ao clicar em **Organize** o OrtogOnBlender vai varrer o diretório e os subdiretórios listando todos os arquivos DICOMs disponíveis;
4. No terminal serão apresentados os estudos relacionados a tomografia em questão com os dados do nome do paciente, número de fatias e o nome do arquivo .txt relacionado ao grupo de fatias;
5. O sistema escolherá o estudo com o maior número de fatias, mas essa escolha automática pode ser alterada, caso o usuário julgue necessário;
6. Assim que o estudo for selecionado, basta clicar em **Import DICOM Slices** e a importação se fará, inicialmente com as fatias axiais e posteriormente com as fatias coronais e sagitais. Nesse ínterim a interface do Blender ficará inacessível (tratava), até que o processo seja finalizado.

Aviso: O usuário pode desabilitar a opção **Decrease to 300!** e assim reconstruir o volume com a matriz nas dimensões originais, no entanto, isso pode aumentar consideravelmente o tempo da reconstrução, bem como gerar travamento em computadores com GPUs menos pontentes.

Importante: O sistema de organização é o mesmo apresentado no capítulo **Sistema de Reconstrução de Tomografia Computadorizada Baseado no Slicer 3D e no DicomToMesh**, especificamente em *Como*

Funciona. Em caso de dúvida é recomendado que o leitor consulte o capítulo citado.

Parte das ferramentas da seção **View/Shader/Material** foram abordadas no tópico anterior. O **Ambient Occlusion** vem desabilitado por padrão de modo a evitar lentidão em computadores com GPUs mais lentas.

O **Use Scene World** permite clarear o voxel data quando observado pela parte inferior. A ferramenta vem desabilitada por padrão, posto que ao ser acionado o *background* do 3D View passa a ser cinza-claro, o que pode causar desconforto visual para alguns usuários.

Logo abaixo estão as opções de *shaders* para visualizar os ossos (Bone), músculos (Muscle), pele (Skin) e retornar ao padrão (Default).

Caso o usuário opte por ver apenas uma fina fatia em cada plano, o *add-on* oferece a possibilidade de fazê-lo ao clicar no botão **Prepare Planes!**.

7.4 Testes e Resultados

Os autores selecionaram aleatoriamente 30 tomografias computadorizadas, de modo a testar uma amostra heterogênea e apresentar uma média geral do tempo necessário para a reconstrução em três sistemas operacionais diferentes. A configuração do notebook e dos sistemas operacionais utilizados estão presentes no capítulo intitulado **Fotogrametria 3D - Desempenho nos Sistemas Operacionais: Windows, Linux e Mac OS X**, mais precisamente na *Descrição Técnica*.

Em linhas gerais trata-se de um notebook com processador i7 de 1.80 GHz, 20 GB de memória RAM com armazenamento SSD SATA de 480 GB.

Para simular uma situação que é de praxe para a maioria dos usuários, os arquivos DICOM foram gravados em um HD externo e acessados via USB.

O teste foi feito em duas etapas, primeiramente levantou-se o tempo da organização dos arquivos e posteriormente o tempo da recons-

trução dos mesmos em um objeto 3D. O fator final é uma junção dos dois valores.

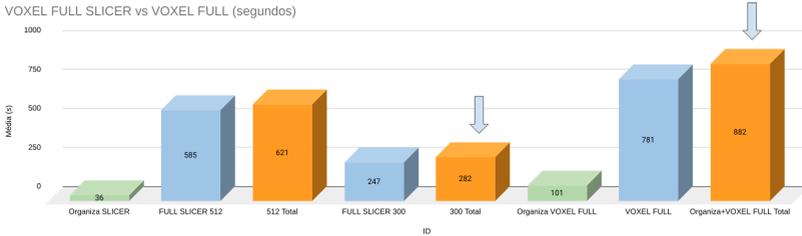


Figura12: Comparação em relação ao método padrão anterior (10 tomografias).

Em um primeiro momento foram utilizadas apenas 10 tomografias para a comparação de abordagens diferentes na reconstrução de voxel data.

As 10 tomografias separadas foram testadas na primeira ferramenta desenvolvida para a visualização pixels tomográficos, a VOXEL-FULL. Esse sistema utiliza a organização baseada em cópia de arquivos, o que já colocava a ferramenta em desvantagem, posto que em média essa etapa levou **101 segundos** frente a média de **36 segundos** necessária pelo outro sistema (VOXEL-FULL SLICER) baseado em varredura de diretórios (Fig. 12).

Aviso: Se tratam de dois sistemas distintos, o mais antigo nomeado de VOXEL-FULL e o mais novo nomeado de VOXEL-FULL SLICER. Os nomes parecidos podem eventualmente gerar confusão.

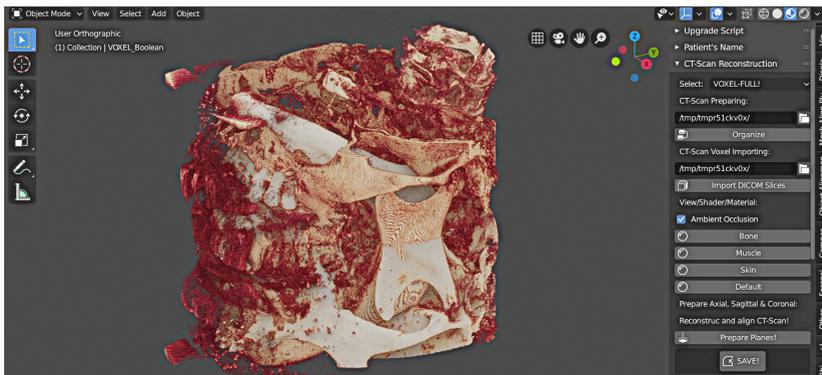
Na segunda parte do teste a ferramenta VOXEL-FULL levou em média **882 segundos** para reconstruir a tomografia. Isso acontece porque a mesma utiliza a tomografia em suas dimensões originais e isso pode ser problemático em matrizes maiores de 512x512x512, gerando modelos que rodem muito lentamente em computadores com GPUs de baixo poder de processamento.

Inicialmente o sistema VOXEL-FULL SLICER havia sido programado para reduzir a tomografia, caso a matriz fosse maior do que 512x512x512. No entanto, mesmo após a redução, a órbita na visuali-

zação apresentava lentidão em muitos casos. Em face a essas dificuldades foram efetuados vários testes até que uma nova matriz, com resolução suficientemente satisfatória foi escolhida. O novo valor padronizado foi o de 300x300x300.

Em linhas gerais a redução da matriz não apenas repercutiu em uma melhor visualização em tempo real, como tornou o processo muito mais rápido. Comparando as matrizes, a de 512 levou em média 586 segundos para ser reconstruída ao passo que a de 300 levou 247. Ao final o processo todo, envolvendo organização + reconstrução resultou em: **612 segundos vs. 282 segundos**.

Ao se comparar os processos do VOXEL-FULL versus o VOXEL-FULL SLICER (300) chegou-se a **882 segundos** do primeiro, frente a **282 segundos**, ou seja, o sistema baseado no Slicer é **3,13 vezes mais rápido** que o anterior.



Figural3: Reconstrução de voxel data com erro de rotação de eixo.

É importante deixar registrado que um dos motivos para a criação da ferramenta baseada no Slicer, além do tempo, eram problemas relacionados a reconstrução dos planos axial, sagital e coronal em rotações incompatíveis com a real posição dos mesmos (Fig. 13).

Das 10 tomografias escolhidas para os testes iniciais, o sistema anterior (VOXEL-FULL) apresentou problema de rotação em 7 deles (Fig. 14, células em vermelho), ao passo o novo sistema (VOXEL-FULL SLICER) reconstruiu todos os volumes sem apresentar o mesmo erro.

Definido o valor padrão da matriz e feita a comparação com a aborda-

ID	Ajusta Slicer	Full Slicer 512	512 Total	Full Slicer 300	300 Total	Ajusta Manual	Voxel Full	Manual+Full Total
1	67	495	562	205	272	118	485	603
2	85	566	651	125	210	277	554	831
3	29	545	574	256	285	86	1483	1569
4	21	594	615	278	299	64	561	625
5	42	567	609	279	321	59	1492	1551
6	14	605	619	268	282	55	587	642
7	1	507	508	277	278	52	740	792
8	26	730	756	258	284	73	720	793
9	69	929	998	277	346	179	884	1063
10	4	310	314	242	246	47	299	346
Média (s)	36	585	621	247	282	101	781	882

Figura14: Planilha de testes com reconstruções problemáticas em vermelho.

gem utilizada anteriormente, os testes se voltaram ao desempenho nos sistemas operacionais Linux, Windows e Mac OS X, agora utilizando as 30 tomografias.



Figura15: Comparação entre sistemas operacionais diferentes (30 tomografias).

Resultado organização dos arquivos (Fig. 15):

- 1º Mac OS X: 29 segundos.
- 2º Linux: 44 segundos.
- 3º Windows: 56 segundos.

Resultados da reconstrução 3D:

- 1º Linux: 310 segundos.
- 2º Windows 422 segundos.
- 3º Mac OS X: 565 segundos.

Resultado final, juntando as etapas:

- 1º Linux: 354 segundos (5m54s).
- 2º Windows 478 segundos (7m58s).

- 3º Mac OS X: 593 segundos (9m53s).

7.5 Conclusão

A nova ferramenta se mostrou funcional em 100% das reconstruções efetuadas, além de ser 3 vezes mais rápida que a abordagem utilizada anteriormente. No entanto, ainda que o processo tenha se mostrado mais célere que a opção anterior, o mesmo ainda é lento se compararmos com o tempo que os editores de DICOM como o Slicer levam para renderizar um modelo baseado em voxel data.

A grande vantagem dessa metodologia é que ela oferece ao usuário um objeto de cena que pode ser explorado de maneiras diversas e muitas vezes bem mais flexível do que aquelas oferecidas pelos editores do mercado, vide o exemplo da composição de voxel data, bem como a junção de várias abordagens em uma só cena (voxel+mesh+fotogrametria).

7.6 Agradecimentos

Agradecimentos ao Dr. Richard Gravalos e ao Dr. Davi Sandes Sobral, por cederem as suas tomografias ao estudo aqui apresentado.

Digitalização 3D de Faces a partir de Vídeos

Cicero Moraes

3D Designer, Arc-Team Brazil, Sinop-MT

Rodrigo Dornelles

Cirurgião Plástico, Núcleo de Plástica Avançada - NPA, São Paulo-SP

Everton da Rosa

Cirurgião BMF, Hospital de Base, Brasília-DF

DOI <https://doi.org/10.6084/m9.figshare.14075456>



O presente capítulo tem por objetivo apresentar a alternativa de digitalizar faces a partir de vídeos, utilizando a fotogrametria no Orto-
gOnBlender.

Atenção: Este material utiliza a seguinte licença Creative Commons: **Atribuição 4.0 Internacional (CC BY 4.0)**.

Aviso: Este capítulo é direcionado a usuários contumazes do OrtoGOnBlender, de modo que não serão explanados conceitos básicos relacionados ao *add-on*, focando principalmente nos resultados dos experimentos.

8.1 Apresentação

O hábito de fotografar com dispositivos digitais se tornou bastante comum na última década graças a popularização dos *smartphones*. A digitalização 3D por fotografias, ou fotogrametria, permite que os usuários explorem esses equipamentos para gerar modelos de grande qualidade. No entanto, se por um lado a técnica é acessível por permitir a digitalização 3D com equipamentos que cabem no orçamento dos usuários, por outro cobra dos mesmos um tempo que muitos podem não contar para preparar o cenário e proceder com fotografias, principalmente os profissionais que necessitam digitalizar muitas faces de uma vez. Diante dessa demanda os usuários têm inquirido constantemente acerca da possibilidade de utilizarem vídeos no lugar de fotografias, posto que os primeiros demandam um tempo menor para a aquisição das imagens necessárias para proceder com a digitalização.

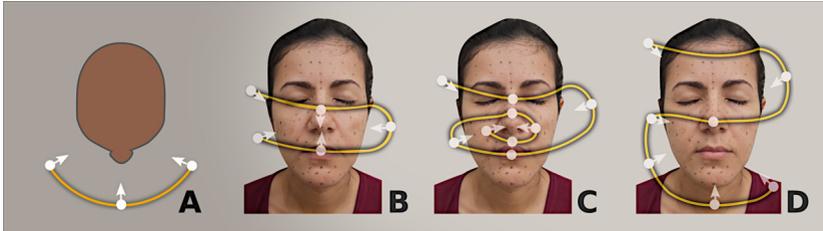
Sabe-se que sim, é possível utilizar vídeos no processo de fotogrametria e o motivo é muito claro, os vídeos são formados por quadros e os quadros podem ser convertidos em imagens que por sua vez são a base da fotogrametria. Mas o problema principal mora na qualidade dos quadros dos vídeos frente as fotografias, seja pela constante presença de quadros com borrão de movimento (*blur*) seja pelo formato do vídeo que, em grande parte ou é em HD ou em FullHD e ambos podem não ter qualidade suficiente para a digitalização 3D.

Felizmente os problemas supracitados podem ser resolvidos ou abrandados das seguintes formas:

- Para resolver o problema do *blur* basta mover a câmera lentamente ao capturar o objeto;
- Para resolver o problema da resolução, basta selecionar o formato 4K ou UHD (3840x2160px).

Com as bases do estudo definidas é possível iniciar o processo de captura e digitalização.

8.2 Captura dos Vídeos



Figural: Esquema gráfico das capturas de vídeo utilizadas no capítulo.

Foram efetuadas três capturas de vídeo para os experimentos abordados no presente capítulo, todas elas envolveram o deslocamento em arco (Fig. 1, A) disponível amplamente na documentação do *OrtogOnBlender* em especial no **Protocolo de Fotogrametria da Face [MdRD]**. Em duas capturas também executou-se a tomada fotográfica de modo a comparar a qualidade das imagens adquiridas por foto, face aquelas resultantes da conversão de quadros para imagens.

A primeira captura foi a mais simples e consistiu em percorrer o caminho do protocolo básico de 26 fotografias em duas alturas buscando a digitalização geral do rosto (Fig 1, B).

A segunda captura consistiu em percorrer o mesmo caminho que resultou na tomada de 63 fotografias (Fig. 1, C). Neste processo manteve-se as duas alturas gerais da face, complementando-as com uma captura mais próxima da região do nariz, algo muito compatível com o “Protocolo Face Duplo 90° + Nariz Duplo 170°” disponível no capítulo **Protocolo Complementar para Melhor Resolução do Nariz em Fotogrametria 3D [MSD+]**.

A última captura foi feita em três alturas (Fig. 1, D), desta vez apenas a versão em vídeo e embora o arco inicial tenha sido feito em uma altura maior a sequência assemelha-se muito ao “Protocolo Face Triplo 90°” disponível no capítulo citado no parágrafo anterior.

Importante: Em todas as capturas o dispositivo mantém-se direcionado para a face, como indicado nas setas. Os autores recomendam a leitura dos capítulos acima citados para que a compreensão acerca da técnica se faça mais clara para o usuário.

8.3 Ferramenta para a Conversão Direta de Vídeo em 3D

A partir da versão 2021-02-14 do OrtogOnBlender os usuários têm à sua disposição a ferramenta **Video to 3D**, presente na aba **Other**.

Segue o passo-a-passo do seu funcionamento:

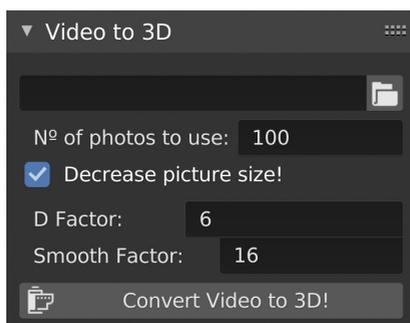


Figura2: Interface da ferramenta de digitalização 3D por vídeos.

1. Ao expandir a seção clique no ícone do diretório e selecione o arquivo desejado, o formato pode ser qualquer um que seja compatível com o FFMPEG, são eles: avi, mpeg, mpg, webm, etc. (Fig. 2);
2. Em **Nº of photos to use** informe o número de fotos que serão utilizados para a geração do modelo, lembrando que quanto maior o número, maior será o tempo necessário para o processo;
3. Para uma melhor resolução do modelo o usuário é orientado a desselecionar a opção **Decrease picture size!** (por padrão vem

selecionada) de modo a explorar mais a definição das imagens originais;

4. Depois de tudo definido basta clicar em **Convert Video to 3D!** e aguardar até que a digitalização seja finalizada.

Aviso: Como se verá adiante, no caso da fotogrametria de faces o uso da imagem nas dimensões originais pode gerar modelos com irregularidades superficiais, de modo que a desseleção do **Decrease picture size!** é mais aconselhável na digitalização de objetos, construções, e afins, não em faces humanas. No entanto em último caso a forma da abordagem é uma escolha do usuário frente as suas necessidades.

8.4 Resultados

A digitalização dos modelos foram efetuadas em um computador de mesa munido de um processador Intel i9-9900K com 64 GB de memória RAM. Como o foco do experimento recaiu sobre a superfície e sabendo-se que o algoritmo padrão do OrtogOnBlender gera boas texturas, a avaliação e as imagens serão apresentadas no modo *solid* sem texturização, posto que a mesma costuma camuflar os defeitos gerados na malha.

8.4.1 Protocolo Padrão para Ortogonática

Visualmente a melhor digitalização foi aquela feita a partir das 26 fotografias (Fig. 3, C) sem redução, as linhas ficaram melhores, mais detalhadas e a superfície mais atenuada. Os detalhes das asas do nariz também evidenciaram-se melhor. A reconstrução que mais se assemelhou a supracitada em qualidade foi aquela feita a partir do vídeo com 100 imagens (Fig. 3, A).

Dica: Para saber qual foi a fonte (vídeo ou foto) e o número de imagens usadas no processo veja o gráfico de tempo gerado a partir das

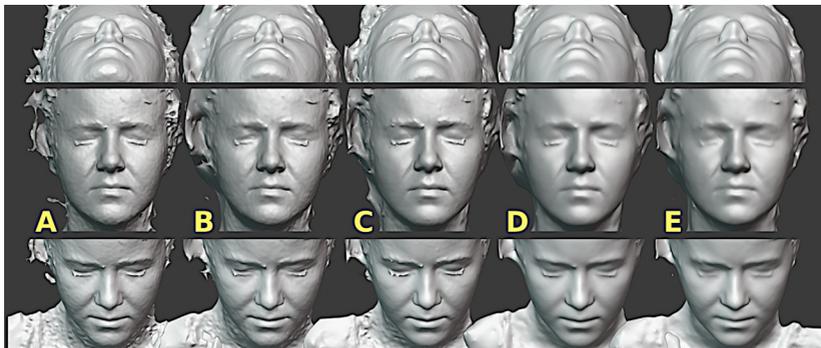


Figura3: Comparação entre as digitalizações com o protocolo padrão de ortognática.

digitalizações (Fig. 4) que conta com todos os dados compilados.

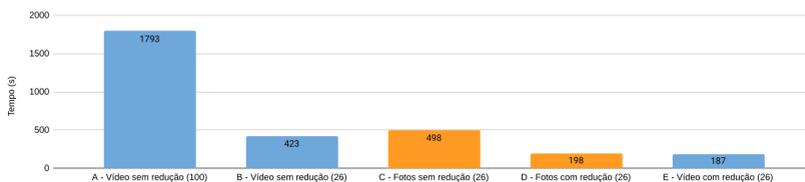


Figura4: Gráfico do tempo em segundos de cada digitalização com o protocolo padrão de fotogrametria.

Analisando o tempo de cada digitalização fica claro que o critério de escolha é feito pelo usuário em face das suas necessidades e tempo para executá-las. A digitalização que apresentou melhores resultados, ou seja, a “C - Fotos sem redução (26)” (Fig. 4), levou 498 segundos para ser efetuada, já a segunda colocada em qualidade a “A - Vídeo sem redução (100)” levou 1793 segundos, uma diferença muito grande e com menos qualidade na digitalização. Se o usuário optar pela praticidade da captura e contar com tempo, a escolha pela abordagem A parece ser a mais coerente. Se o usuário precisa de praticidade e conta com pouco tempo, ele pode optar pelo vídeo com 26 imagens (E), posto que o mesmo apresentou resultado muito semelhante aquele gerado pelas fotografias com redução. No entanto, se a necessidade recair sobre a qualidade da malha e o tempo, a melhor opção será sem dúvida a C.

Pelas imagens ficou evidenciado que as digitalizações com redução de imagem comprometeram a reconstrução de algumas áreas como a asa do nariz. Isso não é um problema relevante a ponto de inviabilizar o modelo para o planejamento de cirurgia ortognática, mas pode ser um ponto crítico no planejamento de rinoplastia.

O problema pode estar mais relacionado ao protocolo de captura do que ao fato das imagens serem fotos ou vídeos, por conta disso é necessário analisar os outros experimentos, buscando assim uma melhor compreensão.

Aviso: Como se trata de uma abordagem experimental o uso do lápis de olho foi ampliado ao longo de toda a face, de modo a garantir a digitalização da região. Mais testes serão necessários para avaliar a qualidade da digitalização pontilhando apenas a região do nariz, como é feito nos protocolos apresentados no OrtogOnLineMag e na documentação oficial do OrtogOnBlender.

8.4.2 Protocolo Face Duplo 90° + Nariz Duplo 170°

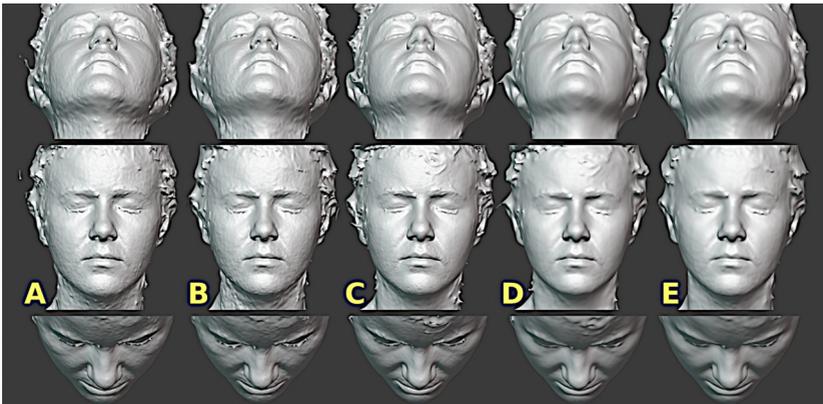


Figura5: Comparação entre as digitalizações com o protocolo padrão de ortognática.

Assim que a última faixa de digitalizações é visualizada na comparação de modelos (Fig. 5) já é possível verificar que o protocolo es-

colhido responde positivamente para a qualidade relacionada a regiões críticas como a do nariz, fazendo com que as digitalizações sem redução de imagens (A, B, C), gerem resultado muito semelhantes aquelas com redução (D,E), mas com a vantagem da segunda ser feita em um tempo significativamente reduzido. Fica clara então uma potencial vantagem de se utilizar a captura em vídeo, uma vez que a mesma é feita de modo mais rápido e prático que a fotografia e gera resultados muito semelhantes (D vs. E) aquela efetuada com fotografias quando a digitalização é feita a partir de imagens reduzidas. A fotogrametria gerada a partir de imagens reduzidas também resultam em modelos com a superfície mais atenuada do que a opção sem redução.

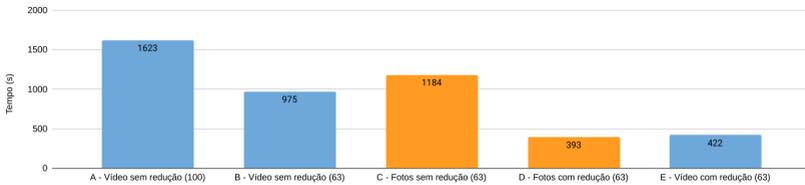


Figura6: Gráfico do tempo em segundos de cada digitalização com o Protocolo Face Duplo 90° + Nariz Duplo 170°.

A análise do gráfico de tempo reforça a vantagem das digitalizações efetuadas a partir de imagens reduzidas gerando modelos de boa qualidade em um tempo duas ou três vezes menor (Fig. 6).

8.4.3 Protocolo Face Triplo 90°

Como abordado anteriormente esta digitalização foi feita apenas com vídeo, dispensando a es equivalente por fotos (Fig. 7). O comportamento do processo se repetiu e a digitalização com 100 tomadas e sem a diminuição da imagem gerou um modelo com mais detalhes, porém com mais irregularidades ao longo da face (A), já o mesmo grupo de imagens (100) mas com redução, gerou um modelo com menos detalhes, mas sem as irregularidades da superfície. Mesmo assim os detalhes importantes como as asas do nariz estão presentes, o que faz o modelo potencialmente utilizável tanto para o planejamento de cirurgia ortognática, o que já era esperado, quanto para o

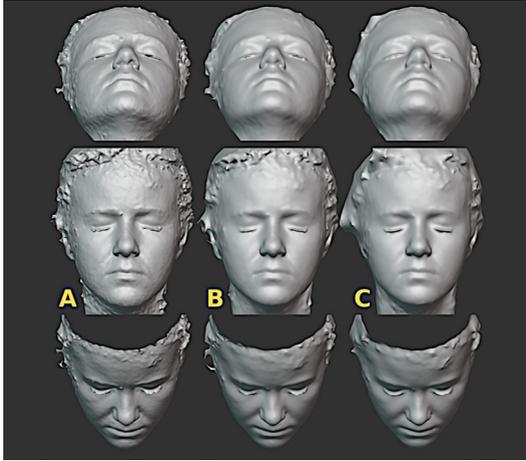


Figura7: Comparação entre as digitalizações com o Protocolo Face Triplo 90°

planejamento de rinoplastia.

Por último é apresentado o modelo digitalizado a partir de 39 tomadas com redução de tamanho (C) e o resultado não se distancia dos outros dois, ainda que os detalhes não sejam tão marcados. A vantagem mais clara é a atenuação da superfície, com menos irregularidades ao mesmo tempo em que mantém os detalhes das asas do nariz.

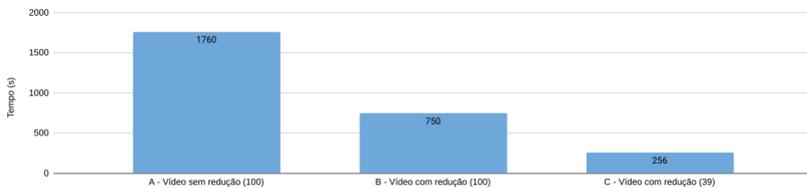


Figura8: Gráfico do tempo em segundos de cada digitalização com o Protocolo Face Triplo 90°.

A vantagem do modelo C se evidencia ainda mais quando comparados os tempos de digitalização, posto que o modelo resultante de 39 tomadas foi gerado em um tempo 2,93x mais rápido que o modelo B e 6,85x mais rápido que o modelo A!

8.5 Conclusão

Os experimentos mostraram que é possível digitalizar uma face a partir de um vídeo gravado em 4K. Os modelos gerados por imagens reduzidas foram os resultados com melhor custo/benefício, posto que geraram superfícies 3D atenuadas ao passo que conservavam os detalhes mais complexos do rosto.

As limitações deste experimento moram no reduzido número de digitalizações e no elevado zelo relacionado a segurança da captura, traduzido no incremento de pontos faciais que auxiliam o algoritmo de fotogrametria no reconhecimento de padrões nas imagens. Mais testes com mais indivíduos e com menos pontos faciais serão necessários para que a viabilização no uso prático seja implementada.

Digitalização 3D a partir de Vídeo de Drones Disponíveis na Internet

Moacir Elias Santos

Arqueólogo, Museu de Arqueologia Ciro Flamarion Cardoso, Ponta Grossa-PR

Cicero Moraes

3D Designer, Arc-Team Brazil, Sinop-MT

Luca Bezzi

Arqueólogo, Arc-Team, Cles-TN, Itália

Alessandro Bezzi

Arqueólogo, Arc-Team, Cles-TN, Itália

DOI: <https://doi.org/10.6084/m9.figshare.14115479>



O presente capítulo tem por objetivo discorrer acerca do uso de vídeos disponíveis na plataforma YouTube para a digitalização de cenas tridimensionais através da fotogrametria implementada no *add-on* OrtogOnBlender.

Atenção: Este material utiliza a seguinte licença Creative Commons: **Atribuição 4.0 Internacional (CC BY 4.0)**.

9.1 Apresentação

O uso de drones para a filmagem de grandes regiões tem se tornado muito comum nos últimos anos. Apenas no Brasil o número de equipamentos registrados na Agência Nacional de Aviação Civil (ANAC) saltou de 48.752 em 2018 para 71.561 em 2019 [Vie]. A realidade brasileira ainda está distante dos EUA, onde a FAA (Federal Aviation Administration) conta com 875.110 drones registrados [Adm].

A popularização da tecnologia se reflete em portais de vídeo como o YouTube onde é possível assistir a uma infinidade de materiais capturados com o auxílio de drones. Muitos geradores de conteúdo optaram por produzir vídeos voltados ao turismo, história e cultura, de modo que utilizam a flexibilidade de voo dos drones para capturar grandes monumentos, oferecendo vistas nunca antes exploradas dos mesmos a um custo acessível e com grande qualidade de imagem, sendo a maioria dos vídeos gravados em HD (1280x720px) e FullHD (1920x1080px) e uma parte considerável em 4K ou UHD (3840x2160px).

Por serem vídeos de movimento sequencial e em alta resolução, tornam-se candidatos perfeitos para a digitalização 3D através da fotogrametria, a técnica que utiliza imagens para gerar modelos tridimensionais.

Há uma série de ferramentas que permitem baixar os vídeos do YouTube em arquivos locais e o OrtogOnBlender oferece a opção de converter o vídeo em uma sequência de imagens e posteriormente transformar essas imagens em modelos 3D.

9.2 Digitalização da Necrópole de Gizé

9.2.1 Sobre a Necrópole

O planalto de Gizé forma uma área ampla, situada 24 quilômetros do Cairo, e mede aproximadamente 1,9 por 1,6 quilômetros. Este local, que é constituído por rocha calcária, foi denominado de Formação Mokattam, que na língua árabe significa “cortado” ou “interrompido”. A formação possui distintas camadas denominadas “Membro I, II e III” que possuem diferenças de qualidade da rocha (Lehner, p. 106) [LEH]. As camadas II e III podem ser reconhecidas na própria esfinge. Enquanto o corpo foi esculpido na camada Membro II, com maior fragilidade visto seu estado precário de conservação, a cabeça encontra-se na camada III, um calcário mais duro que permitiu a preservação das suas feições. Este calcário de melhor qualidade também foi explorado como material construtivo para as pirâmides e mastabas (tumbas retangulares em forma de banco), pois se confirmou a existência de pedreiras na área da necrópole. Além da extração da rocha calcária para as construções é importante lembrar que uma boa parte do planalto foi aplainado, notadamente na área que foi destinada às pirâmides, conforme se pode observar por meio do modelo 3D aqui apresentado.

Os três faraós responsáveis pela edificação das pirâmides e seus anexos, Khufu, Khafra e Menkaura (ou a versão mais conhecida originária do grego: Queóps, Quéfren e Miquerinos) pertencem à IV dinastia (c. 2575 – 2465 a.C.). A escolha da atual região de Gizé pelos faraós estava relacionada a uma questão religiosa. Consideradas como um local de ascensão e transformação para o reis, as pirâmides representavam um meio para uma nova existência. Seus nomes refletem esta questão, pois a pirâmide de Khufu era chamada de *Akhet Khufu*, ou o “Horizonte de Khufu”. A raiz da palavra *akh*, significa “brilhante” ou “radiante”, o que passa o entendimento de que é um lugar “radiante de glorificação” (Lehner, p. 24).

Tais estruturas foram construídas com uma notável orientação das faces em relação aos pontos cardeais, com as entradas posicionadas no lado norte, bem como um alinhamento que permite traçar uma reta a partir do canto sudeste das pirâmides. Outros alinhamentos também podem ser verificados, outrossim há hipóteses que foram

formuladas como a ligação com as estrelas, e que ainda geram debates entre os especialistas.

Ao final de seu processo construtivo as pirâmides foram revestidas com calcário polido, proveniente de Tura. Na atualidade a área melhor conservada pode ser observada na parte superior da pirâmide de Khafra, pois todo o restante foi removido e o material reutilizado para construções do Cairo medieval. No caso da pirâmide de Khufu, cuja altura original atingia 146,75 metros (Edwards, 1985, p. 104) [EDW], além do revestimento externo ocorreu também a remoção de uma parte do topo, bem como seu pyramidion. Isto ocasionou a diminuição de 9,45 metros da altura (Edwards, 1985, p. 104) e deixou uma área de aproximadamente 10 metros quadrados (Araújo, 1992, p. 89) [dA] plana. Já a pirâmide de Menkaura possui uma base feita com granito vermelho (Watson, 1987, p. 33) [WAT], que permanece no local, mas todo o revestimento calcário foi igualmente retirado.

Tanto a pirâmide de Khufu quanto a de Menkaura possuem três pirâmides menores destinadas às rainhas, em suas faces leste e sul respectivamente. Junto a pirâmide de Khufu, em sua face leste, há três poços naviformes. Já na face sul foram encontrados outros dois poços contendo os barcos de Khufu (Hawass, 2019, p. 111) [HAW]. A estrutura que aparece no modelo virtual (veja nas imagens 5 e 6) representa o museu que fica sobre o poço onde o barco foi encontrado em 1954 por Kamal el-Mallakh (Edwards, 1985, 115), e onde encontra-se atualmente exposto. Entre as pirâmides das rainhas foram encontrados dois poços naviformes, de menor tamanho. No canto sudeste da pirâmide de Khufu também foi localizada, por uma pesquisa conduzida por Hawass, a base e a infraestrutura de uma pirâmide satélite (Hawass, 2019, p. 115-116). Outra pirâmide semelhante está na face sul da pirâmide de Khafra. Junto as faces situadas no lado leste temos a presença dos templos das pirâmides, onde o culto funerário dos faraós era feito por sacerdotes. Estes templos estavam ligados a outros, situados no início da zona desértica, por meio de grandes calçadas. A esfinge (figura 3) está localizada ao lado da calçada, junto ao templo do vale de Khafra. A conservação deste templo é notável pela presença dos grandes blocos de granito em seu interior. Em frente à esfinge temos o templo de Hor-em-akhset (Hórus do Horizonte), única estrutura não funerária conhecida da IV dinastia. Podemos perceber os vestígios de praticamente todas estas estruturas no modelo virtual exceto por parte da calçada e o templo do

vale do complexo de Khufu que hoje se encontra sob a área urbana de Gizé, e parte da calçada e do templo do vale de Menkaura cujos vestígios são escassos (figura 9).

O conjunto de edificações é completado por uma grande quantidade de mastabas, tumbas baixas em forma de banco, que foram construídas como casas da eternidade da elite da IV dinastia. Tais tumbas foram situadas a oeste (destinadas aos oficiais e dignitários) e a leste (para os membros da família do faraó), dos muros que cercavam a pirâmide de Khufu, devidamente organizadas em fileiras paralelas. Ao sul da pirâmide há uma única fileira, disposta um pouco além dos poços com as barcas (figura 5 e 6). Posteriormente, já durante a V e VI dinastias, novas tumbas foram sendo construídas nos espaços entre as mastabas para abrigar sacerdotes funerários que, em vida, foram responsáveis pelo culto mortuário do rei (Edwards, 1985, p. 120).

9.2.2 Escolha do Material

Apesar da ampla oferta de vídeos nem sempre é uma tarefa fácil encontrar algum que seja compatível com a região que se pretende digitalizar em 3D, ainda mais quando se trata de uma vasta área como a Necrópole de Gizé. As opções em 4K são bastante variadas e cobrem parte considerável da região, mas houve dificuldades para a digitalização completa da pirâmide de Menkaura, talvez por ser a menor e logo, a menos notória entre as três grandes.

Outra região com escasso material próprio para digitalização é a Grande Esfinge de Gizé, neste caso, para complementar a região faltante compreendida pela cabeça e patas dianteiras foi necessário utilizar um vídeo capturado em um simulador de voo.

Os vídeos utilizados foram os seguintes:

- **Vídeo 1:** Pyramid Giza Egypt 4K Mavic pro drone 4K(part 1/2)⁶
- **Vídeo 2:** Pyramid Giza Egypt 4K mavic pro drone (part 2/2)⁷
- **Vídeo 3:** The Three Pyramids Giza in Egypt by drone dji MAVic pro⁸

⁶ https://www.youtube.com/watch?v=RtsCs_VfszE

⁷ <https://www.youtube.com/watch?v=Ilv6FYVZuVQ>

⁸ <https://www.youtube.com/watch?v=0Anj2fc6SrI>

- **Video 4** Pyramids - Giza, Egypt- by drone [4K]⁹
- **Video 5** Microsoft Flight Simulator 2020 Ancient Pyramids in Egypt Drone Mode¹⁰

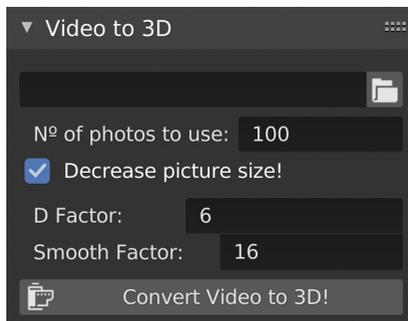
9.2.3 Digitalização da Cena

Existem duas maneiras de se digitalizar uma cena utilizando vídeos no OrtogOnBlender, uma convertendo o vídeo em uma sequência de imagens, selecionando os quadros desejados e rodando a fotogrametria e outra simplesmente indicando o vídeo desejado e ordenando a digitalização.

Há um material bem explanado acerca de como baixar o vídeo do YouTube e posteriormente convertê-lo em uma sequência de imagens, trata-se do capítulo da documentação oficial do OrtogOnBlender intitulado **Convertendo um Vídeo em um Arquivo DICOM [MdRDa]**. Apesar do mesmo ser orientado para a conversão em uma sequência de arquivos DICOM, a forma de conversão inicial é exatamente a mesma daquela utilizada para a fotogrametria.

A outra forma consiste em utilizar a ferramenta **Video to 3D** disponível na aba **Others**.

Segue o passo-a-passo do seu funcionamento:



Figural: Interface da ferramenta de digitalização 3D por vídeos.

⁹ <https://www.youtube.com/watch?v=u1S9p3IfCV0>

¹⁰ <https://www.youtube.com/watch?v=kQvIgvXE370>

1. Ao expandir a seção clique no ícone do diretório e selecione o arquivo desejado, o formato pode ser qualquer um que seja compatível com o FFMPEG, são eles: avi, mpeg, mpg, webm, etc. (Fig. 2);
2. Em **Nº of photos to use** informe o número de fotos que serão utilizados para a geração do modelo, lembrando que quanto maior o número, maior será o tempo necessário para o processo;
3. Para uma melhor resolução do modelo o usuário é orientado a desselecionar a opção **Decrease picture size!** (por padrão vem selecionada) de modo a explorar mais a definição das imagens originais;
4. O usuário é orientado a baixar os valores de D Factor e Smooth Factor para 4 e 4 ou 2 e 2 (por padrão é 6 e 16) de modo a gerar superfícies com mais resolução;
5. Depois de tudo definido basta clicar em **Convert Video to 3D!** e aguardar até que a digitalização seja finalizada.

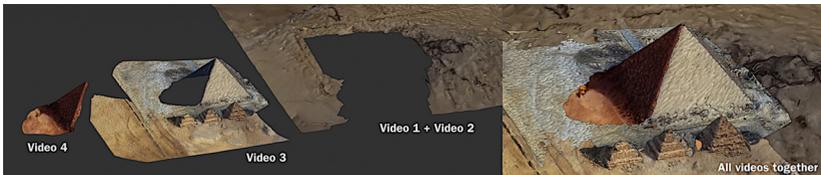


Figura2: Reconstrução de uma mesma região utilizando vídeos diferentes.

Como se trata de uma área muito grande foi necessário utilizar vários vídeos para que a digitalização fosse ampla. Para a região da pirâmide de Menkaura foi imprescindível uma complementação com vários vídeos. Cada um deles gerou um modelo e ao final todos foram alinhados de modo a reconstruir a pirâmide em questão. Esta técnica foi descrita detalhadamente no capítulo **Fotogrametria 3D de uma Ponta de Projétil Arqueológica** [SM].

O mesmo foi efetuado na região da Esfinge de Gizé e com um detalhe que denota bem a flexibilidade da fotogrametria, o vídeo utilizado foi de um voo efetuado no Microfost Flight Simulator, ou seja, não se tratava de um vídeo real, mas de um cenário modelado em 3D.



Figura3: Reconstrução da Esfinge de Gizé utilizando vídeos diferentes.

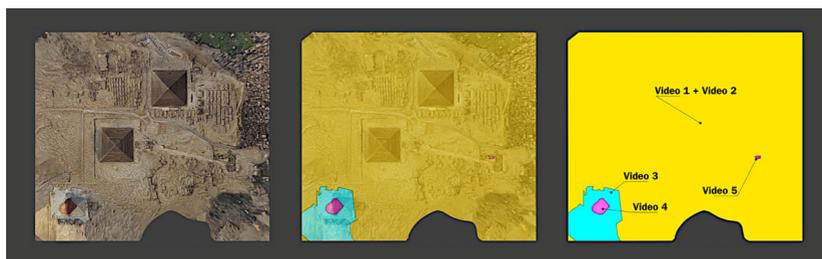


Figura4: Regiões correspondentes aos vídeos utilizados.

Em um mapeamento geral é possível visualizar quais vídeos foram responsáveis pelas regiões reconstruídas em 3D.

Para colocar o modelo na escala, recorreu-se a uma captura de imagem efetuada no [Google Maps](https://www.google.com.br/maps/@29.976387,31.132529,1261m/data=!3m1!1e3)¹¹. A referência de redimensionamento é fornecida por uma pequena escala presente na parte inferior da interface.

O resultado da digitalização a partir de vídeo 4K é surpreendentemente bom, a ponto de evidenciar detalhes do relevo da superfície. É importante lembrar que parte considerável dos quadros capturados em um vídeo apresentam borrão (blur) e isso em tese pode colaborar com uma má digitalização, posto que parte da imagem não fornece dados estruturais ao algoritmo. Felizmente essa característica pareceu não influenciar significativamente no escaneamento da estrutura.

Uma das vantagens de se digitalizar um monumento é poder gerar vistas ortogonais do mesmo, de modo a evitar qualquer tipo de magnificação gerado em vistas com deformação de perspectiva. To-

¹¹ <https://www.google.com.br/maps/@29.976387,31.132529,1261m/data=!3m1!1e3>



Figura5: Redimensionamento na escala, baseado nos dados do Google Maps.

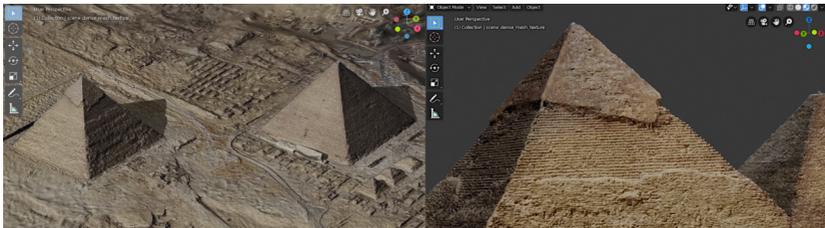


Figura6: Digitalização 3D, vista de duas pirâmides.



Figura7: Comparação ortográfica das construções. Da esquerda para a direita: Pirâmide de Khufu, Pirâmide de Khaфра, Pirâmide de Menkaura e Esfinge de Gizé.

mando todos os monumentos pelas vistas frontais e laterais é possível colocá-los lado a lado e comparar as dimensões, algo difícil de se fazer quando observadas as fotografias do local, justamente por conta da deformação de perspectiva.

Dica: Mais detalhes acerca da diferença entre a visualização com deformação de perspectiva e ortogonal podem ser vistos no capítulo **Princípios Básicos de Computação Gráfica 3D Aplicada às Ciências da Saúde** da documentação oficial do OrtogOnBlender [MdRDb], mais precisamente em **Formas de Visualização** (pág. 8-15).



Figura8: Modelos unidos com texturas heterogêneas vs. Modelo único com textura homogênea.

Se por um lado o alinhamento de várias digitalizações diferentes permite ao usuário montar uma cena com boa cobertura da superfície, por outro as diferenças na captação geralmente geram texturas com coloração diferentes, como visto na pirâmide de Menkaura. Para solucionar esse problema inicialmente é necessário unificar as várias malhas tridimensionais em apenas uma, no entanto não se trata de um simples agrupamento, mas a geração de uma malha limpa com superfície sem sobreposição de vértices e para isso o usuário pode recorrer ao gerador de malha não retentiva fornecido pelo OrtogOnBlender e cuja descrição do funcionamento está disponível no capítulo **Sistema de Geração de Malhas 3D Anti-Retenção Baseadas em Projeção Z-Depth** [MdRDc]. Para homogeneizar a coloração da malha, a forma mais acessível, uma vez que a superfície em questão é

não retentiva, é utilizar uma projeção de **UV Map**¹² do tipo *Project from View* pela vista superior e usar como base uma **imagem de satélite**¹³.

Aviso: A geração de modelo não retentivo se aplica bem a esse exemplo (necrópole com pirâmides), mas pode não ser eficiente em outras situações onde os modelos são mais complexos e com superfícies retentivas.

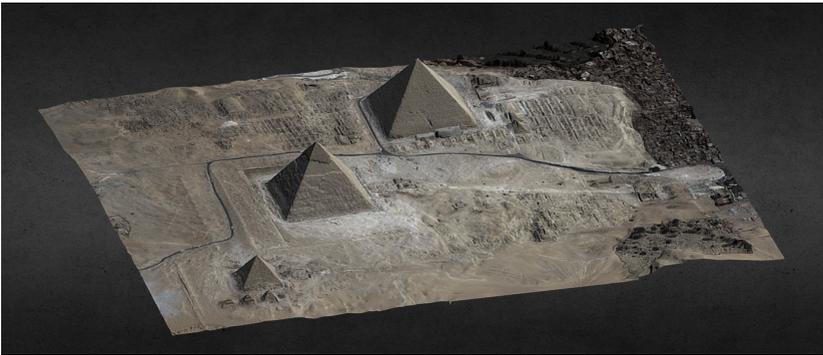


Figura9: Modelo interativo WebGL.

Uma vez que se tem um modelo limpo e homogeneizado há ainda a possibilidade de simplificar o mesmo e gerar uma versão para **visualização 3D interativa via WebGL**¹⁴ que pode ser visualizada tanto de um microcomputador, quanto de um *tablet* ou *smartphone*.

¹² <https://all3dp.com/2/blender-uv-mapping-simply-explained/>

¹³ <https://www.satimagingcorp.com/gallery/geoeye-1/geoeye-1-giza-pyramids/>

¹⁴ <https://skfb.ly/6YUSq>

9.3 Outros Exemplos

9.3.1 Cidade de Pedra de Uplistsikhe

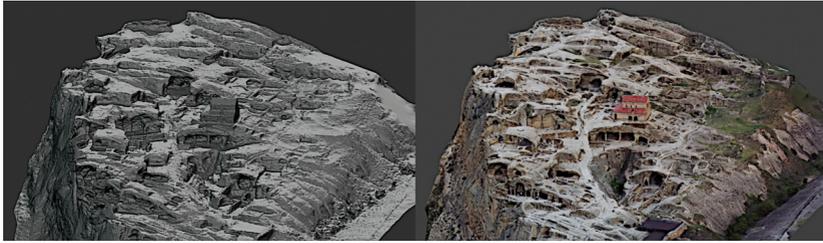


Figura10: Fotogrametria de Uplistsikhe.

A cidade de [Uplistsikhe](#)¹⁵ foi digitalizada utilizando como base apenas um vídeo disponível no YouTube sob o nome [Ouplistsikhé uplistsikhe drone mavic pro 4k](#)¹⁶.

O resultado do processo de digitalização pode ser visto em outro vídeo, [também disponível no YouTube](#)¹⁷.

Importante: É importante frisar que a heterogeneidade das cores e formas acaba por auxiliar o algoritmo de digitalização de modo que, se o mesmo fosse feito com uma estrutura monocromática e sem contrastes a chance de sucesso na digitalização seria pequena.

9.3.2 Stonehenge

A estrutura de Stonehenge também passou pelo mesmo processo de fotogrametria por vídeo, resultando em um [modelo interativo disponível de modo online](#)¹⁸.

¹⁵ <https://whc.unesco.org/en/tentativelists/5234/>

¹⁶ <https://www.youtube.com/watch?v=0zqfvRetHlo>

¹⁷ <https://www.youtube.com/watch?v=9pSZua68mMM>

¹⁸ <https://skfb.ly/6RnUy>



Figural1: Fotogrametria de Stonehenge.

9.4 Conclusão

A presença de vídeos produzidos por drones em sítios arqueológicos e sua disponibilidade em plataformas como o YouTube possibilita a produção de modelos tridimensionais por meio da fotogrametria. Neste capítulo, desenvolvemos um estudo de caso a partir do planalto de Gizé onde estão localizadas as pirâmides de Khufu, Khafra e Menkaura, os templos funerários, as mastabas, a esfinge, entre outras estruturas. Para tanto foram selecionados cinco vídeos disponíveis no YouTube e a partir da conversão destes em imagens, foi possível criar um modelo 3D do planalto de Gizé. Uma vez alinhados os dados obtidos dos vídeos, a escala foi então regulada a partir de uma imagem do Google Maps. A técnica aplicada proporcionou a geração de vistas ortogonais dos monumentos, sem a deformação da perspectiva.

A criação de modelos como o planalto de Gizé permite uma interatividade maior com a área para fins de divulgação científica, ao contrário das plantas baixas existentes encontradas em publicações especializadas ou na internet. Neste sentido, as estruturas e suas dimensões podem ser facilmente reconhecidas. Neste caso também é possível posicionar as estruturas maiores lado a lado para uma comparação das dimensões. Percorrer o modelo virtualmente, observando diferentes ângulos, também auxilia na compreensão de como o espaço do planalto de Gizé foi se transformando pelo crescente número de monumentos adicionados ao longo da IV dinastia e mesmo de épocas posteriores.

O modelo criado também permite a confecção de maquetes impressas, seja do atual estado de conservação da área ou mesmo com a restauração dos monumentos, que podem servir para museus e mesmo para a sala de aula, possibilitando o acesso ao conhecimento para o público em geral ou para estudantes. Por fim, a técnica aplicada no planalto de Gizé pode ser empregada para qualquer sítio arqueológico. Desta forma não só auxilia no registro de informações detalhadas sobre o estado atual de conservação de monumentos, mas também possibilita o registro contínuo de escavações arqueológicas assegurando a preservação de dados para pesquisas e restaurações futuras.

Conversão de Malhas 3D em Arquivos DICOM com o OrtogOnBlender

Cicero Moraes

3D Designer, Arc-Team Brazil, Sinop-MT

Rodrigo Dornelles

Cirurgião Plástico, Núcleo de Plástica Avançada - NPA, São Paulo-SP

Everton da Rosa

Cirurgião BMF, Hospital de Base, Brasília-DF

Johari Yap Abdullah

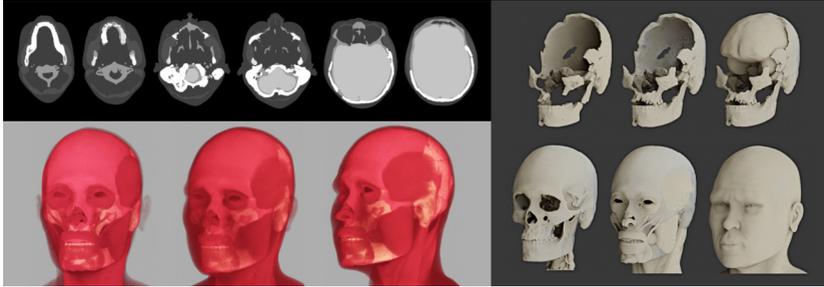
*Craniofacial Imaging Laboratory, School of Dental Sciences,
Universiti Sains Malaysia, Kelantan - Malásia*

Jafri Malin Abdullah

*Department of Neurosciences - School of Medical Sciences, Brain
and Behaviour Cluster - School of Medical Sciences, Department of
Neurosciences and Brain Behaviour Cluster - Hospital Universiti
Sains Malaysia, Universiti Sains Malaysia, Kelantan - Malásia*

DOI: <https://doi.org/10.6084/m9.figshare.14125706>

Atenção: Este material utiliza a seguinte licença Creative Commons: **Atribuição 4.0 Internacional (CC BY 4.0)**.



10.1 Apresentação

Se a conversão de uma série de arquivos DICOM em um modelo 3D é algo relativamente fácil de se fazer com inúmeras opções de aplicativos que oferecem essa possibilidade, já o mesmo não se pode dizer acerca da conversão contrária, ou seja, de um modelo 3D para uma série de arquivos DICOM.

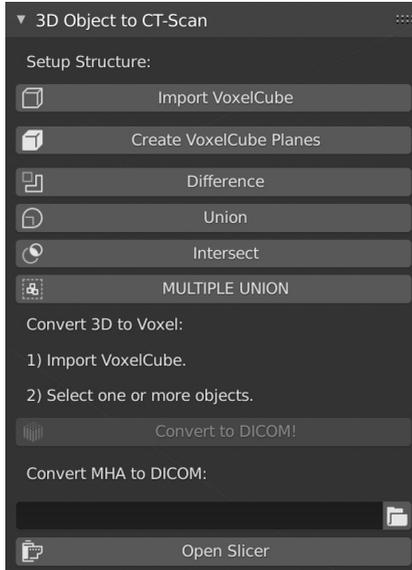
Ainda que em um primeiro momento não pareça tão óbvio, há diversas aplicações para essa segunda solução, uma delas é a documentação 3D de objetos fechados, próprios para a impressão 3D. Outra possibilidade é a interação com programas que utilizam arquivos DICOM para a projeção de próteses, algo bastante comum na ortodontia. Para este capítulo os autores criaram uma tomografia *fake* a partir de uma reconstrução facial forense de modo a apresentar as potencialidades da ferramenta, mas a mesma abordagem pode ser utilizada para a criação de modelos didáticos, tanto para ensino de anatomia visual, quanto para outras abordagens como o planejamento cirúrgico digital.

10.2 Como Funciona

10.2.1 Ferramenta 3D Object to CT-Scan

O conjunto de ferramentas 3D Object to CT-Scan se encontra disponível na aba **Others** ao lado direito da 3D View.

Seguem os passos para a conversão de uma malha 3D em uma sequência de arquivos DICOM:



Figural: Interface para a conversão de malhas 3D em arquivos DICOM.

1. Uma vez que o usuário tenha o modelo no qual pretende converter em DICOM já definido, é necessário importar um cubo com os limites espaciais da tomografia que será gerada, essa estrutura tem as dimensões de 300x300x300mm e, por ora, é a única opção disponível sendo acessada no botão **Import VoxelCube** (Fig. 1);
2. Os objetos que serão convertidos em tomografia, precisam estar acomodados dentro do cubo importado, caso contrário os arquivos gerados conterão erros dimensionais. Para que o usuário possa limitar a região disponível para geração de fatias, há a opção da criação de um cubo com as dimensões dos limites (300x300x300mm) que permitirá o cálculo booleano, caso seja necessário. Se o objeto em questão não exceder os limites disponíveis, não há a necessidade de utilizar essa ferramenta, mas caso seja necessário, basta clicar em **Create VoxelCube Planes** que o cubo para boolean será criado;
3. Caso o usuário precise, ele tem à sua disposição uma série de botões para cálculos booleanos: **Difference**, **Union**, **Intersect** e **MULTIPLE UNION**;

4. Uma vez que os limites estão estipulados e o modelo se encontra dentro deles, é necessário que o usuário selecione o, ou os objetos (se estiverem separados sem intersecção) que serão convertidos em malha e clicar em **Convert to DICOM!**;
5. O DICOM gerado no botão supracitado pode conter uma série de incompatibilidades com o OrtogOnBlender e outros editores de tomografias, de modo que será automaticamente aberto o software VolView e o arquivo apresentado no mesmo, para que lá seja exportado como um volume .MHA;
6. Assim que o volume for salvo, o usuário retorna ao OrtogOnBlender, seleciona o arquivo .MHA no seletor de arquivos e em seguida clica no botão **Open Slicer** para que naquele software o arquivo .MHA possa ser exportado como DICOM e o processo então finalizado.

Dica: Caso haja dúvidas em relação ao salvamento do arquivo no formato .MHA, há uma explanação disponível no capítulo **Convertendo um Vídeo em um Arquivo DICOM**, mais precisamente nas páginas 276 e 277 [OOBDMHA2020]. Se a dúvida estiver relacionada a exportação como DICOM no Slicer, a abordagem está disponível no capítulo **Protocolo de Tomografia para Planejamento de Cirurgia Ortognática**, mais precisamente nas páginas 237 e 238 [MdRDa].

Importante: A conversão de malha 3D em DICOM dentro do OrtogOnBlender só é possível em grande parte, graças ao trabalho do desenvolvedor **Christian Pederkoff** e a sua ferramenta `stl-to-voxel`²².

²² <https://github.com/cpederkoff/stl-to-voxel>

10.2.2 Como Configurar a Malha 3D para a Conversão em DICOM

Os arquivos DICOM fornecem dados de volume, já as malhas 3D são compostas por superfícies com lado (*normal*), mas sem espessura. É muito importante configurar a estrutura corretamente antes de fazer a conversão, sob pena de não gerar um modelo com a volumetria correta.

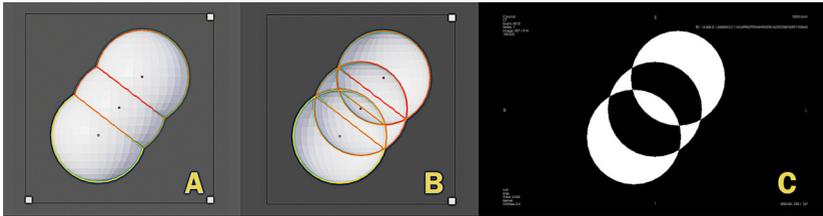


Figura2: Problema na geração do volume por conta das intersecções internas.

A imagem (Fig. 2) mostra o VoxelCube com três esferas alinhadas dentro do mesmo (A). Se for feito um corte do tipo “clipping” é possível ver que as esferas se transpassam em intersecção (B). Olhando por fora isso não parece ser um problema, mas a DICOM resultante a partir dos três modelos em intersecção gera um volume com áreas vazias dentro da estrutura (C), o que não era inicialmente esperado, posto que, ao se observar por fora a impressão que se tem é que se trata de um modelo sólido.

Caso o usuário tenha em mente a criação de um volume “cheio” a simples intersecção das esferas não o fornecerá, sendo então necessário um tratamento na malha para adequá-la a situação esperada.

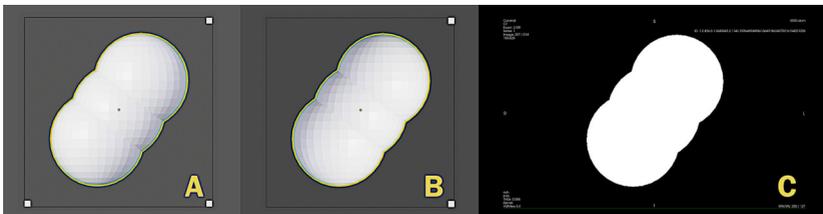


Figura3: Modelo sem intersecções internas.

As três esferas receberam o comando **MULTIPLE UNION** resultando em um único objeto (Fig. 3, A). Ao ser observado com um corte (*clipping*) a estrutura internão não apresenta intersecções (B). O DICOM resultante será um volume “cheio” (C), sem áreas vazias como a presente no *voxel* composto pelas três esferas em intersecção.

10.3 A Tomografia Fake do Homem de Perak

O Homem de Perak é o nome que se dá ao fóssil de um indivíduo adulto do sexo masculino encontrado no início da década de 1990 no Vale Lenggog, Malásia. Segundo levantamento os restos datam de 10.000-11.000 anos antes do presente e depois de um estudo efetuado por uma equipe multidisciplinar, composta parcialmente por autores deste capítulo, foi possível proceder com a reconstrução facial forense do fóssil em questão [Perak2021].

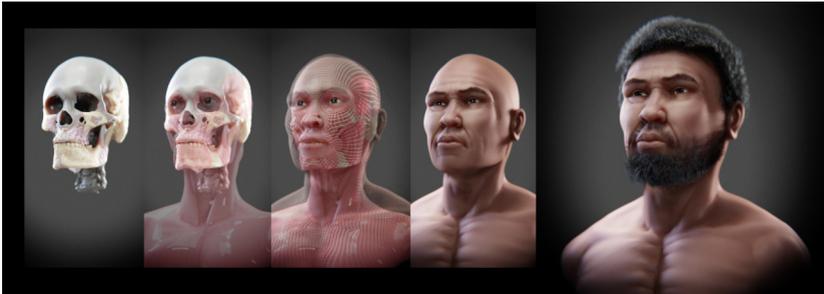


Figura4: Etapas da Reconstrução facial do Homem de Perak.

A reconstrução facial foi dividida em várias etapas (Fig. 4), iniciando pela reconstrução do crânio que originalmente se encontra fragmentado, passando pela modelagem dos músculos principais, em seguida a modelagem do busto e por fim a pigmentação e colocação dos pêlos e dos cabelos.

O objetivo do trabalho de reconstrução facial é apresentar principalmente os aspectos externos da estrutura, de modo que no geral, não há uma preocupação em criar objetos 3D que sofram intersecções internas como aconteceu na reconstrução dos ossos e principalmente dos músculos principais.

O OrtogOnBlender permite a geração de apenas um objeto em um grupo de fatias por vez, não é possível ainda selecionar mais de uma malha e indicar densidades diferentes. Além do mais, por padrão o *add-on* já devolve o arquivo DICOM como saída, no entanto é possível “burlar” o sistema e capturar os arquivos PNG que originaram os DICOMs finais. Esses arquivos tem apenas duas cores: preto e branco.

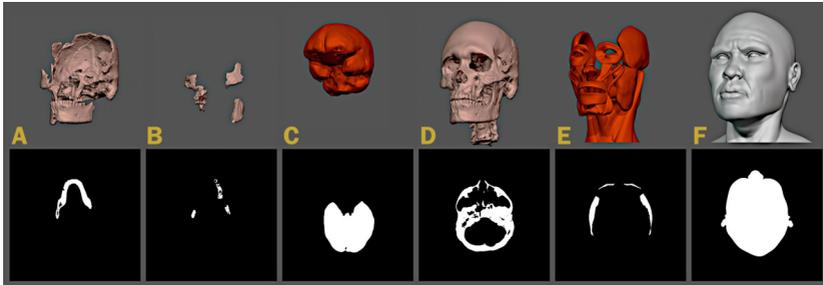


Figura5: Malhas unidas (acima) e fatias correspondentes (abaixo).

A geração dos grupos seguiu a composição original da reconstrução facial, os ssos originais remontados (Fig. 5, A), os ossos originais refletidos para complemento de regiões fantantes (B), o negativo correspondente ao cérebro (C), o crânio final composto com os ossos originais e um doador virtual (D), os músculos principais (E) e os demais tecidos moles (F).

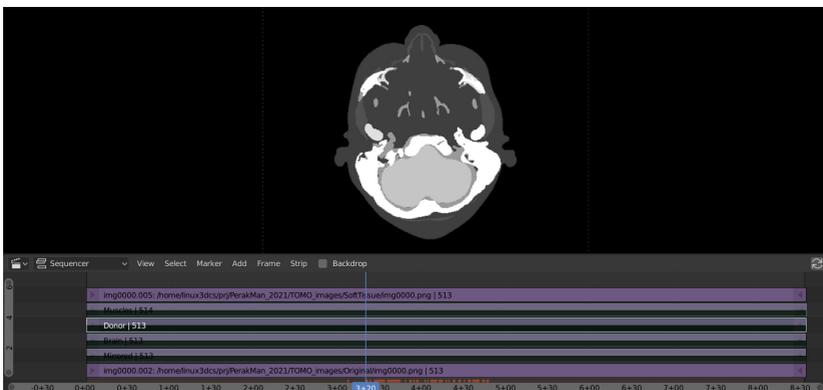


Figura6: Sobreposição de imagens no editor de vídeo do Blender.

O Blender conta com um editor de vídeo disponível em sua interface, dentre vários efeitos possíveis o usuário pode fazer a sobreposição de imagens (Fig. 6) e gerar uma saída que não obrigatoriamente é um vídeo, mas um conjunto de arquivos JPG, PNG ou afins.

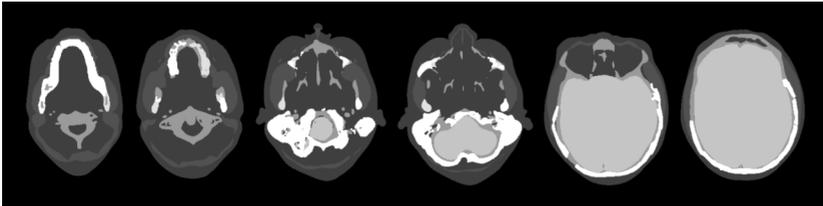


Figura7: Amostra das fatias finais geradas pela composição de imagens.

Cada camada sobreposta recebeu um valor de transparência permitindo que os tecidos fossem destacados pela intensidade da escala de cinza do *pixel* em questão. Grosso modo, o osso ficou branco, o ar ficou preto, o tecido mole um pouco mais claro que o ar, os músculos um pouco mais claros do que o tecido mole, o cérebro um pouco mais claro do que os músculos e assim por diante (Fig. 7).

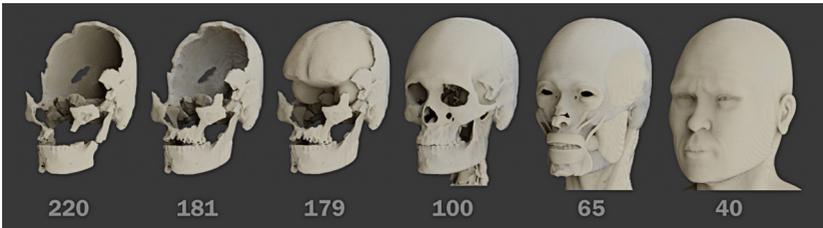


Figura8: Visualização do voxel data pelo valor do threshold.

Ao se importar o *voxel data* no OrtogOnBlender pode-se reconstruir a região de interesse a partir do valor do *threshold* (Fig. 8), onde o valor mais alto (220) reconstrói os ossos, com a maior densidade e quanto mais se baixa o valor, mais se vê os demais tecidos até chegar à pele (40). Como houve uma sobreposição de *pixels* que se somaram, não é mais possível isolar algumas das estruturas que originalmente eram totalmente separadas, como os ossos espelhados ou o cérebro.

Também é possível trabalhar os *voxels* com cores e opacidade diferentes (Fig. 9). Isso permite ao usuário contemplar uma série de te-

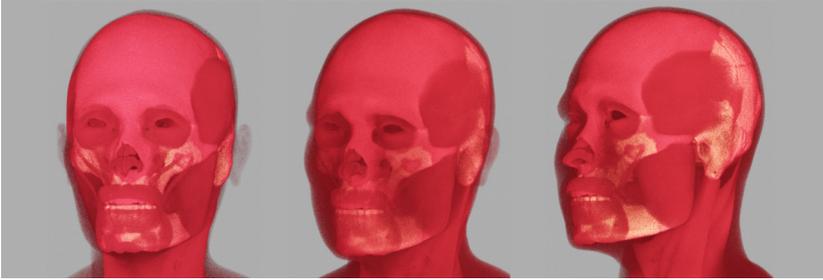


Figura9: Visualização do voxel data com diferença de cores.

cidos distintos de uma só vez. Além desta opção ainda se pode contemplar a estrutura interna do *voxel*, utilizando os cálculos booleanos, como foi abordado no capítulo *Reconstrução de Voxel Data no OrtogOnBlender*.

10.4 Conclusão

A ferramenta de conversão de malha 3D em arquivos DICOM mostra mais uma vez a flexibilidade do OrtogOnBlender, ao oferecer ao usuário uma solução que o permitirá resolver um problema específico e ao mesmo tempo pleno de potencialidades gerais. Com essa ferramenta é possível criar modelos didáticos como o apresentado neste capítulo, gerar volumes fechados e “sólidos” para posterior impressão 3D, manter uma linha de interoperabilidade mais efetiva com o setor de radiologia, trabalhar com aplicativos de odontologia que usam o formato DICOM e muitas outras possibilidades que ainda não foram levantadas, mas que certamente serão abordadas por usuários e leitores após a publicação do capítulo.

A ferramenta obviamente apresenta uma série de limitações, como apenas uma dimensão (300x300x300mm) de matriz, bem como um número limitado de níveis relacionados a escala de cinza (0 a 255). Espera-se que com a procura e discussões junto aos usuários e novos interessados, os desenvolvedores possam solucionar as limitações e expandir o uso com novas e importantes abordagens.

Referências Bibliográficas

- [Bun19] Sabiha Bunek. What's new in digital dentistry? 2019. Dental Advisor. SEPT-OCT, DENTAL Vol. 36, No. 05.
- [eal18] NEAL D. KRAVITZ et al. Cad/cam software for three-dimensional printing. 2018. JCO/January.
- [MGF20] Renato Parsekian Martins, Fabio Guedes, and Bruno D'Aurea Furquim. Alinhadores e ortodontia digital. 2020.
- [MdRD20] Cicero Moraes, Everton da Rosa, and Rodrigo Dornelles. *OrtogOnBlender - Documentação Oficial vol. 3*. Cicero André da Costa Moraes, 2020. ISBN 9786500053494. doi:10.6084/m9.figshare.12871730.v1¹.
- [Rog10] Everett M. Rogers. *Diffusion of Innovations*. Free Press, 2010. ISBN 9781451602470.
- [dAdA94] NS de Araújo and VC de Araújo. Patologia bucal. 1994. 1.ed. São Paulo: Artes médicas.
- [dFreS00] A de Freitas, JE Rosa, and IF e Sousa. Radiologia odontológica. 2000. 5.ed. São Paulo: Artes Médicas, 2000. p.386-91; 431-33; 468-9.
- [Doc20a] 3D Slicer Documentation. Growcutsegmentation. 2020. URL: <https://www.slicer.org/wiki/Modules:GrowCutSegmentation-Documentation-3.6>.
- [Doc20b] 3D Slicer Documentation. Modelmaker. 2020. URL: <https://www.slicer.org/wiki/Modules:GrowCutSegmentation-Documentation-3.6>.
- [LL98] MR Leonardo and JM LEAL. Endodontia- tratamento de canais radiculares. 1998. 3.ed. São Paulo: Editorial Médica Panamericana.

¹ <https://doi.org/10.6084/m9.figshare.12871730.v1>

- [LKL99] J Lindhe, T Karring, and NP Lang. Tratado de periodontia clínica e implantologia oral. 1999. 3.ed. Rio de Janeiro: Guanabara Koogan. p.263-66.
- [MdRD20] Cicero Moraes, Everton da Rosa, and Rodrigo Dornelles. *OrtogOnBlender - Documentação Oficial vol. 3*. Cicero André da Costa Moraes, 2020. ISBN 9786500053494. doi:10.6084/m9.figshare.12871730.v1².
- [RS00] JA Regezi and JJ Sciubba. Patologia bucal- correlações clinicopatológicas. 2000. 3.ed. Rio de Janeiro: Guanabara Koogan.
- [SMB87] WG Shafer, MK Hine MK, and Levy BM. Tratado de patologia bucal. 1987. 4.ed. Rio de Janeiro: Guanabara Koogan.
- [Sli20] 3D Slicer. 3d slicer - a multi-platform, free and open source software package for visualization and medical image computing. 2020. URL: <https://www.slicer.org/>.
- [Tom02] AF Tommasi. Diagnóstico em patologia bucal. 2002. 3.ed. São Paulo: Pancast, p.285-91.
- [Cer20] Dan Cernea. OpenMVS: multi-view stereo reconstruction library. 2020. URL: <https://cdcseacave.github.io/openMVS>.
- [dR] Presidência da República. Lei nº 8.078, de 11 de setembro de 1990. URL: http://www.planalto.gov.br/ccivil_03/leis/18078compilado.htm.
- [Gog] Ronaldo Gogoni. O que é hackintosh? [3 motivos para fazer]. URL: <https://tecnoblog.net/302675/o-que-e-hackintosh-3-motivos-para-fazer/>.
- [Mic] Microsoft. Baixar imagem de disco do windows 10 (arquivo iso). URL: <https://www.microsoft.com/pt-br/software-download/windows10ISO>.
- [Mic20] Microsoft. Io que é o subsistema do windows para linux? 2020. URL: <https://docs.microsoft.com/pt-br/windows/wsl/about>.

² <https://doi.org/10.6084/m9.figshare.12871730.v1>

- [MdRD20a] Cicero Moraes, Everton da Rosa, and Rodrigo Dornelles. Como instalar o linux3dcs em um ssd externo. 2020. URL: http://www.ciceromoraes.com.br/doc/pt_br/OrtogOnBlender/Instalacao_Linux3DCS_SSD.html.
- [MdRD20b] Cicero Moraes, Everton da Rosa, and Rodrigo Dornelles. Instalando o linux no windows para rodar o smvs e o img2dcm. 2020. URL: <https://bit.ly/3Gk0wBQ>.
- [MdRD20c] Cicero Moraes, Everton da Rosa, and Rodrigo Dornelles. Protocolo simples de fotogrametria para crânios. 2020. URL: http://www.ciceromoraes.com.br/doc/pt_br/OrtogOnBlender/Fotogrametria_Cranio.html.
- [MdRD20d] Cicero Moraes, Everton da Rosa, and Rodrigo Dornelles. Protocolo de fotogrametria da face. 2020. URL: http://www.ciceromoraes.com.br/doc/pt_br/OrtogOnBlender/Fotogrametria_Face.html.
- [MdRD20e] Cicero Moraes, Everton da Rosa, and Rodrigo Dornelles. Resultados da digitalização. 2020. URL: http://ortogonline.com/doc/pt_br/OrtogOnLineMag/1/Nariz.html#resultados-da-digitalizacao.
- [MMPM16] Pierre Moulon, Pascal Monasse, Romuald Perrot, and Renaud Marlet. Openmvg: open multiple view geometry. In *International Workshop on Reproducible Research in Pattern Recognition*, 60–74. Springer, 2016.
- [Ubu] Ubuntu. Installation/netboot. URL: <https://help.ubuntu.com/community/Installation/Netboot>.
- [Sch] Adrian Schneider. Dicomtomes. URL: <https://github.com/AOT-AG/DicomToMesh>.
- [Sli] Slicer. Slicer. URL: <https://www.slicer.org/>.
- [dD] Universidade de Darmstad. Multi-view environment. URL: <https://github.com/simonfuhrmann/mve>.
- [LSHG16] F. Langguth, K. Sunkavalli, S. Hadap, and M. Goesele. Shading-aware multi-view stereo. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 2016.

- [MdRD20] Cicero Moraes, Everton da Rosa, and Rodrigo Dornelles. *Fotogrametria na Digitalização 3D de Faces - Comparação entre Ferramentas*. Volume 1. Cicero André da Costa Moraes, 2020. ISBN 978-65-00-05349-4. p. 119-200.
- [WMG14] Michael Waechter, Nils Moehrle, and Michael Goesele. Let there be color! – Large-scale texturing of 3D reconstructions. In *Proceedings of the European Conference on Computer Vision*. Springer, 2014.
- [Foua] Blender Foundation. Blender 2.91 manual - ambient occlusion. URL: https://docs.blender.org/manual/en/2.79/render/blender_render/world/ambient_occlusion.html.
- [Foub] Blender Foundation. Blender 2.91 manual - material. URL: <https://docs.blender.org/manual/en/latest/render/materials/introduction.html>.
- [MdRD20] Cicero Moraes, Everton da Rosa, and Rodrigo Dornelles. *OrtogOnBlender - O que é e Aspectos Técnicos*. Volume 1. Cicero André da Costa Moraes, 2020. ISBN 978-65-00-05349-4. p. 24-26.
- [Sli] Slicer. Slicer. URL: <https://www.slicer.org/>.
- [MdRD] Cicero Moraes, Everton da Rosa, and Rodrigo Dornelles. Protocolo de fotogrametria da face. URL: https://www.researchgate.net/publication/342610887_Protocolo_de_Fotogrametria_da_Face, doi:10.6084/m9.figshare.12923801.v1⁴.
- [MSD+] Cicero Moraes, Davi Sandes Sobral, Daniele Walter Duarte, Gabriel Zorron Cavalcanti, Rodrigo Salazar-Gamarra, and Rodrigo Dornelles. Protocolo complementar para melhor resolução do nariz em fotogrametria 3d. URL: https://www.researchgate.net/publication/344380790_Protocolo_Complementar_para_Melhor_Resolucao_do_Nariz_em_Fotogrametria_3D, doi:10.6084/m9.figshare.13010300⁵.

⁴ <https://doi.org/10.6084/m9.figshare.12923801.v1>

⁵ <https://doi.org/10.6084/m9.figshare.13010300>

- [Adm] Federal Aviation Administration. Uas by the numbers. URL: https://www.faa.gov/uas/resources/by_the_numbers/.
- [dA] L. M. de ARAÚJO. Egípto: as pirâmides do império antigo. Lisboa : Edições Colibri, 1992.
- [EDW] I.E.S. EDWARDS. As pirâmides do egito. Rio de Janeiro: Record, 1985.
- [HAW] Z. HAWASS. Montanhas dos faraós: a história não contada dos construtores das pirâmides. Curitiba: AMORC, 2019.
- [LEH] M LEHNER. The complete pyramids. London: Thames & Hudson, 2000.
- [MdrDa] Cicero Moraes, Everton da Rosa, and Rodrigo Dornelles. Convertendo um vídeo em um arquivo dicom. URL: https://www.researchgate.net/publication/344141704_Convertendo_um_Video_em_um_Arquivo_DICOM, doi:10.6084/m9.figshare.12923822¹⁹.
- [MdrDb] Cicero Moraes, Everton da Rosa, and Rodrigo Dornelles. Princípios básicos de computação gráfica 3d aplicada às ciências da saúde. URL: https://www.researchgate.net/publication/341833430_Principios_Basicos_de_Computacao_Grafica_3D_Aplicada_as_Ciencias_da_Saude.
- [MdrDc] Cicero Moraes, Everton da Rosa, and Rodrigo Dornelles. Sistema de geração de malhas 3d anti-retenção baseadas em projeção z-depth. URL: https://www.researchgate.net/publication/343822666_Sistema_de_Geracao_de_Malhas_3D_Anti-Retencao_Baseadas_em_Projecao_Z-Depth, doi:10.6084/m9.figshare.12881231.v1²⁰.
- [SM] Antonio Carlos Soares and Cicero Moraes. Fotogrametria 3d de uma ponta de projétil arqueológica. URL: https://www.researchgate.net/publication/343980094_Fotogrametria_3D_de_uma_Ponta_de_Projetil_Arqueologica, doi:10.6084/m9.figshare.12894209²¹.

¹⁹ <https://doi.org/10.6084/m9.figshare.12923822>

²⁰ <https://doi.org/10.6084/m9.figshare.12881231.v1>

²¹ <https://doi.org/10.6084/m9.figshare.12894209>

- [Vie] Nathan Vieira. Número de drones cadastrados no brasil salta 55. URL: <https://bit.ly/3FiyvJN>.
- [WAT] P. WATSON. Egyptian pyramids and mastaba tombs. Ay-lesbury: Shire Publications, 1987.