**UTILIZANDO FOTOS PARA CRIAR UM ROSTO 3D:**

Como reconhecimento facial ajuda na criação do modelo

Lucas Amaral Oliveira

Graduanda em Engenharia de Software – Uni-FACEF

lucas.amaral.olivera@gmail.com

Nathan Mendes Maturano

Graduanda em Engenharia de Software – Uni-FACEF

nathanmedesn2m@gmail.com

Daniel Facciolo Pires

Orientador – Uni-Facef

dfpires@gmail.com

**Resumo**

O reconhecimento facial tem como importância a definição dos pontos nodais para facilitar na modelagem 3D que por sua vez reconhece estes pontos e modela o avatar de uma forma mais exata. O trabalho tem como objetivo a modelagem de um avatar que poderá ser utilizado em diferentes aplicações, tais como jogos, filmes, dublês digitais. A metodologia utilizada para a execução da tarefa consiste na captura das imagens, após isso o protótipo segue para os procedimentos da modelagem, tais como a definição dos contornos do rosto, os pontos nodais, para depois partir para o avatar 3D. Os procedimentos atingiram como resultado um modelo 3D, porém em poucos polígonos. O projeto teve êxito em alcançar a modelagem 3D de forma simplificada e eficaz poupando uma quantidade de tempo dos usuários e os encorajando a utilizar a solução no dia a dia.

**Palavras-chave**: Modelagem 3D, Reconhecimento Facial, Visão Computacional, Pontos Nodais.

**Abstract**

*The prototype developed draws attention to the lack of 3d modeling systems with face recognition features. Besides the high cost and complexity that modern softwares presents. It also tackle the delay theta some products presents when generating the model. The prototype will be presented through the article. This had been developed in R language and utilize the according librarys. Only developers pictures were utilized to assure no use of anyone’s image impropely. The pictures was uzed with the intent of optimizing the modeling. With that in mind, the prototype aims at time optimization and the doesn’t require user’s monitoring during modeling.*

**Keywords**: *3D Modeling. Face Recognition. Computer Vision. Nodal Points.*

1. **Introdução**

Tendo em vista a crescente necessidade do uso do reconhecimento facial para diversos fins. O trabalho a seguir mostra uma prototipação do uso de Reconhecimento facial para gerar modelos em 3D. Para o uso da fotogrametria em modelagem 3D, foram utilizadas diversas tecnologias, que serão citadas mais adiante.

O sistema descrito neste artigo têm como objetivos, facilitar e popularizar a modelagem em 3D, gerando uma redução no custo da produção. Cada parte deste artigo mostrará uma parte da forma como o protótipo fora desenvolvido.

O protótipo da solução foi desenvolvido através do *RStudio*, tendo como base a foto de um dos desenvolvedores. Esperando o retorno de um modelo 3D partindo das fotos utilizadas, porém ocorreram alguns contratempos que serão citados posteriormente neste artigo.

Para um melhor entendimento do problema foi realizado uma pesquisa sobre as tecnologias que estão em atividade atualmente. Após a pesquisa foram definidas as estratégias de desenvolvimentos e testes do protótipo.

Com o avanço da tecnologia, vários processos foram facilitados. Chamar o táxi foi facilitado (*Uber*). Uma mensagem que demoraria semanas para ser recebida agora é enviada aos montes através de *e-mails*. Com a localização sendo rastreada em tempo real, mapas agora podem calcular fornecer a melhor rota rapidamente. É natural que até áreas recentes como a modelagem 3D.

Acreditamos que uma forma de otimizar esse último seria utilizando o rosto humano, ele tem um padrão de características únicas que o sistema pode reconhecer e gerar um modelo 3D da face.

Esse modelo pode ser utilizado como uma personagem em um jogo criado pelo usuário, um dublê digital para diminuir riscos ao gravar uma cena de ação para um filme, ter que gravar menos cenas e mascarar com mais facilidade quando utilizam um dublê humano para gravar uma cena.

Com um pouco mais de modificação, o usuário pode criar um avatar para outras aplicações, como *streamar* na *Twitch* de maneira anônima, ou criar um autorretrato estilizado em 3D, podendo até imprimir o mesmo e criar uma estátua do usuário ou um molde para manequim, ou ainda criar um *deepfake* mais realista em videos

E o usuário só precisaria de enviar algumas fotos e o sistema automaticamente geraria um modelo baseado nelas, o que o torna uma forma de modelagem 3D bastante acessível e fácil de usar numa era em que todo mundo tem um celular e um dos maiores diferencial deles é a qualidade da câmera.

1. **Referencias**

Um modelo tridimensional, também conhecido como modelo 3D, é uma representação digital de um objeto ou cena em três dimensões. Ou seja, é uma representação computacional que descreve um objeto em três eixos: altura, largura e profundidade (ROYRIGSS, 2011).

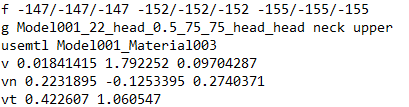
Os modelos 3D são amplamente utilizados em diversas áreas, como arquitetura, engenharia, design de produtos, animação, jogos, entre outras. Eles podem ser criados a partir de softwares de modelagem 3D, escaneamento 3D ou outras técnicas de captura de dados.

Uma das formas salvar um modelo 3D mais populares é o formato .obj. A extensão .obj é um formato de arquivo utilizado para armazenar modelos tridimensionais (3D) em computação gráfica. Esses arquivos geralmente contêm informações sobre a geometria, texturas, normais e outras propriedades de um modelo 3D (ROYRIGSS, 2011).

Os arquivos .obj foram criados em 1980 e atualmente podem ser abertos e editados em softwares de modelagem 3D, sendo suportados por uma grande variedade de programas de computador, tanto comerciais quanto gratuitos, como Blender (TON, 1994), Maya (ALIAS, 1998), 3ds Max (AUTODESK, 1996), entre outros. Eles também podem ser usados em jogos, animações, filmes e outras aplicações de computação gráfica.

Explicando em mais detalhes, .OBJ nada mais é que um dicionário interpretado pelo computador, como vemos na Figura 1, na qual o ‘v’ representa um vértice e sua posição; ‘vt’ representa a textura/a cor do vértice, ‘vn’ o mapa normal do vértice, que é basicamente como que aquele vértice deve reagir a luz, ‘g’ que representa o grupo, a malha na qual se encontram os vértices (um obj pode ter mais de um grupo) e ‘f’ representa uma face, que nesse caso seria um polígono triangular, utilizando os pontos -147 ~~,~~152 , -155 e suas respectivas texturas e normals para construi-lo.

**Figura 1** - arquivo .obj aberto no bloco de notas



**Fonte-** SEA206,2019

**2.1.Visão computacional**

Visão computacional é uma área de estudo da ciência da computação que lida com a interpretação de informações visuais a partir de dados de imagens ou vídeos. Utilizando de técnicas de geometria, transformação e processamento de imagens e estimação de movimento, ela consegue reconhecer objetos desde rostos numa foto a até montar um objeto num espaço 3d (SZELISKI,2011).

Isso por causa de padrões geométricos que a máquina nota nas imagens providas a ela. Razões como essa tornam a visão computacional um atrativo para trabalhar com segurança por meio da biometria, já que um bom sistema pode diferenciar rostos e dizer a quem ele pertence, utilizando 80 pontos nodais do rosto, pontos específicos da face humana, para medir e identificar diferenças, aplicando então o reconhecimento facial (ZHANG, 2017).

Reconhecimento facial é mais do que detecção facial. Detecção diz respeito apenas a dizer se tem um rosto naquela foto e onde está. Reconhecimento analisa os detalhes do rosto para determinar quem é. Isso significa que o reconhecimento pode ser atrapalhado pela qualidade da câmera, iluminação do rosto e orientação da cabeça, precisando que o usuário se ajuste para compensar.

Com múltiplas fotos em diferentes ângulos, é possível cancelar esse negativo e determinar com mais clareza as dimensões do sujeito das fotos. Aplicando essa escalabilidade num ambiente 3d e texturizando o modelo gerado é o que chamamos de fotogrametria. Ela não é aplicada muito em pessoas fora da área de medicina, sendo mais popular para determinar o tamanho de uma área geográfica (ADENILSON, 2018).

A definição de Fotogrametria a proposta pela *American Society of Photogrammetry* em 1979, como sendo: "*Fotogrametria é a arte, ciência e tecnologia de obtenção de informação confiável sobre objetos físicos ambiente através de processos de gravação, medição e interpretação de e o meio imagens fotográficas e padrões de energia eletromagnética radiante e outras fontes*".

Por suas características, essa técnica que originalmente foi criada para estudos geológicos já foi exportada para diversas áreas, como criação de Projetos de estradas, Arqueologia, Robótica, Digitalização de apartamentos para vendê-los, e até medicina, mas mesmo sendo ótima para construir réplicas virtuais didáticas de órgãos humanos, ela é péssima para criar uma pessoa 3d, pelo menos quando se utiliza poucas câmeras. Isso porque fotogrametria utiliza a semelhança das fotos para determinar o tamanho do objeto, mas com qualquer variação ou inconsistência causadas por respirar já pode desalinhar o processo.

**2.2.Aplicativos Similares**

Nessa seção serão abordadas as principais técnicas e ferramentas utilizadas pela concorrência, para realizar tarefas semelhantes

O *Scanner 3D* é um hardware que utiliza recursos de câmeras de alta resolução para captar imagens. Para auxiliar na captura mais precisa destas imagens podem ser utilizados recursos como luz estruturada e feixe de laser. Estas duas formas citadas anteriormente possuem a função de facilitar na captura e envio de dados em imagem (PRINTIT, 2021).

Um bom escâner pode poupar tempo de trabalhos indesejados com redimensionamentos ou correções indesejadas. Respeitando o versionamento e as normas de cada aparelho, o escaneamento obtido será mais preciso.

**2.2.1.Revopoint**

Revopoint é uma companhia focada em escaneamento 3d. Seu último produto, Revo RANGE, permite escanear objetos grandes de perto (até 800 milímetros) o que permite uma precisão melhor que outros *scanners* profissionais, especialmente para pessoas (REVOPOINT, 2023).

Ele é pequeno e possui softwares e compatibilidade com celular mão. Utiliza duas lentes para tirar fotos e dizer a distância dos objetos baseado nas diferenças entre as duas imagens.

Ainda sim, a única diferença que eles fazem para escanear pessoas é configurações da altura do objeto, nenhum processo de otimização extra para pessoas é feito, e precisa de um hardware extra para fazer.

**2.2.2.ZKTeco**

Outro *Scanner 3d*, O ZKTeco se diferencia do antecessor pela sua tecnologia de Luz Estruturada, que permite aplicar o reconhecimento facial em 3D. Essa tecnologia permite que o brilho do ambiente e outros fatores externos sejam desconsiderados pelo sistema, já que ele 30.000 laser na face do usuário (ZKTECO, 2021).

Não é tão parecido com a nossa ideia infelizmente, Esse hardware é utilizado para segurança biométrica, e nesse quesito ele é surpreendente, já que ele consegue distinguir fotos de pessoas e pessoa de verdade com sua tecnologia de infravermelho.

Apesar de criar um modelo 3d e poder acessá-lo para conferir se é o mesmo usuário, por questões de segurança não é permitido baixar o modelo e modificar, já que se ele estiver diferente da pessoa; ou essa pessoa vai ser barrada pelo sistema ou o modelo vai ser revertido baseado na pessoa real.

**2.2.3.Large Pose 3D Face Reconstruction**

Software desenvolvido por estudantes britânicos, esse programa consegue criar um modelo 3d de um rosto utilizando apenas uma única foto. Para isso, eles criaram uma *Convolutional Neural Network* (*Rede Neural Convolucional* ou *CNN)* e passaram várias fotos 2d e seus respectivos modelos 3d, treinando a AI (AARON, 2017).

O estudante Aaaron S. Jackson Se empenhou na criação de um sistema que utiliza marcas faciais para criara a malha.“*Mostramos como a tarefa relacionada da localização de marcos faciais 3D pode ser incorporada à estrutura proposta e ajudar a melhorar a qualidade da reconstruçã*o”[..] “*Demonstramos que nossa CNN trabalha com apenas um único Imagem facial 2D*”

Ela foi criada em 2017 com o propósito de reconstrução facial para usos de investigação forence, e se desempenhou melhor que 3DDFA e EOS, softwares que estavam sendo comparados no estudo.

Existia uma demo online do software, mas infelizmente o projeto foi descontinuado recentemente e sua última versão contem problemas e especificações que o tornam menos acessíveis e impreciso.

**2.2.4.Meshroom**

Meshroom permite fazer modelos 3d por meio de uma serie de fotos, como outros aplicativos de fotogrametria. Seu diferencial se da por ser um aplicativo completamente gratuito, e por permitir configurar cada passo do processo, desde como as fotos serão processadas até como a malha será renderizada (ALICE, 2022).

Ela é semelhante ao *scanner*, mas considerando que o melhor jeito de gerar um modelo humano assim é tirar múltiplas fotos ao mesmo tempo, é mais acessível ter varias câmeras do que ter vários *escaners*.

Isso o torna um sistema bastante complexo, que pode punir bastante iniciantes, mas bastante complexo. Usando um sistema de ponto de nuvens, ela une os pontos conseguidos por meio das fotos em uma única malha.

**2.2.5.PolyCam**

Usando tecnologia de LIDaR, Polycam consegue transformar uma série de fotos em um objeto 3d. Isso tudo usando apenas a câmera do celular e processando nele, o que o torna bem fácil de utilizar e acessível (POLYCAM, 2023).

O aplicativo é mais tratado como uma rede social, você posta os modelos criados com ele online e as pessoas reagem, possivelmente servindo como portfólio. Mesmo você criando o modelo, para você baixar e utiliza é necessário pagar por uma conta premi um.

**2.2.6.Blender Facebuilder**

Esse é possivelmente o aplicativo mais próximo do nosso conceito, Facebuilder é um addon open source para o Blender, criado pela empresa Kentools, que permite a criação de rostos foto realistas para uso profissional (KEENTOOLS, 2023).

Ele não cria um modelo do zero, na verdade fotos são importadas para o Blender e o próprio usuário ajusta o modelo para ficar enquadrado com as fotos e, eventualmente, com a utilização de varias fotos boas, o modelo fica mais e mais parecido com o rosto que estava sendo replicado.

Uma inteligência artificial está sendo desenvolvida para ajustar as fotos automaticamente, mas esta possui uma baixa precisão. Então o fato de ter que combinar várias fotos manualmente combinado com uma licença cara pode ser vista como um negativo.

1. **Resultados de Análise e Projeto**

Com o escopo do projeto definido, começou a fase de desenvolvimento. Nosso projeto foi dividido em duas partes, a primeira em gerar um modelo *lowpoly* utilizando apenas as características do rosto do usuário, e a segunda sendo um moelo mais detalhado utilizando o primeiro para correções.

1. **Linguagem**

Primeiramente procuramos por linguagens que possuem uma biblioteca de reconhecimento facial.

Nossa primeira escolha foi o *OpenCV* em *Python*, ele era o primeiro resultado nas pesquisas, sua precisão era um dos maiores atrativos e poderia ser escolhido quais pontos nodais seriam utilizados para o reconhecimento e a posição deles era salva. Infelizmente a versão disponível da biblioteca não era compatível com nenhuma versão disponível de *python*.

Procuramos então por uma versão em *javascript*, pois seria fácil de implementar para multiplataformas e seria uma opção leve. Um *bug* com o protótipo *js* fez com que o *canva* ocupar a tela inteira em vez de só marcar os pontos e sobrepor a câmera.

Paralelamente pesquisamos também por uma biblioteca na linguagem R, já que essa língua é bastante semelhante a *python*, e nela encontramos outro *OpenCV*. Esse era capaz de reconhecer o rosto, e não apresentou muitos erros, então utilizamos ele.

1. **Limitações**

Só porque a biblioteca funcionava não significava que ela era perfeita. Ela possui duas funções para retornar pontos, mas como podemos ver na figura 2, esses pontos não era referente ao rosto humano, mas sim quando o algorítimo encontrava variação no anglo da silhueta maior que o valor dado (nesse caso de 25°).

**Figura 2** Pontos da Imagem

**Fonte** – Autores, 2023

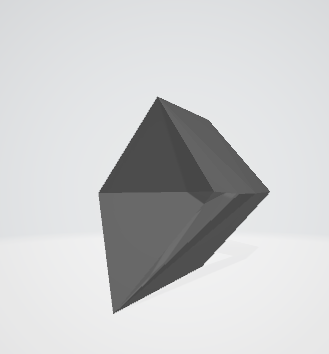
O outro não podia ser modificado, retornando pontos que a máquina considerou importante, podendo ou não ser parte do rosto.

O reconhecimento facial também não foi ideal. Ele não retorna pontos importantes, como a boca e o nariz, e mesmo pontos de referência que ele reconhecia, como o olho, não retornava a localização do mesmo.

Sendo assim, as únicas informações providenciadas pela biblioteca foram o centro do rosto e a distância do queixo até o final da testa, ficando a nosso cargo descobrir a localização de mais referências

1. **Resultados**

Como visto na figura 3, o modelo gerado não se assemelha a uma pessoa. Isso se deve a alguns erros nossos e por uma quantidade baixa de pontos detectados.



**Figura 3 -** Resultado do primeiro teste

**Fonte** – Autores, 2023

A detecção de borda por exemplo, na parte direita do rosto, ela não conseguiu distinguir o limite do rosto e o início da parede. Esse contratempo não foi tão grande, já que a mesma biblioteca de detecção de bordas também oferecia uma ferramenta de aumentar o contraste da imagem, que gerava um bom resultado quando combinado.

Nesse primeiro teste foi utilizado fotos com óculos o que dificultou a detecção dos olhos na vista de frente, e impossibilitou que o rosto seja detectado de lado.

Isso se provou um grande problema não só por limitar a quantidade de pontos que poderíamos conseguir do modelo, mas já que o diâmetro do rosto foi determinado de lado, não foi possível escalar a imagem de lado para condizer com a imagem de frente.

Então é partido do pressuposto que a foto de lado não só está na mesma distância da câmera de frente como também na mesma altura, o que gera imprecisões se o nariz numa foto estava 10 pixeis para cima da outra. Futuros testes ocorreram sem acessórios para aumentar a precisão do sistema, recomendamos que nossos usuários também façam isso.

Concluímos então que teremos de melhorar o método de como o protótipo é gerado, para retornar resultados mais aceitáveis.

A Figura 4 ilustra os procedimentos que nosso software executará de maneira consistente. Inicialmente, quaisquer acessórios removíveis devem ser retirados para aprimorar a precisão ao capturar a fotografia.

**Figura 4** - Fluxograma dos processos do programa **Fonte** – Autores,2023

Após o processamento da imagem do usuário, esta passará por modificações destinadas a intensificar o contraste, resultando em contornos faciais mais nítidos e reduzindo a ênfase em detalhes superficiais.

Dessa forma, a detecção de contornos pode ser aplicada, destacando de maneira clara as características faciais essenciais para análises subsequentes.

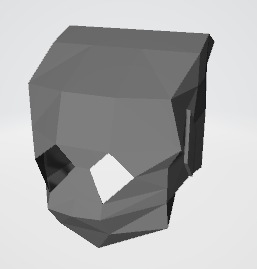
Portanto, empregamos funções para identificar a localização geral do rosto e outras características faciais. No entanto, embora essas características sejam detectadas, a biblioteca OpenCV() requer a realização de um processo adicional para determinar o centro dessas características e suas dimensões.

Utilizando os pontos fornecidos pelo sistema como base, é possível ampliar essa identificação. Muitos sistemas de visão computacional oferecem funcionalidades para recuperar pontos, embora bibliotecas não especializadas em reconhecimento facial possam gerar pontos não pertinentes à estrutura do rosto. Para mitigar esse problema, recorremos à simetria, a fim de localizar pontos não inicialmente detectados.

Isso envolve reconhecer características simétricas no rosto, permitindo a inferência de pontos que possam não ter sido diretamente detectados. Essa abordagem ajuda a aprimorar a precisão da detecção de pontos faciais mesmo em bibliotecas menos específicas para essa tarefa, tornando o processo mais robusto e confiável.

Após a conclusão desses processos nas duas imagens, utilizamos o eixo Y, a orientação da imagem e outros fatores relevantes para identificar pontos equivalentes nas duas perspectivas. Registrando as coordenadas X e Y na imagem frontal e aproveitando a coordenada X da imagem lateral como Z, criamos um arquivo de extensão .obj contendo esses pontos. Ao estabelecer relações entre esses pontos, conseguimos gerar um modelo 3D.

Com dedicação ao aprimoramento do código, conseguimos produzir um modelo mais fiel às características humanas, identificando pontos distintos como os olhos, nariz e orelhas, como ilustrado na Figura 5.



**Figura 6 -** Resultado do segundo teste **Fonte** – Autores,2023

Esse modelo agora incorpora 90 vértices. Embora seja nove vezes maior em relação ao modelo anterior, essa quantidade ainda é relativamente baixa para alcançar níveis realistas de detalhe em modelos tridimensionais.

1. **Resultados da solução proposta**

A proposta abrange a concepção de um protótipo voltado à geração precisa de modelos 3D. Não obstante sua aptidão nesse contexto, ao avaliar tanto os resultados apresentados quanto as imagens empregadas no procedimento, constata-se uma discrepância entre os mesmos.

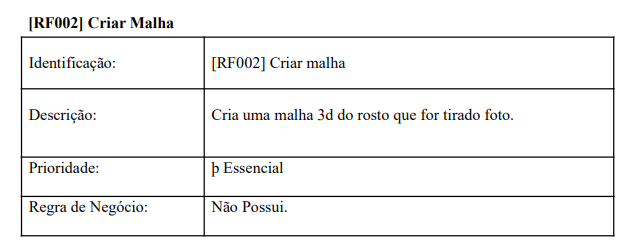
Vale ressaltar que a obtenção automatizada de modelos 3D, sem a necessidade de aplicação de técnicas de redução de ruído, configura-se como uma ocorrência raramente observada.

Dessa maneira, empregaram-se variados métodos com o objetivo de aprimorar a malha e conferir maior fidelidade à pessoa que serviu de modelo exemplar, como resposta a esse desafio.

1. **Testes**

Os testes são uma parte fundamental do processo de validação e verificação. Eles são usados para avaliar a funcionalidade do software, identificar problemas e garantir que ele funcione corretamente em diferentes cenários e condições. Tendo em vista esse cenário foram realizados os testes e procedimentos citados abaixo.

Para verificar se fizemos o sistema com as funcionalidades certas, utilizamos do Documento de Requisitos e verificar se todos os artefatos dele estão sendo implementados de acordo com os requisitos e as normas apresentadas ao projeto, como a Figura 6 Mostra.

 **Figura 6 -** Requisito Funcional RF002 **Fonte** - Autores, 2023

A documentação de requisitos completa está disponível através do link disponibilizado nas referências do artigo.

A nossa validação será um processo constante, nossas visões do projeto e de como realizá-lo pode mudar com nossa maturidade de desenvolvimento e com fatores externos. Sempre que formos implementar algo, vamos verificar todos os métodos possíveis para implementa e usaremos aquele que se provar melhor para nosso caso, se possível.

Com uma foto de frente e de lado dos dois desenvolvedores, o resultado esperado para o teste do primeiro protótipo é um modelo com poucos polígonos e sem textura. O segundo protótipo deve retornar a um modelo mais detalhado, com textura, mas utilizar mais fotos dos devs para evitar complicações éticas.

Para encontrar os pontos com mais facilidades, as imagens utilizadas passaram por um filtro de detecção de borda, onde os limites são pintados de branco e o resto de preto, como visto na figura 6. Dessa forma podemos percorrer pela imagem e procurar por valores de 1 nela, ao contrário de procurar o tom especifico da pele, que pode variar de usuário.

Outro teste realizado foi da autenticação do usuário, com a entrada de diversos padrões de verificar se a as regras de negócio relacionadas a autenticação serão cumpridas, tais como o padrão de tamanho e tipo de senha pedidos, ou se o nome de usuário atende ao padrão da plataforma.

1. **Pós-Processamento**

O modelo foi transferido para o software Blender, no qual cada superfície passou por um processo de subdivisão, resultando em um aumento no número de vértices. Adicionalmente, procedeu-se à aplicação de um alisamento no modelo, conferindo-lhe uma forma mais arredondada.

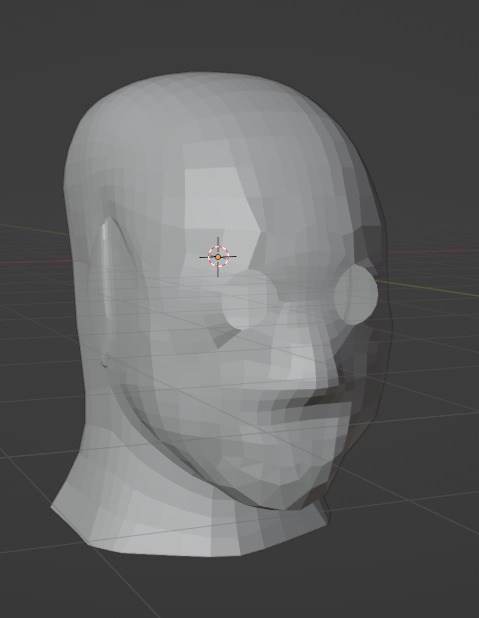
A Figura 7 evidencia que, mediante um período de 20 minutos no software, ao incrementar a densidade poligonal, houve uma significativa melhoria na realidade do modelo gerado.



F**igura 7 -** Nathan com filtro *edge* 2023

**Fonte** – Autores, 2023

Ao investir um maior lapso temporal na etapa de texturização, tornou-se mais evidente a identificação da origem do modelo, conforme ilustrado na Figura 8.



**Figura 7 -** Pós-Processamento do modelo

**Fonte** – Autores, 2023

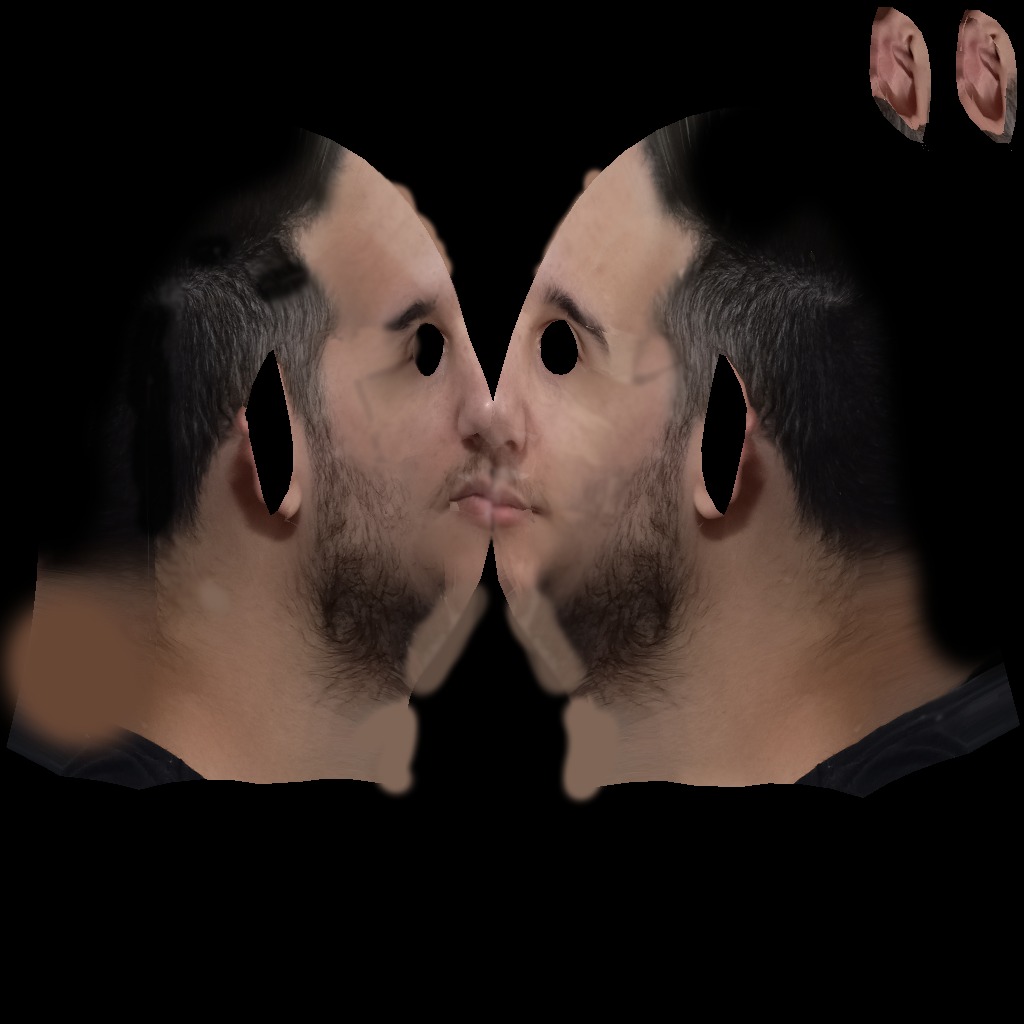


**Figura 8 -** Ultimo modelo com textura **Fonte** – Autores, 2023

1. **Texturização**

Nossa aplicação não incorpora a geração automática da textura do modelo, devido à complexidade desse processo, que é influenciada por diversos fatores.

Primeiramente, a imagem de textura requer a unificação de todas as imagens capturadas para a construção do modelo, consolidando-as em uma única imagem e ajustando-as para corresponder à proporção do modelo, como ilustrado na Figura 9.



**Figura 9 -** Textura do modelo final 2023

**Fonte** – Autores, 2023

O segundo e último motivo reside no fato de que, a fim de aplicar essa textura ao modelo 3D, cada vértice necessita de, pelo menos, uma coordenada de textura na formatação Wavefront OBJ. Estes pontos de coordenadas possuem valores compreendidos entre 0 e 1, que correspondem a uma porcentagem da imagem, embora a geração destes valores não tenha sido realizada.

1. **Conclusão**

O problema central abordado por este artigo é a dificuldade enfrentada por softwares automáticos de geração de modelos 3D ao criar malhas 3D para rostos humanos.

Uma característica importante desse protótipo é a sua acessibilidade. Ele foi projetado de forma a ser facilmente utilizado por qualquer pessoa que necessite criar modelos 3D de rostos humanos, independentemente do nível de habilidade em modelagem.

A abordagem de simplificar as inserções do usuário, limitando-as a apenas duas fotos, demonstra uma compreensão clara das necessidades dos usuários com pouca ou nenhuma experiência em modelagem 3D. Isso reduz a curva de aprendizado e facilita o processo de criação, tornando a tecnologia mais acessível e amigável.

É importante reconhecer que o sistema possui suas limitações, como qualquer tecnologia. No entanto, ele fornece uma base sólida para a criação de modelos 3D, especialmente no contexto de personagens humanos.

A existência de um ponto de partida com elementos humanos pré-modelados reduz consideravelmente o tempo necessário para construir esses modelos a partir do zero. Isso é particularmente valioso para artistas e designers que desejam se concentrar em detalhes específicos ou em outras áreas criativas, em vez de investir tempo na modelagem básica

[**Referências**](https://www.cs.cmu.edu/~mbz/personal/graphics/obj.html)

[[CS.CMU.EDU]. 3D Object File Formats. Disponível em:](https://www.cs.cmu.edu/~mbz/personal/graphics/obj.html) <https://www.cs.cmu.edu/~mbz/personal/graphics/obj.html>. Acesso em: 05 jun. 2023.

ZKTECO. Introduction of 3D structured light facial recognition\_4Jun,2021. Disponível em: [https://zkteco.com.br](https://zkteco.com.br/). Acesso em: 05 jun. 2023.

[ARXIV.ORG]. Large Pose 3D Face Reconstruction from a Single Image via Direct Volumetric CNN Regression. Disponível em: <https://arxiv.org/abs/1703.07834>. Acesso em: 05 jun. 2023.

KEENTOOLS.IO. FaceBuilder for Blender. Disponível em: <https://keentools.io/products/facebuilder-for-blender>. Acesso em: 05 jun. 2023.

CASA DAS CIÊNCIAS. Fotogrametria. Disponível em: <https://rce.casadasciencias.org/rceapp/art/2019/073/>. Acesso em: 05 jun. 2023.

KASPERSKY. What Is Facial Recognition? Disponível em: <https://www.kaspersky.com.br/resource-center/definitions/what-is-facial-recognition>. Acesso em: 05 jun. 2023.

SERASA. O que é reconhecimento facial e como funciona esse mecanismo? Disponível em: <https://www.serasa.com.br/carteira-digital/blog/o-que-e-reconhecimento-facial-e-como-funciona-esse-mecanismo/>. Acesso em: 05 jun. 2023.

[RESEARCHGATE]. Untitled Document. Disponível em: [https://www.researchgate.net](https://www.researchgate.net/). Acesso em: 05 jun. 2023.

BERNAL, Gabriel; BERNAL, Gustavo. Registro de Presença Acadêmico Através de Reconhecimento Facial.

SZELISKI, Richard. Computer Vision: Algorithms and Applications. Springer, 2011.

HARTLEY, Richard; ZISSERMAN, Andrew. An Introduction to Computer Vision. Prentice Hall, 2003.

SOLEM, Jan Erik. Programming Computer Vision with Python. O'Reilly Media, 2012.

ZHANG, Zhanpeng; LUO, Ping; LOY, Chen Change; TANG, Xiaoou. Facial landmarks detection based on 68 points, 2017.

GIOVANINI, Adenilson. Fotogrametria: o que é e para que serve.

REVOPOINT3D.COM. Range. Disponível em: [https://revopoint3d.com](https://revopoint3d.com/). Acesso em: 05 jun. 2023.

UPTODOWN. Meshroom para Windows - Baixe gratuitamente na Uptodown. Disponível em: <https://pt.meshroom.com/windows>. Acesso em: 05 jun. 2023.

[D1WQTXTS1XZLE7.CLOUDFRONT.NET]. IA\_FT\_UNICAMP\_visaoComputacional-libre.pdf. Disponível em: [https://d1wqtxts1xzle7.cloudfront.net](https://d1wqtxts1xzle7.cloudfront.net/). Acesso em: 05 jun. 2023