



RECONSTRUÇÃO VIRTUAL DE OBJETOS POR CAPTURA DE NUVEM DE PONTOS UTILIZANDO DISPOSITIVO DE BAIXO CUSTO

Angel Rodrigues Ferreira^{*1} e Alexandre Carvalho Silva¹

¹IF Goiano - Instituto Federal Goiano - Campus Morrinhos

Resumo - Neste trabalho é estudada a possível viabilidade do uso de um dispositivo de baixo custo no processo de reconstrução de objetos por meio de captura de nuvem de pontos. Os recursos oferecidos pelas APIs de desenvolvimento do ReconstructMe SDK foram utilizados apropriadamente na implementação da aplicação de captura e reconstrução; e o dispositivo utilizado para os procedimentos de captura foi o dispositivo Kinect. A solução estudada foi aplicada no processo de reconstrução de 3 objetos específicos. Os processos de reconstrução foram realizados com a condução do dispositivo manualmente, tentando cumprir movimentação de captura circundante ao objeto, em condições de limitações de recurso de espaço do ambiente utilizado. Considerando uma primeira experiência de implementação de uma aplicação com o ReconstructMe SDK, e ainda a utilização de um dispositivo de baixo custo, os resultados são consideravelmente satisfatórios. A solução tem capacidade de realizar os procedimentos de captura em tempo real e fazer a geração do modelo de volume de reconstrução em considerações de tempo relativamente rápidas e, mesmo que tendo alguns problemas de encaixe ou deformação de superfície, conseguindo resultados com qualidade de malha aceitável.

Palavras-Chave- Digitalização de objetos, Kinect, nuvem de pontos, ReconstructMe SDK, reconstrução virtual.

VIRTUAL RECONSTRUCTION OF OBJECTS BY POINT CLOUD CAPTURE USING LOW COST DEVICE

Abstract - In this paper is studied the possible feasibility use of a low cost device in the object reconstructions process through point cloud capture. The gave features by the ReconstructMe SDK development APIs were used properly in the implementation of the capture and reconstruction application; and the used device for the capture procedures was the Kinect device. The studied solution was applied in the reconstruction process with three specific objects. The reconstructions processes was done leading manually the device, trying to perform the capture movements around the object, under conditions of space limitations of the environment used. Considering a first

implementations experience of an applications by ReconstructMe SDK, and still the use of a low cost device, the results are really satisfactory. The solution has capability to realize the capture procedures in real time, and do the reconstruction volume model generation relatively fast in time considerations, even though having some surface fitting or deformation problems, however getting results with acceptable mesh quality.

Keywords - Kinect, object scanning, point cloud, ReconstructMe SDK, virtual reconstruction.

I. INTRODUÇÃO

A ação de tratamento/abstração de objetos (modelos de formas dimensionais em ambiente) recorre ao uso de dispositivos de captura que desempenhem funcionamento para aquisição estruturada, quanto aos requisitos das partes dispositivas de reconhecimento. Esses dispositivos providenciam meios específicos de contato externo com o ambiente. E como um desses meios, a visão pode ser sem dúvida uma das mais importantes formas de contato externo que o ser humano, dadas suas estruturas biológicas, pode desempenhar funcionamento.

Seguindo uma abordagem em [1], fazendo uma relação entre importância da visão computacional ao funcionamento da visão humana, onde, em contexto, em vez de olhos há câmeras utilizadas para obter imagens, e por trás das imagens capturadas são aplicadas técnicas específicas para a realização do tratamento dos dados obtidos. De tal modo, que se torna interessante o aspecto dos diferentes sensores capazes de propor essa capacidade de coletar dados do mundo real para o computador; a computação dos dados, e estes capturados apropriadamente pelos dispositivos sensoriais. Tão surpreendente como, ou quanto, os aspectos de visão implementados no sentido de máquina, com ampla diversificação de capacidade de sentido devido os mais diferentes e específicos sensores que desempenham devida capacidade, a máquina pode desempenhar papel de ver até mesmo o que o ser humano não pode ver. O que é um recurso importante para a percepção (explorar a percepção em recursos de visão sofisticados) [2]. Portanto, visualizar a informação física da característica de um objeto, é importante para determinadas áreas de trabalho, porém nem sempre a maneira de obter estes dados é eficiente.

*angel.rodrigues@estudante.ifgoiano.edu.br

Tendendo ao objetivo, é ideal explorar uma possibilidade de solução computacional de implementação técnica que atenda as necessidades do trabalho de mensuração de dados de medidas de objetos conseguindo, de maneira eficiente realizar os processos necessários.

A transformação computacional de um objeto em um modelo tridimensional pode demonstrar grande recurso para o trabalho com os dados de medida, viabilizando, além do contato de proximidade dimensional, a possibilidade de qualquer implementação consequente, que recorrendo importância no mesmo modelo, em contato com o próprio modelo; como um protótipo mensurável e de próximo contato. Em geral, existe uma necessidade crescente de se representar objetos em modelos computacionais, cabendo diferentes formas de se obter dados de forma de objetos [3].

Desse modo, com abordagem a alguma tecnologia em potencial, as nuvens de pontos podem ser estratégias suficientes para a digitalização de estruturas em ambiente, necessitando de alguma estrutura de sensores, os quais fazendo o recurso de captura.

Com base nestes cenários de necessidades e potencialidades, este trabalho tem como objetivo geral realizar a digitalização de objeto específico em ambiente real, realizando a reconstrução de superfícies do mesmo em dimensão computacional 3D. Considerando a abrangência de possibilidade de tratamento com o modelo, dada a praticidade de condução com o mesmo na dada dimensão computacional.

Viabilizando em objetivos específicos um modelo computacional de captura e reconstrução, utilizando de uma estrutura de dispositivo de baixo custo; considerando verificações de importância definindo: mensuração e análise indiretamente do tempo de reconstrução; alcance em complexidade de qualidade da malha (nuvem de pontos); e viabilização da implementação com baixo custo (quanto ao dispositivo).

II. FUNDAMENTAÇÃO TEÓRICA

Nos subtópicos a seguir são apresentadas fundamentações com o objetivo de aproximar o leitor as tecnologias utilizadas neste trabalho.

A. Reconstrução de Superfícies

A reconstrução de superfície é um modelo geométrico para usar dados da nuvem de pontos, através da medição e aquisição para ser restaurada a superfície original de objetos [4]. Processar uma nuvem de pontos de maneira rápida e econômica para obter um modelo 3D de objeto real que melhor se adapte à realidade é um trabalho desafiador [5]. Portanto, é significativo estudar a tecnologia de processamento de dados de nuvem de pontos tridimensional em como garantir precisão e eficiência no modelo de reconstrução de superfície [4].

O ReconstructMe SDK é uma biblioteca ISO C para procedimentos de reconstrução 3D em tempo real para operação manual de dispositivos de câmera RGB-D. Projetado como uma biblioteca de implementação de software que providencia fácil interface com o desenvolvedor que precisa de integrar um mecanismo de reconstrução 3D de alta performance

em suas aplicações. Providência interface para todo o pipeline desde a digitalização 3D, incluindo etapas de pré e pós-processamento, como filtragem de dados de sensor e tarefas de pós-processamento de superfícies 3D [6]. Suporta uma ampla variedade de sensores RGB-D comuns, como a família ASUS Xtion, a família PrimeSense Carmine ou a família Microsoft Kinect [7]. Toda a reconstrução é realizada no espaço métrico e o resultado pode ser exportado para vários formatos CAD, como STL, OBJ, 3DS e PLY [6].

Como uma biblioteca de middleware, o ReconstructMe SDK encapsula a comunicação com sensores 3D populares e abstrai os detalhes de hardware, como paralelização de tarefas [6]. Permite o desenvolvimento de aplicativos em questão de minutos; providencia uma API baseada em puro C, sem dependências adicionais; e tem fácil possibilidade de Interoperabilidade com outras linguagens [6].

O ReconstructMe SDK providência interfaces específicas, que ampliam as possibilidades de adaptar uma aplicação. Algumas necessárias e/ou essenciais em qualquer programa. Como as interfaces: de contexto, que fornece gerenciamento de tempo de vida do objeto e mantém a comunicação com um dispositivo de computação (GPU, CPU); de sensor, que fornece dados de imagem do mundo real (pelo sensor); de volume, como um espaço global, alimentado pelos dados de imagem, onde a reconstrução real acontece; e de superfície, que permite a extração de malhas de triângulos 3D do volume. Como também outras, alternativas para fins de testes, como a interface de calibração que permite estimar parâmetros da câmera, e a interface de visualização que abrange fins de depuração com exibição. As interfaces padrões são criadas com valores padrões, mas elas podem ser configuradas por uma interface de opções, o que é de devida importância. Contudo, esses componentes fornecem suporte para a comunicação com sensores, rastreando o movimento da câmera via odometria visual, fundindo os dados da digitalização em um quadro mundial e fazendo pós-processamento das digitalizações [6].

B. Dispositivo de Captura

As câmeras Time of Flight e os scanners a laser são ferramentas escolhidas e estudadas suas aplicabilidades, em trabalhos que exigem mensuração computacional de modelo real em consequências de tempo eficaz [8].

O sensor Kinect é capaz de reconstruir com precisão um objeto 3D e determinar o volume dos objetos, e se comparado com outros sensores de alcance usados para mapeamento 3D, o Kinect permite uma maneira muito mais barata de gerar nuvens de pontos [9], [10]. O sensor Kinect permite que o computador detecte diretamente a terceira dimensão (profundidade) de objetos, superfícies em cena. Com sua ampla disponibilidade e baixo custo, muitos pesquisadores e profissionais de ciência da computação, engenharia eletrônica e robótica aproveitam a tecnologia de detecção para desenvolver novas formas criativas de interagir com máquinas e realizar tarefas [11].

Trabalhos como [9], [12], [13], [14] apontam o alcance do modelo para diversas áreas com a avaliação e ou extensão de aplicabilidade do Kinect (como sensor de profundidade), abrangendo os campos que envolvem visão computacional,

mapeamento e modelagem 3D, robótica, dentre outros (cujos quais viabilizam implementação meio a tecnologia moderna).

Como um tipo de dispositivo híbrido/composto [8], [15], com baixa resolução e alta frequência de varredura (cerca de 30 FPS) [8], o Kinect é um sensor RGB-D que providencia sincronização de imagem RGB e profundidade de imagem [10]. Compreende vários equipamentos avançados de detecção [11]. Assim, composto por uma câmera RGB tradicional, um sensor de profundidade, que consiste em um projetor de padrões a laser de infravermelho próximo (NIR) combinado com uma câmera de infravermelho (IR, um sensor monocromático complementar de óxido de metal – CMOS) [11], [15], uma matriz de quatro microfones e um motor embutido (Servomotor) [9], [10].

Direcionando as definições de detecção ao sensor Kinect; a câmera de infravermelho é usada para observar e decodificar o padrão de projeção de infravermelho para triangular a cena 3D [15]. Basicamente, o emissor IR emite um padrão de manchas pseudoaleatórias (um único feixe que é dividido em vários feixes por uma grade de difração para criar um padrão constante de manchas projetadas) no objeto e a câmera infravermelha captura as reflexões do objeto [14], [16]. Para a captura de pontos de profundidade são calculadas relações geométricas entre o projetor e a câmera IR através de um procedimento de calibração off-line. O projetor dispõe um padrão de luz IR na cena 3D; a câmera IR captura as manchas do padrão em alguma superfície, conseqüente faz correspondência entre os padrões de pontos locais observados na imagem e os padrões de pontos do projetor calibrados [10].

A geometria relativa entre o projetor de infravermelho e a câmera de infravermelho, bem como o padrão de ponto de infravermelho projetado, são conhecidos, e se conseguindo combinar um ponto observado em uma imagem com um ponto no padrão do projetor, torna-se possível reconstruí-lo em 3D usando triangulação [11].

Ao registrar as imagens de profundidade consecutivas, é possível obter uma densidade de pontos aumentada, e criar uma nuvem de pontos completa de um ambiente de cena, possivelmente em tempo real. A nuvem de pontos de um sensor Kinect devidamente calibrado não contém grandes erros sistemáticos quando comparados com dados de varredura a laser [14].

O desempenho do sensor pode ser afetado por uma série de fatores, tanto externos do ambiente e/ou da cena de captura, quanto específicos do dispositivo, que influenciam sua precisão e resolução de profundidade. Fatores como nível de luz ambiente, tipo de material, cor do material, presença de luz infravermelha emitida por outros sensores, variações de temperatura, corrente de ar e a distância do sensor ao contato de captura [8], [9], [10].

III. TRABALHOS CORRELATOS

Em [17] é implementada uma metodologia de software geral que tem em objetivo fazer a captura de uma cena de ambiente com objetos específicos de interesse (na área de aplicação), com uma maneira de reconstruir objetos tridimensionais em nuvens de pontos de LiDAR (Detecção de Luz e Alcance)

terrestre (TLiDAR). Os dispositivos de modelo de captura LiDAR trabalham com a mensuração no tempo de voo do feixe de luz. Utilizando em experimento um aparelho FARO Photon120 TLiDAR, ainda fazendo observação da baixa portabilidade do aparelho em ambientes operacionais complexos; o processo de reconstrução se faz seguindo um layout específico para cumprir de estações de digitalização (com orientações de localização das mesmas) para cada objeto ou cada conjunto de objetos, seguindo um padrão de costura de sena com orientações de localização. Fazendo seleção de parcelas de amostra com base em critérios de importância específicos, os modelos de objeto são estabelecidos em tarefas de pós-processamento em softwares específicos (tais estes AutoCAD e “3DMAX”), tal que em completude dos procedimentos, são preparados modelos de regiões com conjuntos de modelos dos objetos.

Em [18] é explorada e implementada toda uma arquitetura de processamento (um pipeline) para com a nuvem de pontos realizar o processamento de reconstrução de maneira eficiente e limpa. Utilizando de algoritmos em programas de processamento específicos (tais estes, Geomagic e MeshLab). Sendo utilizado um algoritmo K-N (K-Vizinhança) para a eliminação dos ruídos, um método de disco de Poisson para a amostragem/simplificação do número de pontos, a estimação das normais de superfície por outro algoritmo K-N; isso precorre ao processo de reconstrução com base na nuvem de pontos por um algoritmo de reconstrução de Poisson para obter a malha triangular. Reconhecendo a importância de selecionar os parâmetros apropriados para obter resultados ideais; e ainda a questão de sofisticação da implementação para uma generalização aplicável em trabalhos recorrentes do processamento específico.

O trabalho [19], com o objetivo de fazer medições do corpo do objeto específico tendo o mesmo em recurso de modelo tridimensional, recorre a reconstrução do objeto em uma nuvem de pontos por meio de uma técnica de sensoriamento remoto óptico utilizando um sensor LiDAR. O LiDAR usado foi um Velodyne VLP-16. É importante observar que, em característica das especificações do sensor, o mesmo pode trabalhar com distâncias de 1 a 100 m; e a nuvem de pontos gerada recupera linhas horizontais de pontos da superfície. Trabalhando com uma ferramenta de visualização RViz; o modelo pode ser gerado e podem ser feitos tratamentos conseqüentes com o mesmo.

Os dados são trabalhados no software de processamento de nuvem CloudCompare [19]. Necessitando do modelo 3D de nuvem de pontos para a mensuração de partes, é praticada a técnica de medida de maneira manual. Contudo recorre importância do trabalho da reconstrução, a conseqüente possibilidade de contato com o modelo do próprio objeto de interesse em recurso tecnológico de ambiente computacional. Sendo que em primeiro contato com o modelo, são necessárias fil-tragens/cortes (também de maneira manual) no material da nuvem de pontos, a fim de isolar o objeto em específico de importância, como único conteúdo em cena para os métodos de contato de medida. Implementando um sistema que simplifica a tomada de medidas (no ambiente de aplicação específico do trabalho).

Em contexto de avaliação de pontos específicos importan-

tes nos trabalhos relacionados. Podendo encontrar complexidades quanto a viabilidade de implementação de um modelo de reconstrução com relação a uma área problemática específica: o dispositivo utilizado em [17] tem capacidade de captura impressionante, mas o mesmo tem forma tanto robusta, o que exige de bastante esforço operativo no controle de manuseio/montagem; o dispositivo utilizado em [19], também demonstrando capacidades interessantes, no entanto, os pontos capturados não podem descrever toda a superfície do objeto; e ambos esses dispositivos podem ser consideravelmente caros. E como demonstrados em [18], os processos específicos a se dedicar na estrutura de desenvolvimento da sequência dos processos essenciais para a reconstrução, é interessante e de importante recurso poder fazer uso de uma biblioteca que abstraia os processos e permite um direcionamento de pesquisa.

IV. IMPLEMENTAÇÃO E FUNCIONAMENTO DO SISTEMA

Como recurso geral no processo de captura e reconstrução, foram utilizadas, para a implementação aplicativa, as funcionalidades das APIs que integram o ReconstructMe SDK (versão 2.4.1016), recurso esse que abstrai o desenvolvimento com toda uma estrutura de processamento específica. Para trabalhar com o ReconstructMe SDK são necessárias as configurações de ambiente para a compilação e conexão com os recursos de pacotes necessários e de bibliotecas. Para a montagem/configuração do ambiente de desenvolvimento foi utilizada a ferramenta CMake (versão 3.16), cuja qual também recomendada pela própria documentação do ReconstructMe SDK, fazendo a configuração de projeto com a especificação de gerador para trabalhar com Visual Studio 2019 e usando compiladores padrões. Sendo o próprio Visual Studio 2019 (versão 16.7.3) o ambiente de desenvolvimento consequente utilizado na escrita/compilação do programa com a linguagem de desenvolvimento C++.

Como uma parte importante no trabalho com o ReconstructMe SDK, sendo ter um dispositivo RGB-D para trabalhar com o processo de reconstrução, e ainda com o contexto do trabalho, foi utilizado um dispositivo Kinect (modelo 1414) da Microsoft Corporation. Para trabalhar com o dispositivo em um computador, foi necessária a utilização de uma adaptação, como uma fonte de alimentação específica que conduz a conexão dispositivo-para-computador por USB; e como requisito para a utilização do dispositivo no sistema Windows (versão 10 de 64 bits), foi necessária a instalação/configuração da Runtime e dos Drivers para o dispositivo.

Assemelhando ao processo de desenvolvimento praticado neste trabalho, seguindo em estudo da documentação do ReconstructMe SDK, foram desenvolvidas e adaptadas as lógicas de funcionamento no algoritmo, sempre conferindo indiretamente funcionamento pós compilação em pontos importantes na implementação, por meio das interfaces que permitem a visualização da reconstrução em tempo real, e pós processada. Foram exploradas algumas configurações de captura e reconstrução, e também relacionadas ao sensor.

Como uma breve fase de teste e preparo, foram aplicadas alterações em algumas das opções de configuração padrão. As

alterações mantidas, foram: quanto as opções de reconstrução, uma redução nos cantos do volume pelas orientações máximas e mínimas para uma caixa com lados de 0,5 m; ainda em mesma base, uma sincronização automática para média qualidade, considerando a proporção entre qualidade e tempo de processamento; quanto as opções de superfície, uma redução a limite no número de faces (para pós-processamento). Contudo foram configurações básicas, visualizando uma estrutura de funcionamento base neste trabalho.

A. Funcionamento

É necessário o dispositivo Kinect estar conectado. De maneira geral, quando a aplicação é iniciada, são criadas e relacionadas as interfaces essenciais de funcionamento do ReconstructMe SDK para o programa. Essas são interfaces de: contexto, opções, volume, superfície e sensor. O sensor é criado em relação a um contexto (um objeto central para o ReconstructMe SDK).

Fazendo início ao processo de reconstrução, são carregadas as configurações de reconstrução definidas; o sensor é aberto carregando configurações padrões, e definindo posição em relação ao volume de reconstrução como posicionado em frente. Como um programa ainda com interesses na observação do processo de reconstrução (como depuração), ainda é utilizado um visualizador para rastrear o processo de reconstrução em tempo real. Com base nisso é iniciado o processo de reconstrução, sempre tentando determinar a posição do sensor na cena, fazendo relações de posição precorrente, e tentando atualizar o volume e a visualização da depuração.

Contudo, pós a finalização da reconstrução, tendo o volume correspondente ao modelo, o mesmo é extraído para a geração de superfície; um pós-processamento de redução no número de faces é carregado e o modelo pode ser salvo e visualizado em forma limpa ainda na aplicação, e consequente, em outro software 3D, importando do formato ply.

B. Limitações de Uso

A aplicação não está habilitada em uma licença de uso comercial do ReconstructMe SDK, com isso, mesmo sendo possível visualizar na aplicação o modelo 3D reconstruído, o mesmo vai ser salvo com variações de orientação de cena e com formas esféricas posicionadas junto. Limitando ainda a escolha de objeto a ser escaneado, considerando ambiente de proximidade em momento da pesquisa, são escolhidos os objetos limitando a aplicação a um pequeno volume de reconstrução para mais fácil circunscção no processo de reconstrução.

C. Dificuldades de Uso

Dificuldades podem surgir no uso da aplicação por diferentes circunstâncias externas, como: estar utilizando um computador desktop para conduzir a aplicação, e com isso o dispositivo deve estar sempre conectado ao mesmo; junto a isso, tendo a proporção de alcance do dispositivo quanto ao comprimento do cabo utilizado; e ainda, o espaço do ambiente onde ocorre a digitalização.

É essencial destacar o tratamento com o modelo 3D, externo a aplicação, sendo necessário um tratamento manual com o mesmo, tentando retirar as formas e superfícies diferentes do objeto de reconstrução, e fazendo um redimensionamento de escala por definição padrão no modelo em comparação ao objeto real.

V. TESTES PARA A OBTENÇÃO DE RESULTADOS

Na fase de testes foi preparado o ambiente de digitalização, com recurso de espaço e uma plataforma para o posicionamento do objeto a ser escaneado; e escolhidos (de acordo aos de mais próximo recurso) os objetos a serem digitalizados para a avaliação do processo aplicativo de reconstrução. Os objetos escolhidos foram: uma caixa de papel, buscando importância em sua forma retangular; uma caixa de plástico de formato circular, também com importância a sua forma; e um galão de plástico, como um objeto não simétrico.

Os processos de testes se basearam na digitalização dos objetos escolhidos citados acima, por vez, posicionados sobre uma pequena base circular de uma base retangular. A digitalização se fez com condução manual do dispositivo no ambiente, tentando manter seu posicionamento a cerca de 0,60 m do objeto, e assumindo movimentos circundantes ao mesmo; também capturando de ângulos verticais. No entanto, devido a pouco espaço em ambiente não foi possível o movimento circundante completo aos objetos, com isso foram assumidos, ao menos, ângulos horizontais próximos a 270°. Após o procedimento de digitalização em volume, o processo de geração do modelo de reconstrução junto ao de pós-processamento de redução de faces geralmente custou 25 s. Os modelos 3D resultantes passaram por processos manuais em software 3D externo, onde foram feitas reescalas e cortes nas cenas, para destacar o objeto de importância. Veja os resultados em Figura 1.

Figura 1: Modelos 3D recortados da cena de reconstrução.



De acordo com a relação de objeto por legenda (Figura 1), podem ser feitas algumas observações quanto a qualidade das formas geradas pelos processos de reconstrução. Quanto ao objeto A, as partes de superfícies planas podem facilmente ser identificadas, existe baixa conservação das bordas, e ainda deformações de forma. Do objeto B, sua forma circular pode ser parcialmente mantida, no entanto existem deformações de

forma, principalmente nas bordas. Já com o objeto C a qualidade se mostrar superior aos outros objetos, conservando aspectos formacionais com bom grau de precisão; tendo sua forma maior e sem ângulos agudos a circundação ao mesmo mantêm o encontro de pontos de referência no objeto do volume de reconstrução, facilitando a orientação de localização.

Os resultados foram satisfatórios, embora, seja importante destacar, problemas de localização em cena com relação ao dispositivo e ao objeto no volume de reconstrução. A aplicação em momento de digitalização se mostrou necessária de pontos de referência no volume, com isso as partes principais na orientação foram as bases. Mas mesmo nessa condição, algumas formas se desencaixavam em cena, principalmente quanto as do objeto retangular.

O problema encontrado pode ser um fator relativo a configuração da aplicação em relação ao ReconstructMe SDK, talvez algo como a resolução do volume de reconstrução, ou integração de dados pelo ajuste de truncamento. Estes mesmos dados foram configurados automaticamente como: em resoluções com valores de 512 para os eixos x e y, 256 para o eixo z, e em truncamento com o valor 9.766. Também podem ser questionados fatores relativos a de configurações intrínsecas de sensor, cujas quais não foram exploradas neste trabalho.

VI. CONCLUSÃO E TRABALHOS FUTUROS

Apesar dos resultados obtidos com a aplicação não alcançarem sempre uma precisão de encaixe de superfícies do processo de reconstrução, neste trabalho foi possível implementar e estudar a aplicabilidade de um sensor de baixo custo, num processo de reconstrução por nuvem de pontos; e ainda estudando aspectos de implementação/configuração quanto a capacidade dos recursos oferecidos pelo ReconstructMe SDK.

Analisando os resultados é possível fazer observações que destacam a aplicabilidade da solução proposta neste trabalho. Uma vez que, geralmente um dispositivo Scanner 3D tradicional pode ser caro e impor muitas limitações, como em relação ao tamanho limite do objeto possível para o processo de digitalização. Com base nos dados obtidos, e observações do procedimento de atuação na utilização da aplicação de reconstrução com o dispositivo Kinect, é identificada a capacidade de utilização do dispositivo para o processo de reconstrução. Foi possível realizar os processos de digitalização, em manuseio do dispositivo, em custo de tempo relativamente pequeno, podendo levar até menos de 60 s por processo. E ainda levando em consideração que a capacidade de geração de malha de volume de reconstrução ser um processo que leva menos de 30 s. Nessa solução foi possível obter um resultado de malha de objeto relativamente bom, e ainda que tendo instabilidades nas formas, a qualidade da malha pode ser aceitável para procedimentos de pós-produção que via de regra são aplicados em cenários de digitalização tridimensional. Portanto os resultados são consideravelmente satisfatórios.

Com relação aos erros de captura e reconstrução, são propostos argumentos de que a aplicação de reconstrução podera ser estudada e melhorada, em questão dos aspectos de configuração oferecidos pela Interface de desenvolvimento do ReconstructMe SDK; e ainda é possível destacar um fator que

podera influenciar na qualidade de captura do sensor, relativo a distância, a distância mínima de captura, um fator que não foi seguido ao ideal em condições das limitações do espaço ambiente utilizado para os processos de reconstrução. E mesmo que se consequente a ideal configuração nos pontos destacados, a solução não conseguir obter bons resultados, quanto ao modelo de objeto reconstruído, a adaptação específica de algoritmos de pós-processamento com os dados obtidos talvez consiga uma boa estabilização do modelo 3D de objeto.

Aqui o dispositivo de captura trabalha com a reconstrução de objetos relativamente grandes, sendo ainda que possivelmente o mesmo possa trabalhar com objetos ainda maiores, o que abrange a dimensão de aplicabilidade da solução para áreas específicas (considerando que o tamanho do volume de reconstrução com o ReconstructMe SDK pode também abranger grande área de reconstrução), sendo ainda possível a capacidade de melhores resultados se a estrutura de objeto tiver bons pontos de referência quanto aos ângulos de exploração.

Como complemento em visão para um trabalhos futuros, pode ser buscado ainda utilizando o dispositivo kinect, seguindo deste trabalho, além de realizar implementações na sofisticação de precisão da aplicação de reconstrução, capacitar um modelo aplicativo para recursos de classificação com os dados de modelos 3D, mas em aspectos de parametrizar e dispor as características específicas e importantes nos dados capturados; podendo para isso, ainda serem necessários processos precorrentes atuantes no pós-processamento dos modelos de objeto. Definindo uma especificação de tipo de objeto a ser reconstruído, direcionando o foco da aplicação a algum ambiente cujo qual recorre possibilidade de aplicação da sofisticação de reconstrução e maleabilidade para tratamento dos dados, oferecendo capacidade de realizar medições baseadas em padrões estabelecidos para análise e classificação de objetos reais, facilitando recurso de classificação subjetiva, ou automatizada.

REFERÊNCIAS

- [1] G. T. M. Alves, "Um Estudo das Técnicas de Obtenção de Forma a Partir de Estéreo e Luz Estruturada para Engenharia", Thesis, *Pontifícia Universidade Católica do Rio de Janeiro*, pp. 88, June, 2005;
- [2] R. C. Gonzalez and R. E. Richard, *Digital image processing*, Pearson, 4a Edição, New York, 2018.
- [3] C. N. Vasconcelos, "Algoritmos para Processamento de Imagens e Visão Computacional para Arquiteturas Paralelas em Placas Gráficas", Thesis, *Pontifícia Universidade Católica do Rio de Janeiro*, pp. 155, March, 2009.
- [4] Y. Haibo, "Industrial Design Applications of Surface Reconstruction Algorithm Based on Three Dimensional Point Cloud Data", in *Proc. of ICRIS*, Vol. 1, pp. 178-181, October, 2017.
- [5] R. Zhang, Y. Wang, and D. Song, "Research and implementation from point cloud to 3D model", in *Proc. of IC-CMS 2010*, vol. 2, pp. 169-172, January, 2010.
- [6] C. Heindl, H. Bauer, M. Ankerl, and A. Pichler, "ReconstructMe SDK: a C API for Real-time 3D Scanning", in *Proc. of the 6th International Conference on 3DBST 2015*, pp. 185-193, October, 2015.
- [7] D. Chikurtev, I. Rangelov, N. Chivarov, and D. Karastoyanov, "3D modelling for object recognition with depth sensors", *Problems of Engineering Cybernetics and Robotics*, vol. 70, pp. 35-42, 2018.
- [8] N. Rafibakhsh, J. Gong, M. K. Siddiqui, C. Gordon, and H. F. Lee, "Analysis of XBOX Kinect Sensor Data for Use on Construction Sites: Depth Accuracy and Sensor Interference Assessment", *Construction Research Congress 2012*, pp. 848-857, May, 2012.
- [9] N. M. DiFilippo, and M. K. Jouaneh, "Characterization of Different Microsoft Kinect Sensor Models", *IEEE Sensors Journal*, vol. 15, pp. 4554-4564, August, 2015.
- [10] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced Computer Vision With Microsoft Kinect Sensor: A Review", *IEEE Transactions on Cybernetics*, vol. 43, pp. 1318-1334, October, 2013.
- [11] Z. Zhang, "Microsoft Kinect Sensor and Its Effect", *IEEE Multimedia*, vol. 19, pp. 4-10, February, 2012.
- [12] M. R. Andersen, T. Jensen, P. Lisouski, A. K. Mortensen, M. K. Hansen, T. Gregersen, and P. Ahrendt, "Kinect depth sensor evaluation for computer vision applications", *Aarhus University, Department of Engineering*, pp. 1-37, February, 2012.
- [13] W. Khan, E. Phaisangittisagul, L. Ali, D. Gansawat, and I. Kumazawa, "Combining features for RGB-D object recognition", in *Proc. of iEECON*, pp. 1-5, March, 2017.
- [14] K. Khoshelham, "Accuracy analysis of kinect depth data", in *Proc. of ISPRS workshop laser scanning*, vol. 38, pp. 133-138, May, 2011.
- [15] J. Smisek, M. Jancosek, T. Pajdla, "3D with Kinect", *Springer, London*, pp. 3-25, 2013.
- [16] A. S. Sabale, and Y. M. Vaidya, "Accuracy measurement of depth using Kinect sensor", in *Proc. of CASP*, pp. 155-159, June, 2016.
- [17] H. Liu, X. Zhang, S. Wang, and L. Chen, "The reconstruction of three-dimensional tree models from terrestrial LiDAR", in *Proc. of IEEE International CSAFE*, pp. 371-374, June, 2011.
- [18] Q. Xu, J. Wang, and X. An, "A pipeline for surface reconstruction of 3-dimensional point cloud", in *Proc. of ICALIP*, pp. 822-826, July, 2014.
- [19] M. Pérez-Ruiz, D. Tarrat-Martín, M. J. Sánchez-Guerrero, M. Valera, "Advances in horse morphometric measurements using LiDAR", *Computers and Electronics in Agriculture*, vol. 174, pp. 105510, July, 2020.