

Reconstrução 3D de Objetos com Kinect e Câmera Digital

Daniel S. A. Loura¹, Yuri M. A. Oliveira¹,
Pedro O. Raimundo², Karl P. A. Agüero²

¹ Departamento de Engenharia de Controle e Automação de Processos –
Universidade Federal da Bahia (UFBA), Salvador, Bahia, Brasil

² Departamento de Ciência da Computação – Universidade Federal da Bahia
(UFBA), Salvador, Bahia, Brasil

{daniel.alves.spp, yuri.matos}@gmail.com

{praimundo, kaguero}@ufba.br

Abstract. *Objects with complex surfaces and details of impractical reproduction with Computer Aided Design (CAD) software are the main target of scanning and 3D reconstruction processes. As part of this work, a method for 3D models generation and reconstruction was devised. Real object's geometry and color information were respectively captured by a Microsoft Kinect V1 depth sensor and a digital camera, aiming a low-cost process. This article presents the methodology, equipment, processes and initial results from 3D reconstructions of important cultural objects from UFBA's Museum of Archeology and Ethnology (MAE).*

Resumo. *Objetos com áreas superficiais complexas e detalhes de reprodução manual impraticável em software de Computer Aided Design (CAD) são os principais alvos dos processos de digitalização e reconstrução 3D. Neste trabalho, foi concebido um método para a reconstrução e geração de modelos 3D. Informações de geometria e cor de objetos reais foram respectivamente capturadas por um sensor de profundidade Microsoft Kinect V1 e uma câmera digital, objetivando um processo de baixo custo. Este artigo apresenta a metodologia, equipamentos, processos e resultados iniciais da reconstrução 3D de objetos culturais importantes do Museu de Arqueologia e Etnologia (MAE) da UFBA.*

1. Introdução

Digitalização 3D de objetos é uma técnica relativamente recente e possui aplicações em áreas educacionais e mercadológicas. Um dos estudos mais importantes e que motivou este trabalho é o projeto *Michelangelo Digital*, por M. Levoy et al. [1]. Este projeto, pioneiro na área de preservação digital de acervos culturais, permitiu a reconstrução 3D de algumas esculturas de *Michelangelo*, considerado um dos maiores criadores da história da arte. O projeto foi executado no museu *Galleria dell'Accademia*, onde foram capturados os objetos utilizando um *scanner* de alta resolução e uma câmera digital. A partir destas capturas, foi aplicado um processo de reconstrução 3D, gerando resultados interessantes, como por exemplo a reconstrução digital da escultura Davi de Michelangelo. No entanto, o trabalho demandou grande quantidade de recursos humanos e financeiros, pois equipamentos semiautomáticos custosos foram utilizados para reconstruir as peças.

O presente trabalho apresenta um método semiautomático de reconstrução 3D, utilizando as principais técnicas de reconstrução para preservação digital de objetos

culturais, como discutido por L. Gomes, L. Silva e O. Bellon [2]. Contudo, diferente do projeto *Michelangelo Digital*, neste trabalho é proposto um sistema de baixo custo usando o sensor de profundidade Microsoft Kinect 360 e uma câmera digital. O sistema é considerado de baixo custo devido ao menor valor de aquisição dos equipamentos quando comparados a *scanners* de alta resolução, como os utilizados no projeto *Michelangelo Digital*.

O Kinect foi utilizado para capturar a informação de geometria da superfície do objeto através de imagens de profundidade, já a câmera digital para registrar as cores em alta resolução. Diversas etapas do processo de reconstrução 3D foram executadas com *MeshLab* [3], como o alinhamento de nuvens de pontos, geração de malha e mapeamento de texturas. Estas etapas serão discutidas nas seções posteriores.

Atualmente, a metodologia proposta está sendo executada no Museu de Arqueologia e Etnologia (MAE) da UFBA [4], portanto peças do seu acervo foram selecionadas para este trabalho. O MAE guarda um valioso patrimônio arqueológico e etnográfico representativos do passado pré-colonial, colonial e contemporaneidade de povos indígenas que contribuíram para a formação da diversidade de povos e identidades brasileiras. Todos os modelos apresentados neste artigo fazem parte do acervo do museu.

1.1 Trabalhos Relacionados e Metodologias

F. Bernardini e H. Rushmeier [5] apresentam algumas metodologias já disponíveis comercialmente capazes de capturar imagens de profundidade, gerando nuvens de pontos em três dimensões. Diversas tecnologias foram desenvolvidas para este fim, a exemplo dos sistemas de captura baseados em triangulação e dispositivos *time-of-flight* (*ToF*).

Sistemas de triangulação funcionam devido ao conjunto emissor e receptor de luz. É projetado um padrão luminoso conhecido sobre o alvo a ser escaneado e a reflexão deste padrão, já deformado pela geometria do alvo, é captado pela câmera receptora. Um *software* processa essa informação e com base na posição e orientação do conjunto emissor-receptor, realiza cálculos para calcular a imagem de profundidade que representa a superfície em três dimensões desse objeto. Esses métodos geralmente apresentam boa precisão, mas são relativamente sensíveis a mudanças na cena de reconstrução e a utilização de materiais que interfiram na definição do padrão projetado.

No caso de sistemas de *ToF* um pulso muito curto de luz é emitido sobre o alvo e é refletido de volta ao sensor. Com base no tempo emissão-recepção é possível calcular a distância aproximada do ponto de reflexão ao sensor, possibilitando a geração de uma imagem de profundidade. Entretanto esses sistemas precisam ser muito sensíveis a pequenas variações na informação captada, visto que para pequenas diferenças de distância a variação no tempo que a luz demora para atingir esses dois pontos é baixíssima, dada sua velocidade de cerca de 300 milhões de metros por segundo, exigindo um equipamento com muito baixa latência e breve tempo de resposta. Por esse motivo, esses sistemas tendem a ser relativamente ruidosos, como demonstra Y. Cui et al. [6].

Outra técnica de geração de nuvens de pontos 3D é a fotogrametria, que segundo T. Schenk [7] é a ciência de obtenção de informações confiáveis sobre superfícies e objetos sem o contato físico, medição e interpretação dessas informações. Consiste na captura de diversas imagens em diferentes ângulos do objeto, então uma série de algoritmos são aplicados sobre as informações base, como a extração de elementos chave, descrição de características, construção das nuvens de pontos e densificação, conforme citado por A. Utomo e C. Wibowo [8]. No entanto, para obter bons resultados com esse tipo de técnica é necessário estar atento a detalhes como diferenças de escala entre as fotos,

sobre/subexposição da captura e balanço de branco, uma vez que a informação de distância é derivada das imagens e não associada a alguma grandeza física. Outro desafio considerável da fotogrametria é realizar a separação do objeto de interesse do restante da cena, o que requer ou o uso de algoritmos sofisticados de segmentação ou considerável intervenção na cena da captura como também abordado por P. Raimundo, K. Agüero e A. Apolinário Jr. [9].

Muitos sistemas de captura de informação 3D disponíveis comercialmente são baseadas na tecnologia da triangulação com algum tipo de luz estruturada, o que permite gerar modelos virtuais sem o contato físico direto às peças, fato muito importante quando se objetiva escanear objetos frágeis, como esculturas, obras de arte ou artefatos muito antigos. C. Wu-Wei [10] discute sobre a preservação virtual de heranças históricas, poupando-as de condições que prejudicam as condições destas peças, como temperatura, umidade, luz, desastres naturais, entre outros. Todos esses fatores são eliminados quando são geradas representações 3D destes itens, podendo ainda contribuir para a posterior restauração dos objetos reais.

Em uma abordagem semelhante ao presente trabalho V. Reis e L. Jorge [11] utilizam o sensor Kinect V1, uma plataforma giratória e *software* para realizar o processo de captura e reconstrução 3D com o intuito de classificar frutos com o auxílio de inteligência artificial e de processamento digital de imagens para atingir um maior nível de confiança na classificação. B. Adlkhast e O. Manikhi [12] apresentam de forma detalhada o desenvolvimento de um sistema completo e automatizado de digitalização de objetos utilizando Kinect V1. Utilizando também uma plataforma giratória, conseguem realizar reconstruções de baixo custo sem a dependência de *hardware* especial, resultando em um pipeline com tempos fixos de digitalização, independentemente das dimensões do item. M. Zollhöfer et al. [13] também explora o baixo custo do sensor Kinect V1 em conjunto com métodos robustos para realizar digitalizações em larga escala de um campo de escavação. Neste trabalho, é demonstrada a capacidade de converter a baixa resolução de entrada do sensor em informação de alta qualidade utilizando algoritmos como super-resolution em tempo real.

1.2 Equipamentos

Utilizando como referência as metodologias apresentadas na seção anterior, além de outros fatores como custo, disponibilidade em mercado, portabilidade, facilidade de uso, independência de *software* e compatibilidade com sistemas operacionais, resolução de captura e reprodutibilidade, foram comparadas as principais características de alguns equipamentos, como apresenta a Tabela 1.

Tabela 1. Comparação das principais características dos sensores de profundidade.

Sensor	Kinect V1	Kinect V2	RealSense™ SR300
Tecnologia	Triangulação com Luz Estruturada	Time of Flight	Triangulação com Luz Estruturada
Resolução de Imagem de Cor	640x480 30fps 1280x960 12fps	1920x1080 30fps	1920x1080 30fps 1280x720 60fps
Resolução de Imagem de Profundidade	640x480 30fps	512x424 30fps	640x480 60fps
Alcance de Profundidade	0,4m – 3,0m (modo perto) 0,8m – 4,0m (padrão)	0,5m – 4,5m	0,2m – 1,2m
Custo (Maio de 2018)	R\$ 35,00 – R\$200,00	R\$ 180,00 – R\$800,00	R\$ 300,00 – R\$900,00

Com base nos dados da Tabela 1, foi optado pelo uso do sensor Kinect V1, devido ao seu custo muito baixo, alta acessibilidade e características relativamente semelhantes

aos outros sensores. O ponto negativo deste equipamento é a baixa resolução da imagem de cor, todavia esse problema é contornado posteriormente ao serem utilizadas as imagens da câmera digital de resolução muito superior. O. Wasenmüller e D. Stricker [14] realizam uma comparação aprofundada entre ambos Kinect e chegam a conclusão de que a versão V1 apresenta melhor precisão para superfícies planas e especialmente sobre descontinuidades. Contudo possui menor acurácia a medida em que se aumenta a distância sensor-objeto. Diferentemente da versão V2, a informação de profundidade capturada pelo Kinect V1 é independente da cor do item a ser digitalizado e devido ao diferente método de funcionamento, não está sujeito ao efeito *multipath interference* de superfícies côncavas, como abordado por A. Bhandari et al. [15].

É importante mencionar que a manufatura do sensor Kinect V1 e seus sucessores foi oficialmente descontinuada em outubro de 2017, o que contribuiu para a redução no seu preço de aquisição, ao menos até o momento desta publicação. Por outro lado, a tecnologia ainda recebe suporte da Microsoft, e possui uma grande comunidade de desenvolvedores, mantendo sua relevância na comunidade acadêmica sobretudo em projetos de orçamento limitado. No entanto, é fácil inferir que o método proposto apresentará melhores resultados o quão melhores forem as nuvens de pontos capturadas pelo dispositivo, ou seja, maior densidade, menor ruído e maior acurácia de profundidade.

2. Metodologia da Reconstrução 3D

A metodologia é baseada nos *pipelines* genéricos de reconstrução 3D de trabalhos anteriores, como discutido por F. Bernardini e H. Rushmeier [5]. Esta metodologia é apresentada no diagrama da Figura 1 e foi aplicada na reconstrução 3D de todos objetos culturais selecionados do MAE.

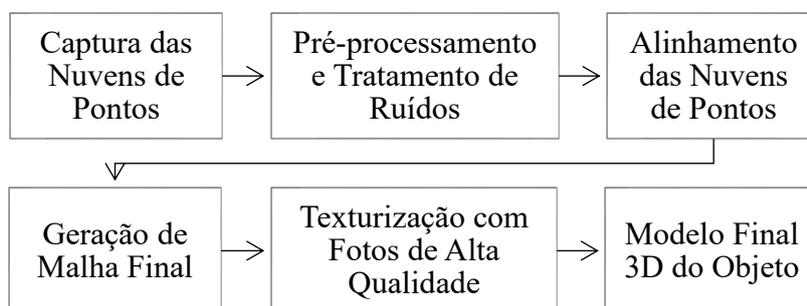


Figura 1. Diagrama esquemático dos processos de reconstrução 3D de um objeto.

2.1 Captura das Nuvens de Pontos

Antes do início do processo de captura, os objetos são posicionados sobre um apoio de acrílico de aproximadamente 15cm de altura, utilizado para evitar o contato direto com os objetos etnológicos ou paleontológicos. Além disso, o apoio é a base de rotação. Durante a captura, o *scanner* não é removido de sua posição inicial. A primeira imagem de profundidade registrada (também referenciada como “vista”) é definida como grau 0 (zero) e as posteriores são capturadas com referência a esta. Para a captura das vistas seguintes, o objeto é rotacionado pelo seu apoio de acrílico em cerca de 30° ou 45°, a depender do tamanho e complexidade da superfície. As novas capturas são realizadas até que se complete um ciclo de 360°. A rotação e digitalização são realizadas manualmente, e se for necessário, são capturadas vistas adicionais, como por exemplo as partes superior e inferior do objeto, usando rotações arbitrárias. A Figura 2 ilustra o esquema de captura.

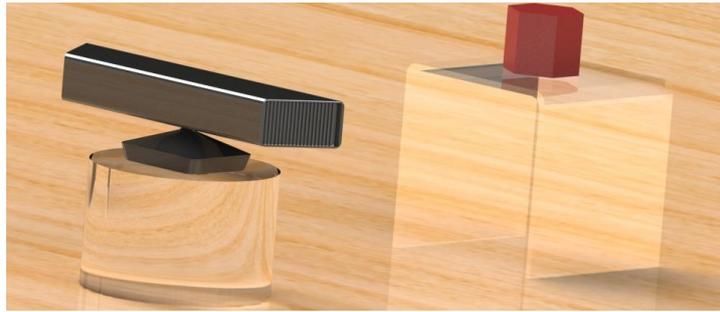


Figura 2. Esquema de captura de vistas com o Kinect.

Dois *softwares* de captura foram utilizados: *Kinect Fusion* [16] e um *software* próprio desenvolvido a partir das bibliotecas OpenCV [17] e OpenKinect [18]. O *Kinect Fusion* gera uma malha de triângulos a partir de uma vista da superfície do objeto capturada com Kinect. Este *software* aplica automaticamente um pré-processamento para remoção de ruídos e suavização da malha registrada, o que reduz detalhes presentes na superfície do objeto sem controle por parte do usuário. As imagens de profundidade capturadas pelo sensor, representam distâncias de captura entre a posição do Kinect e a superfície do objeto, pixel a pixel. Usando a câmera RGB integrada ao Kinect, é possível gerar uma imagem de cor para cada imagem de profundidade, com mesma resolução de 640x480 pixels.

O *software* próprio usa o sensor Kinect para capturar imagens de profundidade e cor, associando essas informações em uma nuvem de pontos (também referenciada somente como “nuvem”) em três dimensões, que representa a superfície do objeto. Além disso, possui a opção de capturar apenas um único frame por vista ou uma sequência de 30 frames em 1 segundo, de forma acumulada, o que aumenta a densidade da nuvem e os detalhes capturados, à medida que preenche, a cada novo frame, pixels não capturados em frames anteriores. Não obstante, as nuvens de pontos em alguns casos podem precisar de um pré-processamento e tratamento de ruídos, antes da etapa de alinhamento. Exemplos de vistas capturadas pelo *software* do *Kinect Fusion* e pelo *software* próprio podem ser observados na Figura 3.

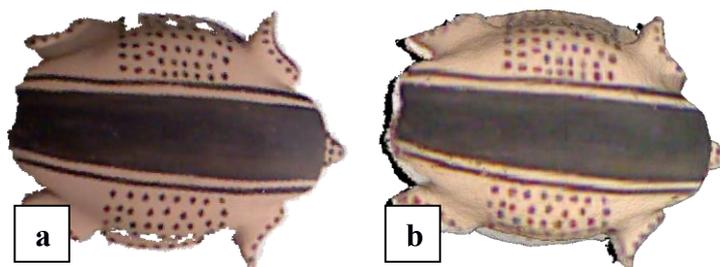


Figura 3. Imagens de profundidade capturadas. (a) Software Próprio. (b) Kinect Fusion.

O Kinect Fusion fornece uma nuvem de pontos conectados por arestas, formando faces com orientação definida e portanto com vetores normais. Já o *software* próprio captura nuvens de pontos, sem gerar malhas. Apesar disso, propicia uma maior flexibilidade sobre o método de captura, a exemplo de quantos frames serão utilizados em uma determinada vista, assim como evita pré-processamentos intrínsecos ao Kinect Fusion. Desta forma é possível o acesso à informação diretamente fornecida pelo sensor, sem procedimentos intermediários. Esses fatores resultam em um maior controle das informações de entrada no *pipeline* de reconstrução.

Os passos da subseção seguinte foram aplicados exclusivamente para as nuvens de pontos capturadas pelo *software* desenvolvido neste trabalho.

2.2 Pré-Processamento e Tratamento de Ruídos

Inicialmente as nuvens de pontos apenas possuem informação de profundidade e cor capturadas pelo Kinect. O vetor normal de cada ponto deve ser calculado para que seja possível a geração de uma malha corretamente orientada. No estado atual do *software* de captura, estas “normais” ainda não são computadas. Utilizamos então o *MeshLab* para realizar esse processo com a sua função integrada de geração de normais para nuvem de pontos, onde o sentido da normal aponta para “fora” do objeto. Um filtro de suavização com base nos pontos vizinhos foi aplicado para posteriormente obter melhores resultados do algoritmo de reconstrução de superfície, como ilustra a Figura 4.

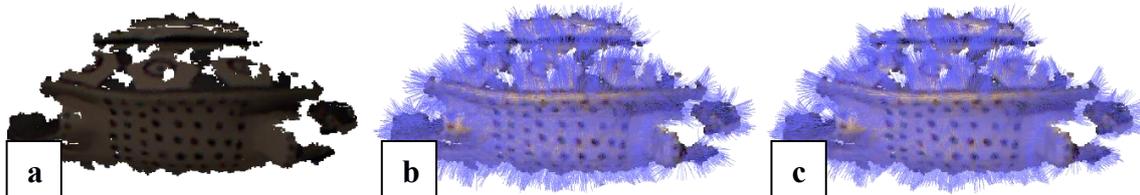


Figura 4. (a) Nuvem sem normais. (b) Normais computadas. (c) Suavização das normais.

Após essa etapa, tem início o tratamento de ruídos. Os ruídos capturados pelo sensor, são elementos externos que não fazem parte do modelo escaneado, como suportes e paredes; podem ser também informações de profundidade que se distanciam muito do objeto real, por exemplo, uma superfície plana no objeto real sendo capturada como uma superfície curvilínea pelo sensor; ou até mesmo regiões com cores incorretas. O tratamento e remoção destes ruídos é realizado de forma manual com *MeshLab*.

2.3 Alinhamento das Nuvens de Pontos

O alinhamento das vistas tem início após o tratamento de ruídos e também é realizado no *MeshLab*. Foram testados dois métodos de alinhamento. Um método é o alinhamento manual, que consiste no posicionamento visual das nuvens, sem auxílio de algoritmos de alinhamento, portanto o usuário aplica operações de rotação e translação nos vértices de cada vista. O alinhamento por pontos, por sua vez consiste em selecionar ao menos 4 (quatro) pontos de correspondência entre duas nuvens distintas, ou seja, pontos que estão na mesma região do espaço (no referencial do modelo alinhado) ou suficientemente próximos. Ambos os métodos são ilustrados na Figura 5.

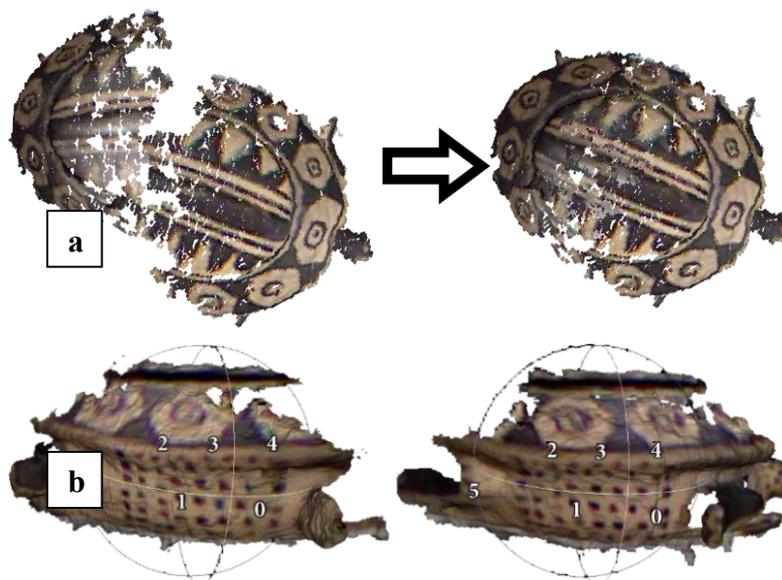


Figura 5. Técnicas de alinhamento aplicadas. (a) Manual. (b) Por pontos.

Finalmente é aplicado um algoritmo de minimização das distâncias, conhecido como *Iterative Closest Point* (ICP), que realiza um alinhamento mais fino, também abordado por S. Rusinkiewicz e M. Levoy [19]. O algoritmo deve ser executado após o alinhamento inicial das vistas, caso contrário o método não converge para um alinhamento final. O algoritmo tenta minimizar a distância média entre pontos próximos aplicando matrizes de rotação e translação a cada iteração, dessa forma, a depender da posição inicial das nuvens de pontos, o resultado do alinhamento será diferente.

Após alguns testes, o alinhamento manual das nuvens de pontos se mostrou mais rápido e prático, visto que mesmo no caso da captura acumulada de 30 frames, a densidade das nuvens não foi muito alta, permanecendo em uma faixa de 600 a 25 mil pontos por vista. Essa baixa densidade impossibilitou o alinhamento por pontos devido à dificuldade em se encontrar pontos correspondentes entre as nuvens. Visando a convergência do algoritmo de ICP, o alinhamento manual deve ser realizado de forma verossímil ao objeto real. Carregam-se previamente ao alinhamento, imagens definidas como *raster layers* no MeshLab e o alinhamento manual é realizado com base nestas imagens, buscando manter as nuvens dentro dos limites das bordas deste objeto, como exemplifica a Figura 6.

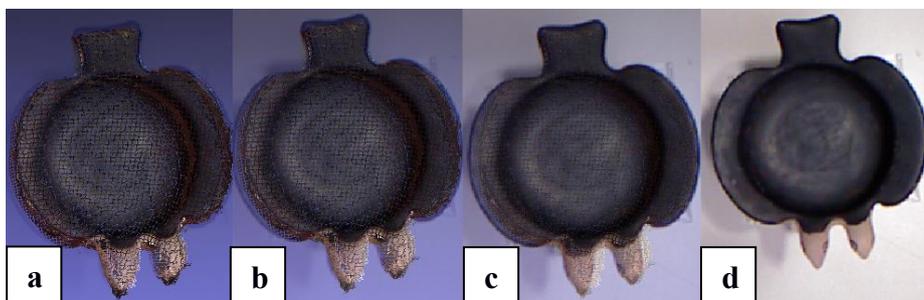


Figura 6. Uso do raster layer para auxiliar durante o processo de alinhamento. Variação da opacidade do raster layer sobre as nuvens de pontos: (a) 10%. (b) 25%. (c) 50%. (d) 100%.

Na Figura 6a observa-se as nuvens de pontos em processo de alinhamento. Já na Figura 6d é possível visualizar apenas a imagem que foi utilizada como *raster layer* guia para o alinhamento, na vista superior da peça. É necessário o uso de diversos *raster layers* em diferentes ângulos para melhorar a verossimilhança do modelo 3D final.

Apesar do alinhamento com o auxílio dos *raster layers* ainda podem ocorrer alguns erros ao aplicar o ICP, podendo convergir para um alinhamento final que seja visualmente distante do objeto real. Foi verificado que é necessária razoável sobreposição de pontos entre as vistas para a convergência mais próxima do modelo real. Esse problema é minimizado para nuvens de pontos com alta densidade.

Outro problema é identificado em peças simétricas e com poucos detalhes, em razão de grande parte da geometria da superfície do objeto ser semelhante, o algoritmo pode convergir para muitas soluções, que não são necessariamente as melhores representações do objeto real. Contudo, percebemos que é possível contornar esse problema com a “clonagem” de nuvens de pontos onde é desejável que os detalhes sejam mais destacados e preservados, dessa forma o ICP tende a preservar os detalhes das zonas clonadas durante o alinhamento.

À direita da Figura 7a foi realizada a clonagem das nuvens de pontos onde os detalhes estão mais presentes e são mais delicados, como as “cabeças” e “cauda” do recipiente em formato de pássaro. Após a execução do ICP o formato de cabeça e cauda é mais facilmente observável nessas regiões e a geometria é mais próxima do objeto real.

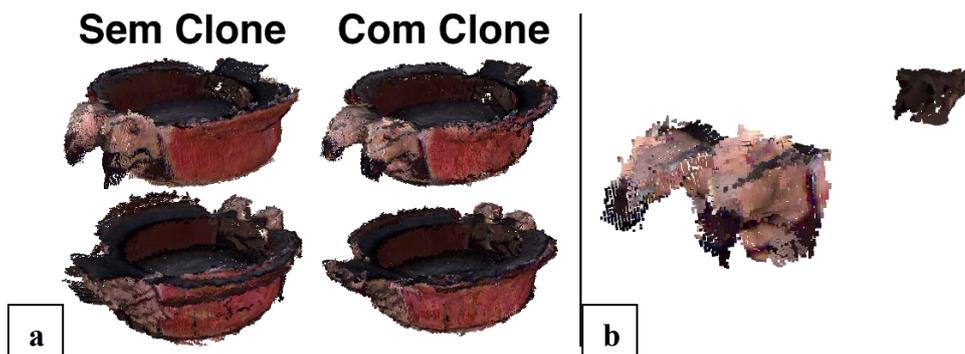


Figura 7. (a) Alinhamento e técnica de clonagem. (b) Detalhes clonados da Figura 7a.

Após o algoritmo de ICP, a distância quadrática média do alinhamento (resultado de saída do algoritmo) sem o uso de clones foi de 1.109mm e com clone de 1.017mm, apresentando uma melhoria de aproximadamente 8,30%. Ao usar esse método, é necessário clonar todos os detalhes relevantes, estando sujeito à interpretação do usuário.

2.4 Geração da Malha Final

Posteriormente ao alinhamento, as nuvens de pontos são unidas e uma malha final é gerada com o emprego do algoritmo *Poisson Surface Reconstruction* do *MeshLab* por M. Kazhdan, M. Bolitho e H. Hoppe [20]. Este método tenta encontrar uma superfície na qual o gradiente de seus pontos seja o mais próximo das normais dos vértices de todas as nuvens de pontos visíveis. Durante esse processo foram testados diferentes parâmetros do algoritmo, como *Reconstruction Depth*, *Minimum Number of Samples* e *Interpolation Weight*. Os valores ótimos para esses parâmetros mudam para cada objeto. A Figura 8 demonstra o resultado da execução do algoritmo para uma nuvem de pontos.

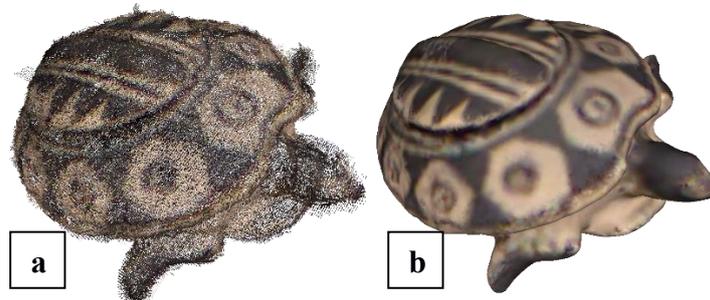


Figura 8. (a) Nuvens de pontos alinhadas. (b) Superfície reconstruída após Poisson.

2.5 Texturização com Fotos de Alta Resolução

Em paralelo ao processo de captura das vistas, fotos de alta resolução foram tiradas com uma câmera digital semiprofissional. Esta câmera capturou imagens de cor com uma resolução de 4288x3216 pixels, sendo esta resolução 6.7 vezes superior às capturadas com Kinect de 640x480 pixels. Como resultado facilmente observável, uma melhor qualidade visual do objeto foi atingida.

Para a texturização da malha final, as fotos em alta resolução são utilizadas como *raster layers*, da mesma maneira visualizada na Figura 6, então é gerada uma parametrização UV para associar coordenadas de textura para cada vértice da malha. Finalmente, um atlas de textura é criado e preenchido com as cores das fotos de alta resolução, como demonstra a Figura 9.

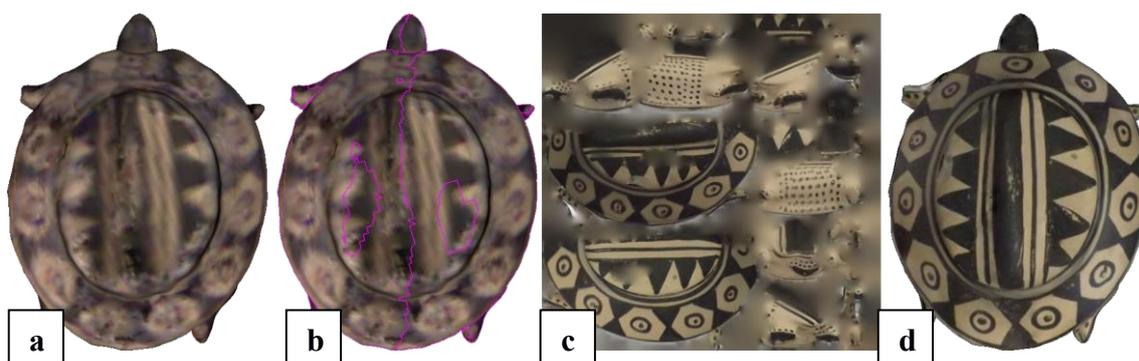


Figura 9. (a) Malha com textura do Kinect. (b) Malha com parametrização UV (c) Atlas de textura final. (d) Malha com textura final

3. Resultados

Para o procedimento de captura das vistas, utilizamos um sensor Xbox 360 Kinect V1 modelo 1414; uma câmera semiprofissional Fujifilm SI300; um notebook de processador I7-7500U, 8Gb de memória RAM, placa gráfica Nvidia GeForce 940M, com sistema operacional Windows 10; um apoio de acrílico para o objeto e equipamento de iluminação para auxiliar na captura de cor pelo Kinect e a câmera digital. Finalmente, o *pipeline* de reconstrução 3D foi executado usando as ferramentas nativas do *MeshLab*.

3.1 Modelos de Teste

Durante os testes, diversos objetos do MAE foram escaneados e reconstruídos. A Tabela 2 os lista e descreve.

Tabela 2. Listagem e descrição dos modelos escaneados e reconstruídos.

OBJETOS ETNOLÓGICOS INDÍGENAS:	
Nome de Referência	Descrição
Tartaruga	Panela em formato de tartaruga
Vaso Rugoso	Parte de vasilhame rugoso
Vaso de Aprendiz	Vasilhame utilizado por um aprendiz em treinamento de olaria
Cachimbo Cônico	Cachimbo de formato cônico
Cachimbo em J	Cachimbo em formato de J
Tembetá	Perfurador labial
Onça	Tigela com formato de onça
Pássaro	Tigela com formato de pássaro
OBJETOS PALEONTOLÓGICOS:	
Raspador	Raspador de pedra lascada
Cunha	Cunha de machado

3.2 Testes de Acumulação

Como comentado na seção 2.1, o programa pode realizar uma captura de 1 ou 30 frames por vista. Para alguns objetos, foram realizadas capturas de imagem de profundidade com ambos métodos para fins de comparação da densidade das nuvens de pontos. Os resultados são avaliados na Tabela 3.

Tabela 3. Comparativo entre métodos cumulativo e instantâneo.

Nome do Objeto	Número de Vistas	Número Total de Vértices		Diferença Relativa
		Método Cumulativo (30 Frames)	Método Instantâneo (1 Frame)	
Onça	14	374.372	262.710	42,50%
Pássaro	14	370.472	281.757	31,49%
Raspador	12	105.422	89.505	17,78%

Tabela 4. Quantidade de vértices iniciais e finais selecionados de cada modelo.

Nome do Objeto	Identificador ID	Número de Vistas	Vértices Iniciais	Vértices Selecionados	Software Utilizado
Tartaruga	1	11	2.388.819	2.353.553	Kinect Fusion
Vaso Rugoso	2	6	135.801	129.441	Kinect Fusion
Vaso Aprendiz	3	15	2.216.237	2.114.472	Kinect Fusion
Cachimbo Cônico	4	9	338.463	333.726	Kinect Fusion
Tembetá	5	5	73.596	70.502	Kinect Fusion
Cunha	6	6	254.801	252.376	Kinect Fusion
Tartaruga	7	15	256.022	249.613	Próprio
Cachimbo em J	8	8	6.852	5.519	Próprio
Onça	9	14	151.576	115.599	Próprio
Pássaro	10	14	151.981	144.112	Próprio
Raspador	11	12	77.090	59.141	Próprio

A Tabela 5 mostra dados relacionados à geração da malha final usando o algoritmo de reconstrução de Poisson e ao processo de mapeamento de texturas. Os “vértices Poisson” e “parâmetros Poisson” mostrados representam, respectivamente, o número de vértices da malha gerada e os valores padrões escolhidos no *MeshLab* para executar o algoritmo de reconstrução de Poisson: *Reconstruction Depth (RD)*; *Minimum Number of Samples (MNS)*; e *Interpolation Weigh (IW)*. O número de *rasters* representa a quantidade de imagens utilizadas na criação do atlas de textura. Todos os modelos possuem atlas de textura de dimensões 2048x2048 pixels.

Tabela 5. Valores utilizados para os parâmetros de Poisson e textura.

Nome do Objeto	Parâmetros Poisson			Vértices Poisson	Número de Rasters	Software Utilizado
	RD	MNS	IW			
Tartaruga	12	180	0	39.305	9	Kinect Fusion
Vaso Rugoso	12	8	0	41.068	6	Kinect Fusion
Vaso Aprendiz	12	45	0	101.765	12	Kinect Fusion
Cachimbo Cônico	12	68	0	13.857	6	Kinect Fusion
Tembetá	10	40	0	6.297	6	Kinect Fusion
Cunha	13	4	0	13.476	6	Kinect Fusion
Tartaruga	12	8	0	92.372	9	Próprio
Cachimbo em J	9	4	0	4.837	8	Próprio
Onça	14	18	0	15.695	8	Próprio
Pássaro	14	40	0.05	13.218	9	Próprio
Raspador	12	2	0	67.570	9	Próprio

4. Análise dos Resultados

Podemos comparar visualmente os modelos reconstruídos em 3D e os reais, contudo essa abordagem não permite avaliações quantitativas das reconstruções, desta forma implementamos um método que contabiliza o quanto a reconstrução se aproxima do objeto real em função da silhueta e das cores.

4.1 Análise de Silhueta

O método empregado compara as projeções 2D do modelo 3D sobre as fotos em alta resolução que foram utilizadas para mapear as texturas. Sendo assim, é possível avaliar pixel a pixel, para cada camada de rasterização o quanto a geometria 3D se distancia da real, conforme demonstra a Figura 11.

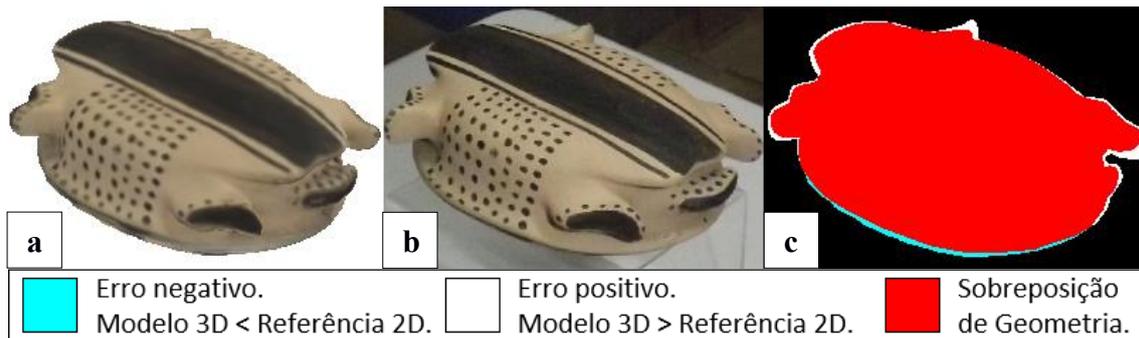


Figura 11. (a) Modelo 3D. (b) Imagem do objeto utilizada como *raster layer* e referência 2D. (c) Projeção 2D do modelo 3D sobre o objeto real.

As imagens utilizadas para fim de comparação são as mesmas que foram utilizadas como *raster layer* para texturização do modelo 3D, portanto possuem alta resolução, minimizando assim o efeito de pixelização das bordas.

É importante atentar para que o *Field of View* utilizado durante a visualização do modelo 3D no *MeshLab* seja o mesmo que foi empregado durante a captura das imagens com a câmera de alta resolução para evitar distorções durante a comparação. Pequenos erros de alinhamento são inevitáveis, visto que as imagens não possuem informação de profundidade para contribuir ao posicionamento, contudo esse tipo de erro possui influência de menos de 1% no resultado final deste método.

O método ignora os pixels pretos, pois estes não fazem parte do modelo 3D ou real. Após a contagem de pixels, o método calcula a quantidade de cada tipo de erro em porcentagem e o resultado para todas as malhas geradas com o software próprio é apresentado no Gráfico 1.

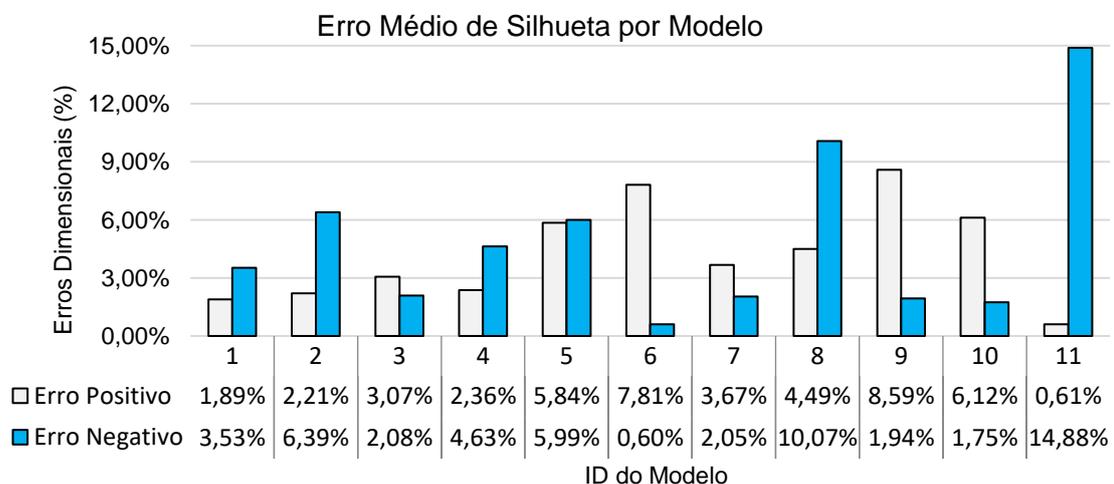


Gráfico 1. Percentagem dos erros do tipo positivo e negativo para cada modelo. Escala de 0 a 15%.

Por fim, podemos verificar os erros médios das reconstruções, assim como os erros para ambos os *softwares*, como observamos na Tabela 6.

Tabela 6. Erros médios de silhueta por software.

	Kinect Fusion	Próprio
Erro Positivo Médio	3,86%	4,61%
Erro Negativo Médio	3,87%	4,24%

Verificamos que a percentagem média de erros positivos e negativos, são relativamente baixas, geralmente abaixo de 10%, exceto para o caso do modelo 11 que por ser predominantemente plano e polido, prejudica a captura da superfície pelo sensor, especialmente nas regiões das bordas. A disposição do Kinect aproximadamente no mesmo nível do objeto também dificulta a aquisição da geometria, visto que o ângulo entre a normal da superfície do objeto e o sensor é muito alto, como ilustra a Figura 12.

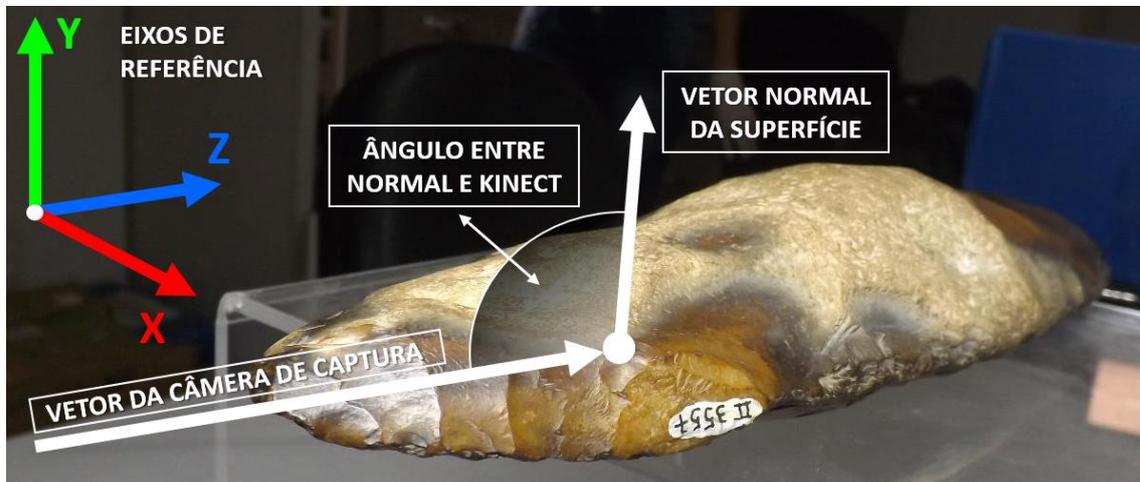


Figura 12. Foto do modelo 11 (Raspador indígena) e vetores.

Ao final, foram analisadas 93 amostras para todos os 11 modelos. Uma visão geral pode ser observada na Figura 13.

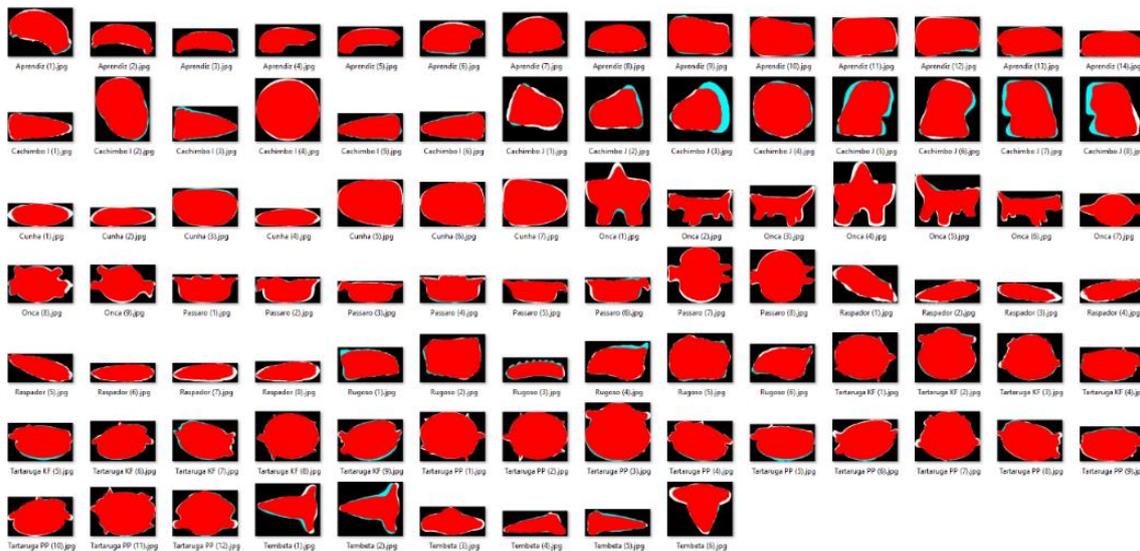


Figura 13. Visão geral da análise de silhueta dos modelos.

4.2 Distância de Cores

Na seção anterior foi avaliada a diferença na geometria do modelo em relação às fotos utilizadas para o mapeamento das texturas. Nesta seção é avaliada a distorção de cor da textura mapeada. Para tal, utilizamos a Distância Euclidiana para calcular a distância entre as cores de pixels correspondentes do modelo 3D gerado e objeto real. A fórmula utilizada pode ser representada genericamente pela Equação 2.

$$\frac{\sum_{(j,k)=pixel(0,0)}^{pixel(n,m)} \sqrt{\sum_{i=1}^{n^{\circ}canais} \left(Canal_i pixel_{(j,k)}(3D) - Canal_i pixel_{(j,k)}(Real) \right)^2}}{n \cdot m}$$

Equação 2. Cálculo da distância euclidiana das cores pixel a pixel.

Para cada pixel de cada canal, é calculado o quadrado da diferença entre a cor mapeada no modelo 3D e a cor do pixel correspondente presente na imagem de rasterização capturada com a câmera digital de alta resolução. Então aplica-se a raiz quadrada a este resultado e divide-se pelo número de pixels (“n” linhas por “m” colunas). Após isso soma-se todos os resultados para cada pixel, resultando na média das distâncias euclidianas para a imagem.

Ao ser codificada a Equação 2 e fornecer como entrada ao *software* as imagens de comparação, obtemos como saída os resultados apresentados pelo Gráfico 2, considerando a distância das cores para os modos de imagem BGR (*Blue; Green; Red*) e LAB (*Lightness, Channel A, Channel B*). O modo de imagem BGR é muito comum em equipamentos eletrônicos, contudo seu espaço de cores é não linear. Já o modo LAB possui espaço de cores linear.

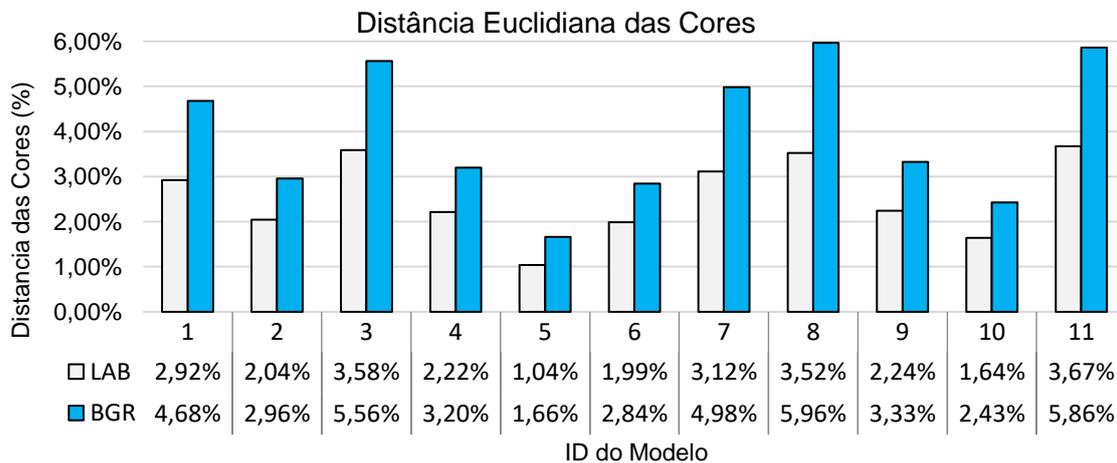


Gráfico 2. Distâncias médias das cores entre os modelos 3D e os objetos reais. Escala de 0 a 6%.

Há um pequeno desvio nas cores, sendo abaixo de 6% para o modo BGR e abaixo de 4% no LAB. A distância euclidiana média para o modo de imagem BGR foi de 3,95% e para LAB 2,54%. Por fim, 186 arquivos base foram utilizados para cálculo, como apresentado na Figura 14.



Figura 14. Visão geral da análise de cor dos modelos.

5. Conclusões

Com este trabalho desenvolvemos uma metodologia capaz de reconstruir e gerar modelos 3D com boa qualidade geométrica e visual. Valiosos objetos culturais do Museu de Arqueologia e Etnologia da UFBA foram escaneados e reconstruídos virtualmente, utilizando apenas o sensor Kinect 360 e uma câmera digital de alta resolução.

Adicionalmente, foram implementadas metodologias de comparações numéricas pixel a pixel dos modelos virtuais com os objetos reais, avaliando aspectos de geometria e aparência. Os resultados apresentados indicam reconstruções com desvios de geometria menores que 5% e distâncias de cores menores que 4%, em média. A partir disso é constatado que reconstruções 3D de objetos são plausíveis de serem realizadas com equipamentos de baixo custo, fácil disponibilidade em mercado e utilizando *softwares* gratuitos.

Certamente o sensor Kinect V1 apresenta limitações como foi observado no modelo 11, onde grande parte do contorno do objeto não foi capturada pelo sensor. No entanto a maioria das reconstruções se mostraram visualmente e geometricamente próximas dos objetos reais, representando-os de forma muito satisfatória. Por outro lado, a possibilidade do uso desses modelos como referência para possíveis reconstruções fiéis dos itens reais ainda deve ser estudada mais a fundo.

Este trabalho permitiu ampliar o acervo virtual do MAE, possibilitando uma preservação inicial das peças reconstruídas. Não obstante, ainda há espaço para melhorias. O processo de captura e aquisição de informação poderia ser automatizado, assim como os passos iniciais da reconstrução, a exemplo do alinhamento como evidenciam V. Reis e L. Jorge [11] e B. Adlkhast e O. Manikhi [12]. A automatização poderia resultar em modelos 3D fiéis aos reais, principalmente em geometria, provavelmente com menores erros sistemáticos.

A quantidade e qualidade dos pontos capturados também poderia ser aumentada ao usar técnicas de Super-Resolution como apresentadas por J. Silva, L. Gomes, K. Agüero, O. Bellon e L. Silva [21]. Ruídos capturados pelo sensor poderiam ser minimizados com o uso de técnicas como apresentadas por M. Jaiswal, J. Xie e M. Sun [22]. Ademais, C. Nguyen et al. [23] logra ótimos resultados em termos de redução de ruídos e melhoria na

qualidade final dos modelos a partir da modelagem da distribuição axial e lateral da captura de ruídos pelo Kinect V1.

6. Agradecimentos

Os autores gostariam de agradecer ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e a Fundação de Amparo à Pesquisa do Estado da Bahia (FAPESB) pelo auxílio financeiro fornecido a este projeto.

Os autores também gostariam de agradecer a todos os funcionários do MAE por sua colaboração e suporte em termos de infraestrutura, conhecimento, manipulação dos artefatos culturais, além da escolha de objetos interessantes para o escaneamento.

Outro colaborador a quem gostariam de agradecer é Gustavo Mota Simões pelo auxílio e tempo dedicado ao projeto.

7. Referências

- [1] M. Levoy et al. (2000). *The Digital Michelangelo Project: 3D Scanning of Large Statues*. SIGGRAPH, Proceedings of the 27th annual conference on Computer graphics and interactive techniques.
- [2] L. Gomes, L. Silva e O. Bellon (2014). *3D Reconstruction Methods for Digital Preservation of Cultural Heritage: A Survey*. Journal Pattern Recognition Letters; Vol. 50.
- [3] MeshLab. *The open source system for processing and editing 3D triangular meshes*. www.meshlab.net (acessado em maio de 2018)
- [4] Museu de Arqueologia e Etnologia (MAE) da UFBA. www.mae.ufba.br (acessado em maio de 2018)
- [5] F. Bernardini e H. Rushmeier (2002). *The 3D Model Acquisition Pipeline; Computer Graphics Forum vol 2; Blackwell Publishing Ltd*.
- [6] Y. Cui et al. (2013). *Algorithms for 3D Shape Scanning with a Depth Camera*. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 35, N° 5.
- [7] T. Schenk (2005). *Introduction to Photogrammetry. Department of Civil and Environmental Engineering and Geodetic Science. The Ohio State University*.
- [8] A. Utomo e C. Wibowo (2017). *3D Reconstruction of Temples in the Special Region of Yogyakarta by Using Close-Range Photogrammetry*. Seminar Nasional Teknologi Informasi dan Multimedia.
- [9] P. Raimundo, K. Agüero e A. Apolinário Jr. (2018). *Low-cost 3D reconstruction of cultural heritage artifacts*. Revista Brasileira de Computação Aplicada. Vol. 10, N° 1, p. 66–75.
- [10] C. Wu-Wei (2016). *Digital Heritage: Digital Sculpting, Cyber-Archiving and Education. Pedagogic Project for Digital heritage: Digital Sculpting, Projection Mapping and Beyond*.

- [11] V. Reis e L. Jorge (2013). *Reconstrução Digital de Objetos em 3D por Nuvem de Pontos Utilizando o Kinect*. 176 Rev. Científica Eletrônica UNISEB, Ribeirão Preto, Vol. 1, N° 2, p. 176-191.
- [12] B. Adlkhast e O. Manikhi (2013). *A 3D Object Scanner. An approach using Microsoft Kinect*. Halmstad University, Master's Thesis in Information Technology 120 ECTS.
- [13] M. Zollhöfer et al. (2014). *Low-Cost Real-Time 3D Reconstruction of Large-Scale Excavation Sites using an RGB-D Camera*. Eurographics Workshops on Graphics and Cultural Heritage (2014), pp. 1–10
- [14] O. Wasenmüller e D. Stricker (2016). *Comparison of Kinect v1 and v2 Depth Images in Terms of Accuracy and Precision*. German Research Center for Artificial Intelligence (DFKI)
- [15] A. Bhandari et al. (2014). *Resolving Multipath Interference in Kinect: An Inverse Problem Approach*. Massachusetts Institute of Technology, Cambridge MA, USA.
- [16] Kinect Fusion. *Kinect for Windows Developer Toolkit v1.8*. msdn.microsoft.com/en-us/library/dn188670.aspx (acessado em maio de 2018)
- [17] OpenCV. *Open Source Computer Vision Library*. www.opencv.org (acessado em maio de 2018)
- [18] OpenKinect. *A userspace driver for the Microsoft Kinect*. www.github.com/OpenKinect/libfreenect (acessado em maio de 2018)
- [19] S. Rusinkiewicz e M. Levoy (2001). *Efficient variants of the ICP algorithm*. In *Third Int. Conf. on 3D Digital Imaging and Modeling (3DIM) IEEE Comp. Soc.*
- [20] M. Kazhdan, M. Bolitho e H. Hoppe (2006). *Poisson surface reconstruction*. *Proceedings of the fourth Eurographics symposium on Geometry processing*.
- [21] J. Silva, L. Gomes, K. Agüero, O. Bellon e L. Silva (2013). *Real-time acquisition and super-resolution techniques on 3D reconstruction*. *IEEE International Conference on Image Processing (ICIP)*, p. 2135.
- [22] M. Jaiswal, J. Xie e M. Sun (2014). *3D Object Modeling with a Kinect Camera*. Department of Electrical Engineering, University of Washington, Seattle, USA. APSIPA.
- [23] C. Nguyen et al. (2012). *Modeling Kinect Sensor Noise for Improved 3D Reconstruction and Tracking*. *Second Joint 3DIM/3DPVT Conference: 3D Imaging, Modeling, Processing, Visualization & Transmission*