



Automatização de teste de software no domínio de reconstrução 3D

Software testing automation in the 3D reconstruction domain

Luan William Marques Alvares¹, Cléber Gimenez Corrêa²

RESUMO

O teste de *software* visa a qualidade e a confiabilidade do produto de *software*. Adicionalmente, testar pode ser uma atividade repetitiva, seguindo padrões de execução, o que requer a automatização. Há sistemas que são considerados complexos, como os sistemas para reconstrução tridimensional (3D) a partir de imagens, dificultando o teste. Dessa forma, o objetivo do trabalho é criar uma forma de automatizar o teste de sistemas dessa natureza, empregando a técnica de *Record & Playback*, para gravar as saídas esperadas e comparar posteriormente com as saídas obtidas a partir das mesmas entradas (imagens). As saídas ou modelos 3D gerados são arquivos com elementos (pontos ou coordenadas 3D formadas por vértices, faces formadas pelas coordenadas ou triângulos e texturas), exigindo adaptações para a comparação, como divisão dos elementos em arquivos separados e a ordenação das informações desses arquivos. Diversos sistemas foram utilizados, uma ferramenta foi utilizada para a comparação e os resultados mostraram que os tamanhos dos arquivos exigem alto processamento; no entanto, a paralelização é possível.

PALAVRAS-CHAVE: Automatização; Reconstrução 3D; Teste de Software.

ABSTRACT

Software testing aims at the quality and reliability of the software product. Additionally, tests can be a repetitive activity, following execution patterns, which requires automation. There are systems that are considered complex, such as systems for three-dimensional (3D) reconstruction from images, making testing difficult. Therefore, the objective of the work is to create a way to automate the testing of systems of this nature, using the Record & Playback technique, to record the expected outputs and later compare them with the outputs obtained from the same inputs (images). The outputs or 3D models generated are files with elements (points or 3D coordinates formed by vertices, faces formed by coordinates or triangles and textures), requiring adaptations for comparison, such as dividing the elements into separate files and ordering the information in these files. Several systems were used, a tool was used for comparison and the results showed that file sizes require high processing; however, parallelization is possible.

KEYWORDS: Automation; 3D reconstruction; Software Testing.

INTRODUÇÃO

O teste de *software* visa a qualidade e a confiabilidade de sistemas (DELAMARO; MALDONADO; JINO, 2016), o que é fundamental no mundo moderno, visto que esses sistemas estão presentes em todos os lugares.

¹ Bolsista do(a) Conselho Nacional de Desenvolvimento Científico e Tecnológico - Cornélio Procópio, PR, Brasil. E-mail: luan.2014@alunos.utfpr.edu.br. ID Lattes: <http://lattes.cnpq.br/3583145129752061>.

² Docente no Curso Engenharia de Software/Departamento Acadêmico de Computação. Universidade Tecnológica Federal do Paraná, Cornélio Procópio, PR, Brasil. E-mail: clebergimenez@utfpr.edu.br. ID Lattes: lattes.cnpq.br/0521761025000380.



O teste possibilita a revelação de falhas, que são indícios de defeitos ou implementações incorretas ou inconsistentes com o esperado pelo usuário, contribuindo para a entrega de produtos de alta qualidade (AMMANN; OFFUTT, 2016). Isso se aplica a todos os tipos de sistemas, como os sistemas para reconstrução tridimensional (3D) a partir de imagens, empregados em diversas áreas, como agricultura, educação, indústria e entretenimento.

Entretanto, testar é uma atividade repetitiva e que deve seguir procedimentos, o que a torna suscetível a erros dos testadores e a necessidade de tempo. A automatização é uma forma de superar essas dificuldades. Esta pesquisa visou a automatização do teste de sistemas de reconstrução 3D, ressaltando que na literatura esse assunto é pouco explorado.

OBJETIVO

O objetivo é a automatização do teste de sistemas de reconstrução 3D, empregando a técnica de *Record & Playback*, para gravar as saídas esperadas e comparar posteriormente com as saídas obtidas a partir das mesmas entradas (imagens). As saídas são os modelos 3D; as entradas e as saídas formam os casos de testes para testar versões liberadas posteriormente do *software*.

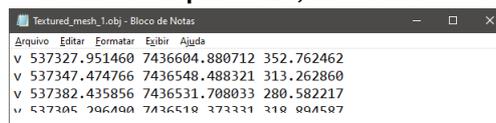
METODOLOGIA

A pesquisa foi organizada em etapas. A primeira etapa foi selecionar *softwares* capazes de aplicar a fotogrametria em imagens, ou seja, realizar a reconstrução 3D a partir de fotografias ou vídeos. Em seguida, a etapa foi buscar na Internet e criar *datasets* com imagens e vídeos de entrada para os *softwares*. Alguns *softwares* testados são capazes de importar vídeo e extrair *frames* ou imagens do vídeo.

As reconstruções foram realizadas utilizando os *datasets* e os *softwares*, gerando arquivos no formato OBJ (*Object File Wavefront 3D*) que são os modelos 3D ou saídas. As entradas e saídas devem compor os denominados casos de teste, empregados para testar os sistemas.

Os arquivos OBJ possuem suas particularidades, como uma referência para um arquivo MTL (Material), que contém as texturas, normalmente no formato JPG, bem como os elementos do modelo 3D, gravadas em linhas do arquivo e contendo vértices (v), que formam as coordenadas 3D, vértices normais (vn), vértices texturizados (vt) e faces ou triângulos (f), formados pela ligação entre os vértices. Cada linha tem a sigla que representa o tipo do elemento e os respectivos valores (Figura 1).

Figura 1 – Trecho de arquivo OBJ, mostrando os vértices (v)



Fonte: Autoria própria

Normalmente, os *softwares* produzem modelos 3D com muitas informações, inviabilizando a manipulação do arquivo devido a quantidade de memória e processamento necessários. Uma questão de pesquisa foi elaborada: é possível automatizar o teste de *software*, mesmo para *softwares*



comerciais, empregando a técnica de *Record & Playback* e usando o conteúdo dos arquivos dos modelos 3D?

A forma de testar é por meio da leitura e a comparação do conteúdo de ambos os arquivos, linha a linha, considerando que cada linha normalmente possui três informações, como por exemplo, valores para as coordenadas x , y e z . Porém, no caso de arquivos OBJ, as linhas podem estar em posições diferentes em dois arquivos para o mesmo objeto e há uma grande quantidade de linhas.

Dessa forma, comparar todas as linhas de um arquivo com todas as linhas do outro arquivo é inviável. Ao se analisar a estrutura lógica, a complexidade do algoritmo de comparação é de tempo fatorial, inviável para arquivos com milhões de linhas. Para resolver esse problema foram adotadas estratégias: (1) Dividir o arquivo OBJ em arquivos de texto contendo apenas seus respectivos elementos, ou seja, os arquivos de vértices somente com os elementos cuja primeira letra da linha fosse “v”; o mesmo para faces com letra “f”, vértices texturizados com letras “vt” e vértices normais com letras “vn”; (2) Ordenar cada um dos arquivos separados, de forma que os elementos estejam em ordem crescente. Como a complexidade dos algoritmos de ordenação também é relevante, o método escolhido foi o *Merge Sort*, devido a sua complexidade no melhor e no pior caso ser $\Theta(n \log n)$, em que n representa a quantidade de linhas; (3) Comparar os arquivos relacionados já ordenados, finalizando a execução quando uma diferença é encontrada, o que é considerada uma revelação de falha.

Também foi criada uma ferramenta com interface gráfica em linguagem de programação Java para implementar automaticamente as estratégias, com o usuário informando o arquivo e onde está armazenado. Os dispositivos utilizados para captura, processamento e visualização das informações foram: *Smartphone Samsung A22*, que possui quatro câmeras traseiras de 48MP, 28MP, 2MP e 2MP, com resolução máxima de 8000×6000 pixels, para a criação dos *datasets*, com fotos e vídeos; computador *desktop* com processador AMD *Ryzen 5 5600g* (gráfico integrado), 16GB de RAM, sem placa gráfica dedicada.

Os *softwares* que implementam a técnica de fotogrametria para a reconstrução 3D selecionados foram: 3DF Zephyr da empresa 3Dflow (3DFLOW, 2023), Agisoft Metashape da empresa Agisoft LLC (AGISOFT LLC, 2023) e Autodesk Recap Photo da empresa Autodesk (AUTODESK, 2023).

Os *datasets* utilizados foram: o *South Building* (ZACH, 2023), um prédio disponibilizado via Internet; automóvel, vaso de flor com fundo claro; vaso de flor com fundo escuro, criados para a pesquisa. O automóvel foi selecionado por ser um objeto menor do que o prédio. O vaso de flor foi escolhido por ser menor do que o automóvel, bem como para estudar a viabilidade da reconstrução de plantas, uma demanda da Empresa Brasileira de Pesquisa Agropecuária (Embrapa), que visa a disponibilização da planta de soja em três dimensões e em diferentes estádios fenológicos (estágios de desenvolvimento da planta). Os fundos claro e escuro no caso do vaso de flor é para verificar a influência da iluminação. Nas Figuras 2, 3, 4, 5, 6 e 7 são apresentadas imagens do *dataset* vaso de flor, que foi utilizado em forma de fotografia e vídeo.

Figura 2 – Dataset: automóvel



Fonte: Autoria própria

Figura 3 – Dataset: imagem do prédio South Building



Fonte: (ZACH, 2023)

Figura 4 – Dataset: imagem de vídeo do vaso de flor



Fonte: Autoria própria

Figura 5 – Dataset: fotografia do vaso de flor



Fonte: Autoria própria

Figura 6 – Dataset: imagem do vaso de flor em fundo escuro **Figura 7 – Dataset: imagem de vídeo do vaso de flor em fundo escuro**



Fonte: Autoria própria



Fonte: Autoria própria

RESULTADOS

Nas Figuras 8, 9, 10 e 11 são mostradas as reconstruções obtidas por meio do *software 3DF Zephyr*. Na Tabela 1 são apresentadas as quantidades de elementos dos modelos gerados usando esse *software*. Mais de um teste foi realizado para o mesmo *software* e o mesmo *dataset*, para verificar se a reconstrução é determinística. Nas Figuras 12, 13, 14 e 15 são mostradas as reconstruções obtidas por meio do *software Agisoft Metashape*. Na Tabela 2 são apresentadas as quantidades de elementos dos modelos gerados em dois testes usando esse *software*.

Tabela 1 – Comparação das reconstruções: Automóvel e Vaso de flor

Atributo	Automóvel		Vaso de Flor	
	Teste um	Teste dois	Teste um	Teste dois
Tamanho do arquivo (MB)	33.5	33.5	20.1	19.9
Tempo de processamento	9m:51s	12m:35s	3m:37s	4m:14s
Vértices (v)	151968	151968	102422	101465
Vértices Normais (vn)	151968	151968	102422	101465
Vértices Texturizados (vt)	228413	228413	120303	119046
Faces/Triângulos (f)	301808	301808	204284	202462

Fonte: Autoria própria.

Figura 8 – Reconstrução do automóvel - Primeiro teste **Figura 9 – Reconstrução do automóvel - Segundo teste**



Fonte: Autoria própria



Fonte: Autoria própria

Figura 10 – Reconstrução do vaso de flor - Primeiro teste com Fotografias **Figura 11 – Reconstrução do vaso de flor - Segundo teste com Fotografias**



Fonte: Autoria própria



Fonte: Autoria própria

Figura 12 – Reconstrução do automóvel - Primeiro teste **Figura 13 – Reconstrução do automóvel - Segundo teste**



Fonte: Autoria própria



Fonte: Autoria própria

Figura 14 – Reconstrução do vaso de flor - Primeiro teste com Fotografias **Figura 15 – Reconstrução do vaso de flor - Segundo teste com Fotografias**



Fonte: Autoria própria



Fonte: Autoria própria

Tabela 2 – Comparação das reconstruções: Automóvel e Vaso de flor

Atributo	Automóvel		Vaso de Flor	
	Teste um	Teste dois	Teste um	Teste dois
Tamanho do arquivo (MB)	1370	1370	58.5	58.6
Tempo de processamento	1h:40m:40s	1h:42m:23s	9m:11s	8m:08s
Vértices (v)	5570683	5576165	258814	259145
Vértices Normais (vn)	5570683	5576165	258814	259145
Vértices Texturizados (vt)	5697161	5700263	266599	268571
Faces/Triângulos (f)	11132222	11143376	515397	516104

Fonte: Autoria própria.



CONCLUSÃO

Com a pesquisa é possível observar que a técnica *Record & Playback* pode ser empregada na automatização de teste de sistemas para reconstrução 3D, mesmo para *softwares* comerciais, normalmente sem acesso ao código fonte, criando casos de teste que podem ser utilizados quando o sistema for modificado ou uma nova versão for lançada.

Entretanto, a forma de comparação não é invariante a translação, rotação e escala do modelo 3D. Adicionalmente, o custo computacional é alto devido ao tamanho dos arquivos que armazenam os modelos 3D; no entanto, com a divisão em arquivos com cada elemento e a ordenação é possível realizar o processamento em tempo viável, abrindo a possibilidade de executar em paralelo. A reconstrução não foi determinística em alguns casos e novas investigações são necessárias. O fundo escuro de determinadas entradas não prejudicou as reconstruções e as entradas de vídeo possibilitaram reconstruções com maior qualidade em alguns *softwares*.

Com base nos resultados da pesquisa, é possível averiguar que o número de vértices, vértices normais, vértices texturizados ou faces, não está relacionado diretamente com a qualidade gráfica de uma reconstrução. Foram observadas reconstruções que geraram arquivos grandes e foram custosas computacionalmente; no entanto, não obtiveram as melhores qualidades gráficas.

Agradecimento

À Universidade Tecnológica Federal do Paraná, pela bolsa (Edital PROPPG nº 05/2022).

Conflito de interesse

Não há conflito de interesse.

REFERÊNCIAS

- 3DFLOW. **3DF Zephyr**. [S.l.: s.n.], 2023. <https://www.3dflow.net/>. Acesso em: 20 set. 2023.
- AGISOFT LLC. **Agisoft Metashape**. [S.l.: s.n.], 2023. <https://www.agisoft.com/>. Acesso em: 20 set. 2023.
- AMMANN, Paul; OFFUTT, Jeff. **Introduction to Software Testing**. Cambridge: Cambridge University Press, 2016. ISBN 9781107172012. DOI: DOI:10.1017/9781316771273.
- AUTODESK. **Autodesk ReCap**. [S.l.: s.n.], 2023. <https://www.autodesk.com/>. Acesso em: 20 set. 2023.
- DELAMARO, M. E.; MALDONADO, J. C.; JINO, M. **Introdução ao teste de software (2 ed.)** Rio de Janeiro: Elsevier, 2016.
- ZACH, Christopher. **South building at UNC Chapel Hill**. [S.l.: s.n.], 2023. <https://onedrive.live.com/?cid=c58a258d760e1b58&id=C58A258D760E1B58%2146879&ithint=folder&authkey=%21AAQumsDDwZBIW3w>. Acesso em: 12 set. 2023.