**RESUMO**

O presente artigo relata o design e implementação de um sistema para geração semiautomatizada de modelos CAD 3D utilizando-se parâmetros de coordenadas geométricas relativas coletadas por sistema sensorial. A metodologia de desenvolvimento possui três etapas principais, geração automatizada de modelos CAD 3D por meio de utilização de Interface de Programação de Aplicações em Ruby do software de modelagem SketchUp, aquisição de dados de coordenadas geométricas relativas por meio da utilização de sistema sensorial inteligente embutido em plataforma mobile e integração das etapas anteriores por meio de utilização da classe WebDialog da API em Ruby do SketchUp para comunicar os parâmetros de coordenadas geométricas relativas obtidos aos scripts Ruby suportados pelo SketchUp. O sistema desenvolvido tem sido testado por meio de entrada de dados randômicos para geração de modelos CAD coerentes e comparação entre os valores de determinadas propriedades geométricas de objetos 3D reais e os processados com o auxílio de sistema de sensores. Este paper é concluído apresentando os atuais resultados da prototipagem do sistema proposto e fazendo-se uma análise com o intuito de verificar as suas vantagens, também são apresentadas novas direções de investigação neste campo.

**INTRODUÇÃO**

Noção geral do tema que será apresentado

Criar expectativa positiva no leitor

Despertar interesse

Delimitação do assunto

Objetivos do assunto

Finalidade

Ponto de vista sob o qual o assunto será tratado

Elementos necessários para situar o tema do artigo

Modelos 3D são utilizados em muitas aplicações, na indústria de games e filmes, setor médico, simulações de treinamento, realidade aumentada, arqueologia, inspeção industrial e muitas outras. Modelos 3D podem ser gerados por meio de processo de natureza artística ou técnica utilizando-se algum *software* de modelagem. No entanto, aplicações como inspeção industrial e projeto de plantas arquitetônicas exigem uma representação realística de objetos complexos [malik2009simulation].

Em vários tipos de indústria o desenho de modelos 3D de forma descritiva e precisa se torna necessária por vários motivadores, por exemplo, para legalizar determinada propriedade ou pedir financiamento para determinado projeto é necessária a modelagem de planta baixa do terreno. Na indústria mecânica e de manufatura se utiliza a modelagem 3D para reprodução de modelos e escalabilidade da produção. No entanto, o processo de modelagem de um objeto 3D pode se tornar dispendiosa em termos de dinheiro, tempo e mão de obra.

Analisando-se tal problema, este trabalho visa reduzir etapas do processo da modelagem de objetos 3D propondo um sistema para geração semiautomatizada de Projetos Auxiliados Por Computador, industrialmente conhecidos como *Computer-Aided Design* (CAD), utilizando-se parâmetros de coordenadas geométricas relativas coletadas por meio dos sensores de natureza de *hardware* e/ou *software* embutidos em dispositivo móvel como giroscópio, acelerômetro e sensores de proximidade.

O processo de digitalização 3D apresentado neste trabalho exige a seleção de diferentes pontos de visualização selecionados por um operador humano especializado. A qualidade do resultado final depende da quantidade e seleção de tais pontos. Portanto, a eficiência do resultado final depende também da expertise do operador.

A automação do processo de digitalização 3D pode melhorar a eficiência e qualidade do resultado final reduzindo custos de pós-processamento e exigindo menos das habilidades do operador. Atualmente não há um sistema de digitalização 3D completamente automatizado comercialmente, e há uma forte demanda por este na indústria (malik2009simulation).

O processo de modelagem 3D nesse projeto pode ser dividido em três etapas gerais: a primeira se refere ao processo de aquisição de dados de coordenadas geométricas relativas utilizando-se dispositivo móvel com sistema operacional Android; a segunda é relativa ao pós-processamento, onde a modelagem é feita por meio de *scripting* utilizando-se a API em Ruby do SketchUp; e última etapa envolve a integração das duas fases anteriores por meio da utilização da classe WebDialog da API em Ruby do SketchUp.

Interessa-se nesse trabalho em objetos cujas geometrias são facilmente especificadas como uma combinação de formas básicas. Por exemplo, muitas partes mecânicas podem ser desconstruídas como um conjunto de entidades geométricas simples (paralelepípedos, cilindros, esferas etc.).

Trabalhou-se em um algoritmo capaz de conectar os pontos coletados, através de relações espaciais aferidas.

**Formulação do Problema**

Segundo, a automação do processo de digitalização 3D pode melhorar a eficiência e qualidade do resultado final reduzindo custos de pós-processamento e exigindo menos das habilidades do operador. Atualmente não há um sistema de digitalização 3D completamente automatizado comercialmente, e há uma forte demanda por este na indústria.

A disponibilidade de dispositivos para aquisição de dados geométricos acurados, como um *scanner* 3D, tornou a aquisição de coordenadas espaciais de um amplo conjunto de pontos da superfície de um objeto 3D relativamente simples [[BERNADINI1999](#LyXCite-BERNADINI1999)], no entanto, nem sempre pode-se contar com a disponibilidade de um *scanner* 3D. Desse modo, verificam-se brechas para exploração de métodos que possam automatizar o processo de modelagem 3D utilizando-se ferramentas mais acessíveis e populares, não só do ponto de vista tecnológico, mas também econômico e cognitivo.

Hoje o processo para representação de objetos 3D é feito majoritariamente de modo manual, primeiro verifica-se as medidas dos elementos utilizando-se ferramentas como trena, régua, paquímetro etc., e em seguida é feito o desenho deste de forma manual ou utilizando-se algum software CAD. Tal processo, dependendo da complexidade do objeto a ser representado, pode se tornar entediante, trabalhoso, dispendioso e até mesmo repetitivo. Percebe-se assim, que o tempo utilizado com medições e desenhos manuais poderia ser utilizado para outras atividades de maior valor agregado.

**Justificativa**

A escassez de metodologias e ferramentas acessíveis do ponto de vista econômico e prático, que possibilitem reduzir passos no processo de modelagem de objetos 3D é o principal motivador deste trabalho. A possibilidade de poder gerar modelos 3D de forma automatizada ou semiautomatizada utilizando-se dispositivos populares como smartphones, hoje dotados de variados sensores, permite a construção de várias aplicações por parte de profissionais da área de tecnologia e até mesmo aficionados.

A velocidade dos avanços tecnológicos computacionais está continuamente abrindo caminho para novas ferramentas de design, aumentando a perspectiva de reutilização de projetos e automação. Esse fator pode ser usado para abolir tarefas rotineiras, eliminando possibilidade de erros humanos e abrindo espaço para desenvolvimento de produtos mais personalizados [[TARKIAN2009](#LyXCite-TARKIAN2009)].

Tal trabalho contribuiria para redução de atividades repetitivas, onerosas, além de possibilitar a modularização de projetos CAD, permitindo maior colaboração entre um time através do compartilhamento de modelos para utilização em outros projetos com diferentes medidas. Poder-se-ia até mesmo estender sua utilidade para a área de Manufatura Auxiliada por Computador (CAM) e prototipagem de sólidos em 3D, como impressão 3D.

O sistema proposto também abriria espaço para o desenvolvimento de novos produtos, beneficiando-se da produção modular e redução de trabalho com manutenção.

**Motivação**

Na sociedade moderna são notáveis os avanços obtidos nas mais diversas áreas do conhecimento por meio do auxílio de computadores, tecnologia web e tecnologia móvel (smartphones, wearables, etc.). As infinitas possibilidades de criação proporcionadas pelas atuais plataformas *mobile*, dotadas de variadas funcionalidades de hardware e software, abrem espaço para a exploração de diversos temas. Tais ferramentas aliadas à área de Controle e Automação possibilitam o foco em produtividade, racionalização de recursos (tempo, dinheiro, mão de obra etc.), melhoria da qualidade de produtos entre outros.

A possibilidade de explorar todos esses temas estudados durante o curso de bacharelado em Engenharia de Controle e Automação na Universidade do Amazonas em conjunto com experiência profissional e cursos relacionados oferece uma base, no sentido de atribuir caráter prático ao que se estuda em sala de aula.

O estudo dos sistemas produtivos e das tecnologias a eles associadas, como é o caso da automação industrial, está na frente dos assuntos que devem ser detalhadamente estudados dos diversos ramos relevantes da engenharia, de forma a obter o crescimento sustentável da sociedade, capaz de garantir o aumento da qualidade de vida sem desperdiçar em logo prazo os recursos disponíveis. Assim, a automação industrial evolui para uma disciplina multidisciplinar integrativa de várias áreas do saber associadas com o objetivo de obter sistemas produtivos eficientes, ágeis e fiáveis, dando uma perspectiva mais ampla ao conceito de automação industrial [[ROSARIO2005](#LyXCite-ROSARIO2005)].

A necessidade de evolução do processo produtivo tendente à sua automação global requer a formação de engenheiros com conhecimento não apenas de tecnologias que constituem a base da automação industrial, mas também das matérias ligadas aos sistemas computacionais, às teorias da decisão e da organização e aos sistemas distribuídos, o que lhes permite desempenhar um papel relevante no desenvolvimento e na manutenção dos sistemas da área[[ROSARIO2005](#LyXCite-ROSARIO2005)].

Poder aprender mais sobre a indústria CAD, softwares de modelagem 3D modernos, o caráter computacional envolvido na produção desses softwares, a sua integração com o ambiente Web, utilização programação orientada à objetos para o desenvolvimento de novas funcionalidades ou utilização das já existentes abre um leque de possibilidades e oportunidades. Além disso, o fato de poder explorar a utilidade de plataformas mobile neste trabalho garante um caráter inovador a propostas já existentes para o presente tema.

**METODOLOGIA**

O presente trabalho possui natureza de pesquisa aplicada, cujo objetivo é a realização de uma análise exploratória e explicativa sobre o material bibliográfico coletado. Os procedimentos técnicos a serem utilizados são os de pesquisa bibliográfica e experimental. O método de abordagem é o hipotético-dedutivo e a elaboração segue o método de procedimento monográfico. A coleta de dados é feita por meio de documentação direta e indireta, e a análise e interpretação dos dados são feitas de modo global.

O processo de pesquisa foi iniciado fazendo-se uma revisão bibliográfica de assuntos relacionados ao tema do presente trabalho. Foi realizada a leitura de artigos, livros, websites relacionados às ferramentas utilizadas e cursos online em plataforma de Curso Online Aberto e Massivo, popularmente conhecidos em língua inglesa como MOOC (Massive Open Online Course).

Durante a revisão bibliográfica de artigos, exploraram-se temas como:

* Reconstrução automática de modelos CAD 3D a partir de scanners digitais;
* Reconstrução de um objeto 3D a partir de um esboço desenhado manualmente;
* Geração de modelos de construção 3D a partir de plantas arquitetônicas;
* Reconhecimento de objeto tridimensional a partir de uma única imagem de duas dimensões;
* Reconhecimento automático de objetos de modelo CAD 3D em uma nuvem densa pontos utilisando laser;
* Construção automática de modelos 3D de ambientes inabitados utilizando scanners a laser e sensores RFID;
* Microambiente de sensoriamento para smartphones;
* Framework de sensoriamento em smartphones.

A pesquisa por meio de artigos forneceu uma visão mais madura sobre as atuais descobertas e avanços na área de automação na indústria CAD e também proporcionou o confronto com problemas que ainda precisam de soluções ou melhorias.

A revisão bibliográfica por meio de livros explorou temas como:

* Arquitetura de computadores;
* Desenvolvimento de software para sistema operacional Android;
* Computação gráfica;
* Programação orientada a objetos;
* Tecnologia web;
* Projetos auxiliados por computador;
* Sistemas sensoriais;
* Linguagem de programação Ruby.

Após realização da revisão bibliográfica inicial, foi realizada uma pesquisa por softwares de geração CAD 3D, onde ao final do processo, escolheu-se o *SketchUp*®. A automação de geração de modelos CAD 3D pode ser feita por meio de tal software utilizando a sua API em Ruby.

De modo a obter conhecimento para utilização da API em Ruby do SketchUp, participou-se do curso *Ruby on Rails: An Introduction* por meio da plataforma de MOOC Coursera, de forma a obter noções básicas de Ruby e programação orientada a objetos

Também foi realizada exploração pelo software SketchUp de modo a entender o seu funcionamento. Utilizaram-se tutoriais online e vídeo aulas existentes no site oficial do software.

Em seguida, foi feito um estudo das funcionalidades oferecidas pela API em Ruby do SketchUp. Pode-se ver algumas de suas classes e métodos, assim como exemplos de aplicações que acionam funcionalidades do SketchUp. O aprendizado de tais ajudou consideravelmente na fase de implementação do algoritmo para o projeto.

Sendo assim, pode-se criar scripts próprios que possuem objetivo de automatizar modelagem 3D. Nessa fase, foram feitos testes com coordenadas falsas, por meio da entrada de dados em caixas de diálogo.

Tendo os objetivos acima alcançados na área da aplicação desktop, partiu-se para o trabalho na área de subsistema móvel. Para este fim, o sistema operacional Android foi o escolhido por possuir uma considerável fatia de mercado.

De modo a obter produtividade no desenvolvimento da aplicação Android, escolheu-se utilizar a linguagem de programação Processing, que possui ambiente de desenvolvimento próprio, e a biblioteca Ketai, desenvolvida exclusivamente para manipulação de sensores e câmera de dispositivo existentes na maioria dos dispositivos móveis com sistema operacional Android.

Em seguida, testes foram realizados para verificação da funcionalidade dos sensores giroscópio, acelerômetro e a câmera do celular, podendo-se assim inicializar trabalho em algoritmo para captação de coordenadas geométricas. Os testes dessa etapa consistem na simples comparação entre as medidas obtidas por meio de aplicação Android desenvolvida e as medidas reais/analógicas.

Com a coleção de coordenadas obtidas por sistema mobile, foi feita então, a integração destes em software de modelagem 3D SketchUp utilizando-se a classe WebDialog do mesmo.

Tendo o modelo gerado após o processo de integração anteriormente descrito, é feita uma análise sobre sua coerência e propostas de melhoria são elaboradas.

**ORGANIZAÇÃO DO TRABALHO**

O presente trabalho se encontra organizado em seis capítulos.

O capítulo 1 consiste na introdução, onde procura-se situar o leitor no contexto da pesquisa, apresentando problemática, justificativa, motivação, objetivos, bases teóricas técnicas, visão geral da metodologia e estrutura do trabalho. Procura-se detalhar o alcance de investigação e aspectos de análise, servindo como complemento de delimitação do problema.

O capítulo 2 aborda o referencial teórico de forma detalhada, proporcionando embasamento teórico obtido para produção desta pesquisa, e fazendo-se um apanhado geral do que se sabe sobre o assunto investigado. Também apresenta trabalhos com temáticas relacionadas, tendo em vista o estágio atual da ciência, suas divergências e contribuições proporcionadas para o problema abordado.

O capítulo 3 aborda uma visão geral do sistema proposto, detalhando como ele é dividido, que tecnologias de software e hardware são utilizadas em cada camada do sistema assim como suas respectivas funcionalidades e papéis de forma sistêmica.

O capítulo 4 aborda a metodologia e materiais utilizados para a criação do sistema proposto, discutindo o arranjo experimental e sua precisão. Para equipamentos comerciais apresentam-se seus fabricantes e modelos. Também se discute como os dados foram analisados e incertezas foram consideradas.

O capítulo 5 aborda os resultados obtidos e discute sobre as descobertas do presente estudo, relatando sua acurácia e significância, no entanto, ainda não há interpretação de dados obtidos.

O capitulo 6 apresenta as conclusões sobre a presente pesquisa, retornando ao problema inicial lançado na introdução e revê suas principais contribuições, avaliando os pontos fortes e positivos através da reunião sintética das principais ideias desenvolvidas ou conclusões parciais obtidas. Os resultados obtidos são interpretados e discutidos de forma independente. Também se apontam problemas decorrentes do tema investigado que poderão servir para futuras pesquisas.

**Visão Geral do Sistema**

A Figura 1 mostra que a arquitetura do sistema descrito neste trabalho pode ser dividida em três camadas. A camada de hardware basicamente diz respeito aos sensores que ajudarão a capturar as coordenadas relativas a um ponto absoluto e a unidade de processamento central, também conhecida como CPU (*Central Processing Unit*), ou simplesmente processador.

A Unidade Central de Processamento é a parte que interpreta (ou executa) instruções armazenadas na memória principal. No seu núcleo se encontra um dispositivo de armazenagem de tamanho de palavra (ou registrador) chamado *program counter* (PC), ou em língua portuguesa, contador de programa. A qualquer momento, o PC aponta para alguma instrução linguagem de máquina na memória principal. (OHALLARON).

Os sensores de movimento, como acelerômetro giroscópio, ajudam a monitorar a movimentação do dispositivo. Eles também são importantes nesse projeto pois ajudam a diminuir o erro de medição de um ponto de visualização para outro, pois o operador humano possui movimentos imprecisos e sujeitos a vibrações, rotações e movimentos de variada intensidade.

O acelerômetro determina a orientação do dispositivo, assim como sua aceleração no espaço tridimensional, o qual será utilizado para detectar balanços (SAUTER, 2013).

O giroscópio mede o movimento do dispositivo, retornando a taxa de rotação em torno de cada um dos eixos do dispositivo. Se disponível, este sensor é frequentemente utilizado em jogos que dependem em respostas imediatas e precisas aos movimentos do dispositivo (SAUTER, 2013).

A câmera torna-se útil aqui pois, existem os sensores detectores de posição (SDP) produzidos para sensoriamento preciso de posição e auto-focagem em câmeras fotográficas. Tais sistemas utilizam um diodo emissor de luz (LED) e um sensor fotodetector. Desse modo, a posição de um objeto é determinada pela aplicação do princípio de medição triangular (Fraden, 2010).

A camada de aplicação móvel é responsável pelo gerenciamento e manipulação dos sensores, para tal será utilizada a linguagem de programação Processing e a biblioteca Ketai. Para armazenamento de coordenadas processadas será utilizado o banco de dados padrão do sistema operacional Android, o SQLite. Assim, será possível enviar tais dados via Web para que estes possam ser utilizados pela classe WebDialog do SketchUp.

E finalmente, a camada de aplicação Desktop utilizará a Web API em Ruby do SketchUp para automação de geração de modelos 3D por meio de programação. A integração de dados processados pela camada de aplicação móvel e o trecho de código criado para automação de modelagem 3D é feita utilizando-se a classe WebDialog do SketchUp.

**TRABALHOS RELACIONADOS**

**Simulação de Aquisição 3D Automática e Pós-Processamento**

Tal artigo, intitulado originalmente como *A Simulation of Automatic 3D Acquisition and Post-processing Pipeline (MALIK2009)*, apresenta uma simulação de aquisição 3D automática e etapa de pós-processamento. A metodologia proposta é aplicada a um *scanner* baseado em triangulação a laser e braço robótico com seis graus de liberdade. Os pontos de visualização são computados resolvendo-se um problema de cobertura de conjuntos para reduzir o número de potenciais posições. A qualidade do plano de visualização é determinada por seu tamanho e porcentagem de área da superfície do objeto que se cobre.

Nesse artigo vê-se que o processo de digitalização 3D pode ser dividido em dois passos gerais chamados *Aquisição* e *Pós-processamento*. O sistema de aquisição geralmente possui um digitalizador 3D (por exemplo, *scanner* com varredura laser, câmera do tipo *Time of Flight* ou *Stereo Vision*) montados em um sistema de posicionamento (exemplo: braço articulado). O passo de pós-processamento inclui registro de dados, preenchimento de buracos, mapeamento do ambiente, fotometria e limpeza de dados.

O processo de digitalização 3D requer a seleção de diferentes pontos de visualização, usualmente selecionados por um operador humano especializado. A qualidade final dos resultados depende da seleção dos pontos. A quantidade de dados digitalizados depende do número de pontos de visualização.

A aquisição 3D automática requer planejamento de caminho ou vista. O planejamento de vista é um problema bem estudado e tem muitas soluções propostas e pode ser classificado em método *baseado em modelo* e *não baseado em modelo*. No planejamento de vista baseado em modelo, um modelo completo ou parcial (CAD ou *Mesh*) do objeto se encontra disponível. O passo de planejamento de vista acontece usualmente *offline* e de modo interativo. Em um planejamento de vista não baseado em modelo, não existe informação disponível sobre o objeto, o planejamento de vista é feito durante o processo de aquisição. O trabalho apresentado nesse artigo se concentra em um planejamento de vista baseado em modelo, ao contrário do que se propões neste trabalho.

O processo de simulação apresentado em tal artigo possui as seguintes etapas:

Carregamento do modelo de referência. O modelo de referência é uma malha triangular de alta qualidade do objeto a ser digitalizado.

Uma cópia decimada de modelo de referência é feita para reduzir o número de faces e a complexidade de passo de planejamento da vista. O modelo de referência usualmente tem um amplo número de vértices e faces (triângulos). O espaço e complexidade do tempo do algoritmo de planejamento da vista depende do número de faces utilizadas no modelo.

Um scanner de triangulação a laser é modelado na simulação. O scanner pode ser posicionado em qualquer posição e orientação do espaço de seis graus de liberdade. O tronco do scanner é modelado como um cone esférico, com ângulos de visão de campo horizontal e vertical, como mostrado na figura x. O centro da esfera se encontra na origem do sistema de coordenadas do scanner. Os dois cilindros representam a fonte do scanner e sensor. As linhas representam os feixes selecionados com o tronco do scanner.

A posição e orientação do scanner 3D é referenciado como ponto de visualização ou pose. A amostra dos possíveis pontos de visualização é chamada de espaço de pontos de visualização, tal espaço é gerado utilizando-se dois métodos diferentes: esfera de visualização e zona de varredura ótima. O método de Esfera de visualização é baseado na amostra de esfera de visualização posicionada no centro de massa do objeto com o raio apropriado que pode encapsular o objeto por inteiro. A amostragem é feita utilizando resolução de ângulo de rotação sobre os eixos x e y. A figura [Esferas de visualização com diferentes resoluções.](#Esferas_de_visualiza____o_com_diferente) mostra três esferas de visualização com diferentes resoluções (45°, 30° e 15°).

O método de zona de varredura ótima usa modelos de objetos decimados para gerar espaço de pontos de visualização. Ele é baseado em um algoritmo de zona de varredura ótima. Para cada face no modelo decimado, a face normal é computada. Então a posição do scanner é selecionada em uma distância fixa do centro da face em direção à face normal. A orientação do scanner é selecionada na direção oposta à face normal. O número de pontos de visualização gerados utilizando esse método é igual ao número de faces utilizadas no modelo.

Restrições também são adicionadas no sistema de posicionamento, neste trabalho foi utilizado o robô industrial Kuka KR16 como sistema de posicionamento na simulação, que possui seis eixos ou seis graus de liberdade. O robô é modelado matematicamente com a Tollbox de Robótica do MATLAB. A cinemática de avanço do robô é modelada usando o algoritmo de Denavit-Hartenberg. Uma vez que o robô é matematicamente modelado, um ponto de visualização pode ser testado para ver se a posição desejada pode ser obtida. Isso é feito utilizando-se problema de Cinemática Inversa, um processo para determinar os parâmetros de juntas do robô de modo a atingir a posição desejada. Se a posição desejada não pode ser atingida, não há solução para a Cinemática Inversa. A Cinemática Inversa é resolvida para cada ponto de visualização, e se não há solução, o ponto de visualização correspondente é rejeitado, do contrário, ele é adicionado ao espaço de pontos de visualização.

O plano de visualização do algoritmo seleciona o número mínimo de pontos de visualizações a partir do espaço de visualização que cobre a superfície máxima do objeto.

Após isso, a funcionalidade de um scanner baseado em triangulação linear a laser é simulado. Primeiramente, o tronco do cone é gerado para cada posição. O tronco do cone é representado por uma amostra de raios com origem no seu centro, que cobre o campo de visão inteiro. O número de feixes é dependente das resoluções horizontais e verticais e podem ser achadas através da seguinte fórmula:

FÓRMULA

Os dados obtidos a partir do escaneamento é um conjunto de nuvem de pontos. Os pontos da nuvem de todas as vistas são fundidos em uma única configuração. A triangulação de superfície converte o conjunto de nuvem de pontos em uma malha triangular.

**SEGUNDO ARTIGO**

Tal artigo, intitulado originalmente como *Correlation-Based Reconstruction of a 3D Object From a Single Freehand Sketch*, propõe uma nova forma de reconstruir um objeto tridimensional a partir de um desenho manual 2D. Ele se baseia no fato de que um esboço é essencialmente uma projeção com ruídos de um objeto 3D em um plano 2D arbitrário. Neste artigo chama-se de *reconstrução* a projeção inversa da geometria do esboço a partir de duas dimensões de volta em três dimensões.

Enquanto humanos podem fazer essa projeção reversa de forma fácil e intuitiva, esse processo é matematicamente indeterminado e complexo de emular computacionalmente e é exatamente esta habilidade de entender e gerar esboços que se deseja emular no trabalho relatado por tal artigo. Este artigo propõe que a habilidade dos humanos de perceber um objeto 3D previamente não visto a partir de um esboço é baseado na simples relação geométrica 2D-3D aprendida a partir de experiência visual. Por outro lado, mostra-se como o mesmo dado de correlação pode ser usado para medir o entendimento de projeções de objetos 3D gerados sinteticamente.

Usando tais princípios demonstra-se pela primeira vez uma conversão completamente automática de um esboço feito à mão em um objeto sólido físico. Esses resultados possuem implicações para comunicação homem-máquina bidirecional de conceitos de gráficos 3D, e também pode lançar luz sobre o sistema visual humano.

**1 REFERENCIAL TEÓRICO**

Segundo \citeonline{BERNADINI1999}, o planejamento do design, engenharia e manufatura de novos produtos está cada vez mais sendo conduzido por simulação computacional. Uma necessidade comum nesse processo é a incorporação de objetos existentes nesse ambiente de prototipagem eletrônica, para reutilizá-los como parte de um novo produto, ou para adaptação e melhoria de seu design para atender a determinados requisitos.

A disponibilidade de dispositivos para aquisição de dados geométricos acurados, como um scanner 3D, tornou a aquisição de coordenadas espaciais de um amplo conjunto de pontos da superfície de um objeto 3D relativamente simples. Aplicações que se beneficiariam de um método eficiente e confiável para construir modelos geométricos a partir desta coleção de medidas incluem:

Engenharia reversa: A partir de um objeto existente, reconstrói-se seu modelo computacional, e então seu design é analisado e modificado. Engenharia reversa tem aplicações relevantes na indústria de manufatura.

Análise de Forma: Analisa-se a deformação de uma peça mecânica após uma colisão.

Criação de mundos virtuais 3D: Construir modelos de caracteres, atores e espaços rapidamente a partir de peças analógicas, réplicas ou moldes.

Fax 3D: Faz-se a varredura de um objeto e transmite-se o dado digitalizado em uma linha de telefone. A estação receptora reconstruíra o modelo e fabricará uma cópia utilizando alguma técnica de prototipagem rápida como estéreo-litografia.

Modelagem adaptada: Manufatura de aparelhos customizados, capacetes ou próteses a partir de uma varredura do corpo.

A construção do sistema abordado neste trabalho/artigo envolve basicamente três fases. A primeira consiste na geração automatizada de modelos 3D em determinada plataforma CAD por meio de scripting, a segunda fase é referente à coleta de coordenadas geométricas relativas por meio de sistema sensorial e a última envolve a integração entre as duas fases anteriores.

**1.1 GERAÇÃO AUTOMATIZADA DE MODELOS 3D**

O processo de automação de modelagem em várias plataformas CAD pode acontecer por meio de scripting, ou seja, através da criação de programas feitos em determinadas linguagens para implementação de comandos existentes em tal plataforma. As seções a seguir descrevem as tecnologias utilizadas para automação de geração de modelos 3D neste trabalho, assim como as motivações para suas escolhas.

**1.1.1 SketchUp**

SketchUp é um *software* utilizado para criar modelos 3D disponível nas versões Make e Pro. Vários fatores influenciaram na escolha desta ferramenta no projeto, como facilidade de uso e aprendizado, disponibilização de versão gratuita, acessibilidade, rapidez, abundância de bibliotecas de modelos 3D disponíveis online, qualidade de documentação, existência de fóruns informativos na web, leitura e escrita a partir de outros formatos de arquivo 3D, entre outros. No entanto, os fatores mais relevantes para esta escolha é a existência da API em Ruby para codificação e automação de modelagem e a existência da classe WebDialog que permite integração com dados da Web.

**1.1.1 Ruby**

Ruby é uma linguagem de programação open source e dinâmica com foco em simplicidade e produtividade. Ela possui uma sintaxe elegante que a torna de simples leitura e escrita. Ruby é uma linguagem de scripting interpretada, projetada para fácil programação orientada a objetos que combina os melhores elementos de linguagens populares como Perl, Smaltalk, Eiffel, Ada e Lisp. SketchUp pode entender tal linguagem através de sua API em Ruby.

**1.1.1.1 Classes**

**1.1.1.2 Módulos**

**1.1.2 Script**

Um script é um programa de tamanho pequeno que se difere de um programa compilado por ser um arquivo interpretado no tempo de execução ao invés de ser compilado em código binário.

**1.1.3 API em Ruby do SketchUp**

Consiste em um conjunto de classes Ruby específicas do SketchUp que podem ser utilizadas em seus scripts para estender a funcionalidade do SketchUp. SketchUp possui três importantes módulos que fornecem formas de manipular SketchUp com Ruby

**1.1.3.1 Módulo UI**

**1.1.3.2 Módulo Sketchup**

**1.1.3.3 Módulo Geom**

**1.1.3.1 O que se pode fazer com a API em Ruby do SketchUp?**

Codificação de novas ferramentas de modelagem;

Construção de ferramentas avançadas que aumentam a velocidade de modelagem e permitam gerar objetos complexos de forma simples e rápida;

Anexação de texto ou atributos numéricos em qualquer parte de um modelo SketchUp e geração de relatórios;

Automatização de ações repetidas frequentemente;

Criação e amostragem de novas janelas que permitam a realização de ações como mostrar próprias vistas renderizadas customizadas, navegação por dados customizados (exemplo: conta de materiais);

Fazer ligação com bibliotecas customizadas (exemplo: drivers de dispositivo);

Leitura e escrita para ou a partir de outros formatos de arquivo 3D, etc.

Anexação e obtenção de meta-data de modelos;

Animar objetos;

Criar jogos;

**1.1.5 Console Ruby do SketchUp**

O console Ruby do SketchUp é um pequeno diálogo que pode ser mostrado utilizando o item do menu Window > Ruby Console. O Console fornece acesso direto ao interpretador de Ruby do SketchUp. Nele há uma caixa de entrada onde se pode digitar comandos Ruby e uma janela de saída onde se pode ver a saída de debugging.

**Local de armazenamento de scripts Ruby para que o SketchUp possa executá-los**

Ao criar um arquivo script de extensão .rb, pode-se armazenar este na pasta de Plugins do SketchUp. No sistema operacional Windows ela se localiza tipicamente em C:\Program Files\Google\Google SketchUp [n]\Plugins. Quando o SketchUp for inicializado, ele executará todos os scripts localizados na pasta Plugins.

**1.1.6 SketchUp e a Web**

SketchUp é uma aplicação desktop, portanto, não é possível executá-la na Web, no entanto com seu objeto WebDialog é possível integrar conteúdo da web.

**Qual é a melhor forma para realizar debug de scripts?**

A forma mais simples de realizar debug de scripts é incluir declarações de impressão de dados na tela ao longo do código para obtenção de valores intermediários usando o comando de Ruby “puts”. As saídas dessas declarações podem ser vistas no Console Ruby do SketchUp à medida que seu script é interpretado.

**1.2 AQUISIÇÃO DE DADOS PARAMÉTRICOS DE COORDENADAS GEOMÉTRICAS RELATIVAS**

A atual difusão das plataformas mobile, como Android e iOS, estão ampliando o conceito tradicional de dispositivos móveis para fornecer não apenas recursos computacionais, mas também capacidade de sensoriamento, tais como sensores embutidos, incluindo acelerômetros, giroscópios, GPS, microfones e câmeras. Estas novas funcionalidades transformam os dispositivos móveis em plataformas de sensoriamento poderosas, estando continuamente atentos ao mundo físico, comportamentos e movimentos do usuário [Cardone2013MobPhonSensFramew].

**1.2.1 Sensores Embutidos de Plataforma Android**

A maioria dos dispositivos que possuem sistema operacional Android possuem sensores embutidos capazes de medir movimento, orientação e condições do ambiente. Estes sensores são capazes de fornecer dados brutos com alta precisão e acurácia, e são úteis para monitorar o movimento ou posicionamento do dispositivo em três dimensões. Por exemplo, um determinado jogo desenvolvido para o sistema operacional Android pode acompanhar leituras feitas por um sensor de gravidade do dispositivo de modo a inferir gestos complexos dos usuários, como inclinação, vibração e rotação.

(<http://developer.android.com/guide/topics/sensors/sensors_overview.html>) 7:36PM 10/10/2015

Você pode acessar sensores disponíveis no dispositivo e adquirir dados brutos destes utilizando o framework de sensores do Android. Tal framework oferece uma séria de classes e interfaces que ajudam o desenvolvedor a executar uma variedade de tarefas envolvendo sensores, como

* Determinar que sensores se encontram disponíveis no dispositivo;
* Determinar uma capacidade individual do sensor, como sua variação máxima, fabricante, requisitos de potência e resolução;
* Registrar e deletar registros de ouvintes de eventos que monitoram mudanças do sensor.

Alguns dos sensores existentes em dispositivos que suportam plataforma Android possuem natureza de hardware e outros são de natureza de software. Os primeiros sensores citados são componentes físicos incorporados no telefone ou tablet. Eles derivam seus dados de medições diretas de propriedades específicas do ambiente. Sensores com natureza de software, também conhecidos como sensores sintéticos ou virtuais tentam reproduzir o comportamento de sensores de natureza de hardware. O sensor de aceleração linear e o sensor de gravidade são exemplos destes.

A plataforma Android suporta três amplas categorias de sensores.

**1.2.1.1 Sensores de Movimento**

Estes sensores medem forças de aceleração e rotacional ao longo dos três eixos. Esta categoria inclui acelerômetros, sensores de gravidade, giroscópio e sensores de vetor rotacional.

**1.2.1.2 Sensores de Ambiente**

Esses sensores medem vários parâmetros do ambiente, tais como temperatura, pressão, iluminação e umidade. Esta categoria inclui barômetros, termômetros e sensores fotométricos.

**1.2.1.3 Sensores de Posição**

Esses sensores medem a posição física de um dispositivo. Esta categoria inclui sensores de orientação e magnetômetros.

**1.2.2 Processing para Android**

É possível criar aplicativos Android avançados utilizando Processing, uma linguagem de programação de código aberto e de ambiente gratuito, projetada para aprender os fundamentos de programação. Ela possui mais de 130 bibliotecas expandindo o seu núcleo, e há possibilidade de estendê-la com classes e métodos de Java e Android. Ela é uma linguagem simples, mas poderosa. Processing vem com três módulos que permitem criar aplicações para diferentes dispositivos e sistemas operacionais. O modo Java permite criar aplicações para GNU/Linux, Mac OS X e Windows. O modo Android permite criar aplicativos Android para tablets e smartphones. E finalmente, o modo Javascript habilita o desenvolvedor a criar aplicações Web utilizando Processing.js, estas serão executadas em todos os navegadores web com habilitação HTML5 instaladas em celulares, tablets e computadores desktop. Processing possui seu próprio Ambiente de Desenvolvimento Processing (ADP).

1.2.2.1 Estrutura de um Sketch

Qualquer esboço de *Processing* que interagirá com os usuários ou faz uso de gráficos animados – deve incluir dois métodos:

* Uma instância do método setup(), que inicializa variáveis chaves e configurações que o sketch (esboço) utilizará, executado apenas uma vez quando o aplicativo inicializa.
* Uma instância do método draw(), que continuamente atualiza ou redesenha a tela para responder às entradas do usuário e eventos em tempo real.

Um típico esboço de Processing inicia definindo-se as variáveis globais que serão utilizadas, seguidas pelos métodos setup() e draw(). Setup() é chamado exatamente uma vez quando se começa um esboço para inicializar parâmetros chave. Por exemplo, pode-se configurar determinada janela com size(), orientação da tela com orientation(), ou carregar fontes customizadas e ativos de mídia. Em resumo, setup() é responsável por tomar conta de tudo que precisa ser feito uma vez para configurar um esboço.

Em contraste, o método draw(), é chamada repetidamente para atualizar a tela, 60 vezes por segundo por padrão. Pode-se ajustar tal taxa utilizando o método frameRate(). Caso os desenhos sejam complexos, ou exijam uma quantidade substancial de poder de processamento, Processing talvez não consiga manter a taxa de 60 quadros.

Os métodos setup() e draw() não aceitam parâmetros. Eles são métodos *void* e não retornam valores. Ambos são usados virtualmente em cada esboço de Processing

Processing armazena a informação de onde o mouse está localizado em determinado momento. Tal informação é armazenada relativa ao canto esquerdo superior da janela, em duas constantes: mouseX e mouseY. Ambas constantes retornam a localização do pixel atual do mouse em relação a origem da janela de display. Em Processing a coordenada [0, 0] se encontra no canto superior esquerdo da janela. [comprimento -1, altura -1] se localiza no canto inferior direito.

1.2.2.2 Biblioteca Ketai

Esta biblioteca para Processing foca particularmente na facilidade de desenvolvimento com funcionalidades de hardware móvel embarcadas em celulares e tablets Android. Ela foi projetada particularmente para oferecer acesso a sensores, câmeras e redes de dispositivos Android, e foi desenvolvida para executar unicamente em modo Android.

O termo “Ketai" é usado no Japão para descrever sua cultura de celular, habilitada por dispositivos móveis. Há mais de 130 bibliotecas disponíveis para Processing, no entanto em dispositivos Android, pode-se apenas utilizar bibliotecas escritas em Java.

Bibliotecas são definitivamente o aspecto de maior sucesso do projeto de código aberto Processing. Elas estendem o núcleo da Processing com classes escritas para contextos particulares, incluindo 3D, Animação, Visão Computacional, Compilações, Protocolos e Dados, Geometria, Interface Gráfica, Interface de Hardware, Matemática, Simulação, etc.

A biblioteca Ketai permite considerar os dispositivos móveis como um complemento aos microcontroladores, como Arduino.

Ketai inclui um número de classes que

1.3 Integração de dados om WebDialog

A classe WebDialog de Ruby permite criar e interagir com caixas de diálogo DHTML a partir de Ruby. Essa é a melhor forma de gerar Interface de Usuário embarcada e complexa no SketchUp, mas geralmente requer conhecimento de HTML e Javascript.

**METODOLOGIA**

Analisar Market share entre sistemas operacionais de plataforma móvel. Citar porque escolheu Android

Com o auxílio do acelerômetro e do giroscópio procura-se obter o ângulo α descrito na Figura 4. A distância “d" é obtida por meio de sensor óptico linear (PSD – Position Sensitive Device). O módulo de medição de posição incorpora um diodo emissor de luz (LED – Light Emitting Diode) e um PSD de fotodetecção. A posição do objeto é determinada aplicando-se o princípio de medição triangular. Tal princípio possui este nome porque os sensores, o laser emitido e a luz do lazer refletida formam um triângulo, como demonstrado na Figura 5.

O feixe de luz é projetado a partir do dispositivo móvel e é refletido a partir de uma superfície alvo para uma coleção de lentes. Estas lentes tipicamente se localizam adjacente ao emissor de luz. As lentes focam uma imagem do lugar numa câmera de vetor linear. A câmera visualiza o raio de medição. A posição da imagem flagrada nos pixels da c6amera são então processados para determinar a distância do alvo.

**Amostragem de Valores do Acelerômetro**

O acelerômetro é o sensor mais comumente encontrado em dispositivos móveis, ele é projetado para detectar aceleração do dispositivo em relação à gravidade, retornando coordenadas X, Y e Z do dispositivo, medidas em metros por segundos quadrados. O acelerômetro nãoé capaz de distinguir entre rotação e movimento, logo, movimentar o dispositivo para trás e para frente, e rotacionar-lo em torno de seu eixo pode produzir valores idênticos. Para diferenciar entre movimentação e rotação, um sensor adicional é necessário, o giroscópio.

**2 DESENVOLVIMENTO**

**2.1 Guia de Design e Arquitetura Lógica**

Desenvolver

**NOVAS DIREÇÕES DE INVESTIGAÇÃO**

Existem aspectos do problema da engenharia reversa que nós não investigamos nesse trabalho. Por exemplo, dados estão suscetíveis a ruídos.

Modelos de produtos CAD 3D podem seu usados como referência de modo a gerenciar processos de verificação e controle da qualidade, inspeção automática de formatos 3D, incluindo tolerâncias geométricas e dimensionais. A informação geométrica representada em um modelo CAD 3D pode ser utilizada de modo a planejar virtualmente, simular e conduzir inspeções no processo [Raffaeli2011].

**CONSLUSÃO**

Dar respostas às questões da pesquisa, correspondentes aos objetivos propostos e hipóteses levantadas.

Deve ser breve, podendo se necessário

Este trabalho se trata de um humilde início para um projeto ambicioso. Foi proposto por meio deste a semiautomatização de modelagem de um objeto 3D de propriedades simples por meio de coleta de pontos geométricos relativos. Entende-se que com a utilização de outras ferramentas e metodologias a automatização completa de modelagem 3D de objetos mais complexos pode se tornar possível. Por exemplo, em projetos futuros e/ou relacionados pode-se utilizar scanners 3D, métodos de processamento de imagem e visão computacional para que os dados do ambiente percebidos sejam mais acurados e descritivos.

A utilização de novas ferramentas disponíveis no mercado e de código aberto, como Ruby, Processing e Android neste projeto revela como cada vez mais é possível ousar em projetos customizados feitos por estudantes, engenheiros, técnicos e simpatizantes.