

POLYANA BEZERRA DA COSTA

**MENSURAÇÕES AUTOMÁTICAS ATRAVÉS
DE REALIDADE AUMENTADA EM OBRAS
CIVIS: MEDIÇÃO E CONSTRUÇÃO DE
AS-BUILT**

SÃO LUÍS - MA

2018

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).
Núcleo Integrado de Bibliotecas/UFMA

Bezerra da Costa, Polyana.

Mensurações Automáticas Através de Realidade Aumentada
em Obras Civas : Medição e Construção de As-Built /
Polyana Bezerra da Costa. - 2018.

49 f.

Orientador(a): Anselmo Cardoso de Paiva.

Monografia (Graduação) - Curso de Ciência da
Computação, Universidade Federal do Maranhão, São Luís -
MA, 2018.

1. As-Built. 2. Dispositivos Móveis. 3. Medição
automática de distâncias. 4. Plataforma Tango. 5.
Realidade Aumentada. I. Cardoso de Paiva, Anselmo. II.
Título.

POLYANA BEZERRA DA COSTA

**MENSURAÇÕES AUTOMÁTICAS ATRAVÉS DE
REALIDADE AUMENTADA EM OBRAS CIVIS:
MEDIÇÃO E CONSTRUÇÃO DE AS-BUILT**

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Anselmo Cardoso de Paiva

SÃO LUÍS - MA

2018

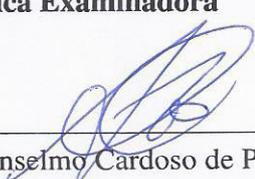
Polyana Bezerra da Costa

**Mensurações Automáticas Através de Realidade Aumentada em
Obras Civas: Medição e Construção de As-Built**

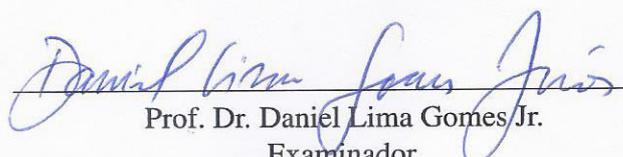
Monografia apresentada ao curso de Ciência da
Computação da Universidade Federal do Maranhão,
como parte dos requisitos necessários para obtenção
do grau de Bacharel em Ciência da Computação.

Data de Aprovação: 17/01/2018

Banca Examinadora



Prof. Dr. Anselmo Cardoso de Paiva
Orientador
Universidade Federal do Maranhão



Prof. Dr. Daniel Lima Gomes Jr.
Examinador

Instituto Federal de Educação, Ciência e Tecnologia do Maranhão



Profa. Dra. Simara Vieira da Rocha
Examinadora
Universidade Federal do Maranhão

São Luís - MA
2018

Aos meus pais.

Agradecimentos

A Deus, por me ajudar e guiar nos bons e maus momentos. Aos meus pais e meu irmão, pela confiança, afeto, e pelo suporte emocional e financeiro.

Ao meu namorado, Alexandre Pessoa, por ser um companheiro para todos os momentos, por me motivar, e por me inspirar com seu esforço e dedicação em tudo que faz.

Aos meus queridos amigos Anderson, Daniela, Nigel e Petterson. Anderson, obrigada por ser aquela pessoa prestativa que sempre me lembrava das cadeiras de férias, ou quando período de matrículas estava aberto, e mais recentemente, por me avisar das incontáveis regras da ABNT. Dani, obrigada pelo companheirismo e por ser minha dupla por todos esses anos seja na graduação, no Labmint, na escrita de artigos, e até nas viagens, muito obrigada também pelos almoços maravilhosos. Nigel, obrigada por me ajudar com a plataforma Tango e por ser aquela pessoa positiva que acredita que tudo vai dar certo. Petterson, muito obrigada por sempre tirar minhas dúvidas em relação ao Unity e pelo companheirismo.

Ao meu antigo chefe Tiago, obrigada pelos ensinamentos e por ter paciência e não desistir de mim mesmo eu sendo incapaz de chegar no horário pras nossas reuniões. Ao meu outro chefe Paulo, pelo ensinamentos passados, pela paciência, por me incentivar a aprimorar meus conhecimentos, e por ser uma inspiração por sua competência e esforço.

A todos os professores da graduação, pelo conhecimento passado e pela inspiração.

Por fim, ao Prof. Anselmo, por todo aprendizado e oportunidades que me proporcionou, pela sua paciência, pelos conselhos e por toda confiança depositada em mim.

"The people I've met are the wonders in my world. "
(Adele: Hometown Glory)

Resumo

Este trabalho combinou técnicas de Realidade Aumentada e Computação Gráfica para o desenvolvimento de uma ferramenta de automatização de mensurações de superfícies, assim como cálculos de área. A ferramenta proposta visa auxiliar profissionais das áreas de Arquitetura, Engenharia e Construção (AEC) e usuários comuns na captura de dados de *As-Built*. *As-Built*, ou "como construído", se refere ao conjunto de medidas tomadas em construção ou levantamento topográfico de um área ou elemento. Tais medidas podem ser utilizadas para inspeções, para comparar se a área construída corresponde à área planejada e para construção de registros de *As-Built*, um desenho técnico que irá representar a atual situação da área de interesse. A aplicação desenvolvida atua de forma semiautomática, de modo que o usuário define os pontos que deseja analisar na tela de seu *smartphone*, sem a necessidade de contato direto com o ambiente analisado, transformando os pontos delimitados pelo usuário em um polígono, e de forma automática apresenta todas as medidas e a área total do mesmo. Com o objetivo de minimizar erros que possam ser causados por falha humana, aumentar a precisão das mensurações e poupar tempo, a ferramenta proposta usa técnicas como Triangulação de Delaunay, da Computação Gráfica, e conceitos como marcadores naturais, da Realidade Aumentada, em conjunto com a plataforma Tango para extrair informações da superfície analisada e para a construção de uma aplicação móvel confiável e que funcione em tempo real. Uma série de testes foi realizada a fim de avaliar a confiabilidade das medidas obtidas com o método proposto, e os resultados foram comparáveis aos obtidos de forma manual, comprovando a viabilidade do uso da ferramenta desenvolvida.

Palavras-chaves: Medição automática de distâncias, Realidade Aumentada, Dispositivos Móveis, Plataforma Tango, As-Built.

Abstract

This work combined Augmented Reality and Computer Graphics' techniques for the development of a tool for automation of surface measurements and area calculations. The proposed tool aims to assist professionals in the areas of Architecture, Engineering and Construction (AEC) and common users in the capture of As-Built data. As-Built refers to the set of measurements taken under construction or topographic survey of an area or element. Those measurements can be used for inspections, to compare if the constructed area corresponds to the planned area and for construction of As-Built drawings, a technical drawing that will represent the current situation of the area of interest. The application presented in this project acts semi-automatically so that the user defines the points that he wants to analyze on the screen of his own smartphone, without the need of direct contact with the environment. The tool transforms the points delimited by the user into a polygon and automatically displays all measurements and the total area of the surface. This tool aims to minimize errors that can be caused by human failure, increase the precision of the measurements and save time. Techniques such as Delaunay's Triangulation and concepts like natural markers were used in conjunction with the Tango platform to extract information from an area and to build a reliable, real-time mobile application. It is expected that the results obtained by the application are comparable to those obtained manually. A series of tests were carried out in order to evaluate the reliability of the measurements obtained with the proposed method, and the results were comparable to those obtained manually, proving the feasibility of using the developed tool.

Keywords: Automatic measurements, Augmented Reality, Mobile Development, Google Tango, As-Built.

Lista de ilustrações

Figura 1 – <i>Continuum</i> de realidade-virtualidade.	19
Figura 2 – Câmera usada em Motion Tracking, e possíveis orientação e direção. . .	23
Figura 3 – Exemplo de tecnologias usadas na captura de dados de profundidade. .	23
Figura 4 – Exemplo de dados de profundidade em forma de nuvem de pontos obtido por um dispositivo equipado com a plataforma Tango.	25
Figura 5 – Tipos de triangulação.	26
Figura 6 – Condição do circuncírculo vazio.	26
Figura 7 – Etapas da Triangulação de Delaunay.	26
Figura 8 – Registro de <i>As-Built</i> de uma residência.	28
Figura 9 – Lenovo Phab 2 Pro.	29
Figura 10 – Etapas da Metodologia Proposta.	30
Figura 11 – Diagrama de atividades da principal funcionalidade da aplicação. . . .	31
Figura 12 – Tipos de planos e seus vetores normais.	33
Figura 13 – Informação virtual vista de diferentes perspectivas.	36
Figura 14 – Exemplos de funcionamento da aplicação.	37
Figura 15 – Mais exemplos de funcionamento da aplicação.	38
Figura 16 – Obtenção de área com a ferramenta desenvolvida.	42

Lista de tabelas

Tabela 1 – Interpretação dos valores do ICC	39
Tabela 2 – Medidas de até 1 metro.	40
Tabela 3 – Medidas de até 2 metros.	40
Tabela 4 – Medidas de até 3 metros.	41
Tabela 5 – ICC de cada grupo de medidas.	42
Tabela 6 – MSE de cada grupo de medidas.	43
Tabela 7 – MSE do terceiro grupo de medidas sem o pior caso.	43

Lista de abreviaturas e siglas

RA	Realidade Aumentada
AEC	Arquitetura, Engenharia e Construção
VE	Virtual Environments
API	Application Program Interface
VIO	Visual-Inertial Odometry
ICC	Coefficiente de correlação intra-classe
MSE	Mean Square Error
BIM	Building Information Modeling

Sumário

1	INTRODUÇÃO	13
1.1	Justificativa	14
1.2	Objetivo	15
1.2.1	Objetivos Específicos	15
1.3	Organização do trabalho	16
2	TRABALHOS RELACIONADOS	17
3	FUNDAMENTAÇÃO TEÓRICA	19
3.1	Realidade Aumentada	19
3.2	Nuvem de Pontos	21
3.3	Plataforma Tango	21
3.3.1	Motion Tracking	22
3.3.2	Depth Perception	23
3.4	Triangulação de Delaunay	24
3.5	Medições <i>As-Built</i>	27
4	MATERIAIS E MÉTODOS	29
4.1	Ferramentas	29
4.2	Método Proposto	30
4.2.1	Aquisição de Imagens e Nuvem de Pontos	30
4.2.2	Definição e Validação de Pontos de Interesse	32
4.2.3	Triangulação de Pontos	33
4.2.4	Obtenção de medidas e Cálculo de Área	34
4.2.5	Aplicativo e Visualização em Realidade Aumentada	35
5	RESULTADOS E DISCUSSÃO	39
5.1	Coeficiente de Correlação Intraclasse	39
5.2	Testes e Resultados	40
5.3	Discussões	41
6	CONCLUSÃO	44
	REFERÊNCIAS	45

1 Introdução

A realidade aumentada complementa o mundo real com informações ou objetos computacionalmente gerados, de forma que eles coexistam no mesmo espaço que o ambiente real, conforme afirma [Azuma et al. \(2001\)](#). Em contraste com Ambientes Virtuais (VE) onde o usuário é completamente imerso num ambiente sintético e não consegue ver o mundo real, realidade aumentada permite que o usuário veja o ambiente real ao seu redor com a adição de objetos virtuais superimpostos nele. Dessa forma, sistemas de AR complementam a realidade, em vez de substituí-la, e permitem combinar informações na perspectiva de visualização normal sem perder nenhuma das vantagens de movimentação em ambientes do mundo real ([WANG et al., 2014](#)).

Aplicações de Realidade Aumentada tem se tornado populares devido a sua capacidade de auxiliar usuários na realização de tarefas do mundo real, contribuindo para a melhora de seu desempenho. Por oferecerem informações que o usuário não consegue detectar automaticamente com seus próprios sentidos, tais aplicações podem melhorar a percepção que o mesmo tem do ambiente ao seu redor, bem como sua interação com o ambiente ([AZUMA, 1997](#)).

Nas últimas décadas, os avanços tecnológicos tem contribuído para o progresso da Realidade Aumentada e para a popularização de aplicações nas mais diversas áreas, como robótica, manufatura, indústria militar, indústria do entretenimento, educação, design, área médica etc. ([CHI; KANG; WANG, 2013](#)). Entretanto, o uso de aplicações de realidade aumentada nas áreas de Arquitetura, Engenharia e Construção (AEC) é consideravelmente limitado, apesar do crescimento e demanda por esse tipo de aplicação ([Gomes Jr., 2017](#)).

Uma pesquisa conduzida por [Bae, Golparvar-Fard e White \(2013\)](#) mostrou as vantagens do uso de sistemas de realidade aumentada nas áreas de AEC. Segundo os autores, tais aplicações podem melhorar o desempenho de tarefas físicas, reduzir a carga de trabalho mental de engenheiros e outros trabalhadores, podem proporcionar melhor cognição espacial, ajudando em atividades de *design*, e podem auxiliar na recuperação e apresentação de informações do local de construção, a fim de otimizar o tempo de realização de tarefas e para apresentar dados mais precisos.

Apesar da busca por aplicações de Realidade Aumentada em AEC, alguns fatores prejudicam seu uso. De acordo com [Azuma et al. \(2001\)](#), dentre os problemas que envolvem o desenvolvimento de aplicações de realidade aumentada está a operação em ambientes não-preparados, a necessidade de extensa calibração pra produção de registros precisos, percepção precisa da profundidade e processamento em tempo real, já que a latência causa sérios erros de registro e pode prejudicar o desempenho de tarefas. Outro tópico importante

mencionado pelo autor é a forma que o usuário deverá interagir com a aplicação e como as informações serão apresentadas no ambiente. Entretanto, o rápido avanço das tecnologias voltadas para a área de RA tem permitido lidar com esses fatores adversos (Gomes Jr., 2017).

Contudo, há obstáculos inerentes às próprias áreas de Arquitetura, Engenharia e Construção. De acordo com Golparvar-Fard et al. (2009), é um desafio melhorar desempenho de trabalhadores em tarefas relacionadas a AEC, pois vários fatores podem afetar a produtividade dos mesmos e o desenvolvimento da obra, como condições climáticas não previstas, ambientes de trabalho insalubres e até mesmo fatores humanos, como desatenção e cansaço. Atividades que envolvam mensurações e cálculos podem ser especialmente afetadas pela desatenção. A fim de minimizar erros e melhorar a produtividade em atividades de construção civil, a automatização de tarefas como essas é desejável e possível, especialmente em ambientes internos.

Com base nos fatores apresentados acima, este trabalho apresenta o desenvolvimento de uma aplicação móvel de realidade aumentada para automatizar mensurações de superfícies e cálculo de área, bem como estudo de viabilidade de tal aplicação em atividades voltadas às áreas de AEC. A ferramenta proposta visa ser utilizada em ambientes internos em tarefas como captura de dados *As-Built* para construção de registros de ambientes, para planejamento de ambientes internos ou simplesmente para obtenção de informações.

A aplicação proposta utiliza a plataforma para realidade aumentada da Google - Projeto Tango (GOOGLE, 2015) - para capturar dados de profundidade e para superimpor informações no ambiente analisado. Técnicas de Computação Gráfica são utilizadas para processar os dados de profundidade obtidos pela aplicação. Por fim, testes foram realizados para avaliar a fidedignidade e precisão das medidas obtidas com a ferramenta desenvolvida, a fim de comparar seu desempenho em relação aos métodos tradicionais de mensuração e determinar a viabilidade de uso da mesma.

1.1 Justificativa

Dispositivos para obter estimativas de distância utilizando sensores de visão têm sido amplamente utilizados em aplicações industriais, como em testes não destrutivos, engenharia reversa, manutenção preditiva e manufatura de produtos (AMANN et al., 2001). Seu uso em robótica também é especialmente importante, já que muitos algoritmos de posicionamento de robôs usam distância para calcular a posição do robô como base para tarefas mais complicadas. (LÁZARO et al., 2009). Entretanto, o uso de tais ferramentas nas áreas de AEC não é muito difundido, devido a dificuldade de trabalhar em tais cenários.

Portanto, mostra-se necessário verificar a aplicabilidade de uma ferramenta para mensurações automáticas no cenário de AEC através da análise dos resultados obtidos por

uma aplicação desse tipo. Com o constante crescimento do ramo da construção civil no Brasil, o número de empresas que tem se preocupado com a redução de custos em suas obras e aceleração do processo também tem aumentado (SERON; OLIVEIRA; SANTOS, 2009). A ferramenta proposta neste trabalho visa operar nesse cenário, a fim de reduzir custos, aumentar a produtividade e poupar tempo e recursos.

1.2 Objetivo

Este trabalho tem por objetivo o desenvolvimento de uma arquitetura para automatização da obtenção de medidas em obras civis (*As Built*) e a realização de testes de validação da arquitetura proposta.

As-Built é o levantamento métrico dos elementos e estruturas existentes numa área a fim de criar registros de ambiente, chamados de desenhos de *As-Built* (PINHAL, 2014). Usualmente, esses dados são obtidos através de métodos manuais de mensuração; a aplicação proposta neste trabalho visa automatizar a obtenção dessas medidas, excluindo a necessidade de contato direto com o ambiente analisado. Para atingir esse objetivo, será preciso escanear ambientes e analisá-los para extrair as medidas existentes, a fim de que os dados obtidos possam posteriormente ser transcritos em um desenho técnico que irá representar a atual estrutura do ambiente analisado. Como suporte para essa tarefa, será utilizada a plataforma para realidade aumentada, o Projeto Tango (GOOGLE, 2015), que está presente no dispositivo móvel Lenovo Phab 2 Pro. O dispositivo combina uma câmera de profundidade e técnicas de Visão Computacional para extrair informações da área observada. A aplicação a ser desenvolvida usará marcadores interativos, na qual o usuário delimita a área a ser analisada na tela de seu *smartphone*.

1.2.1 Objetivos Específicos

- Escanear ambientes e obter a nuvem de pontos do ambiente escaneado;
- Desenvolver interface interativa de modo que o usuário possa delimitar a área a ser analisada;
- Identificar planos;
- Implementar algoritmos de Triangulação que funcione em tempo real;
- Calcular a área total do polígono triangulado;
- Realizar testes de validação da aplicação desenvolvida em ambientes *indoor*, a fim de analisar a precisão das mensurações obtidas com a metodologia utilizada.

1.3 Organização do trabalho

Este trabalho está organizado em cinco capítulos, de modo que o capítulo 2 apresentará trabalhos relacionados que também visam a automatização de medidas usando realidade aumentada.

Em seguida, o capítulo 3 apresenta todas as técnicas e conceitos necessários para o desenvolvimento da aplicação proposta neste trabalho. Este capítulo subdivide-se em outros 6 tópicos: o primeiro explica o conceito de Realidade Aumentada, o segundo tópico trata do método utilizado para representação de dados de profundidade, o terceiro aborda sobre o projeto Tango (GOOGLE, 2015) e sua utilização na aplicação desenvolvida como tecnologia de realidade aumentada. O quarto tópico, por sua vez, apresenta uma técnica de computação gráfica para triangulação de pontos, enquanto que o quinto tópico explica o conceito de *As-Built*.

O capítulo 4 apresenta cada etapa utilizada no desenvolvimento da aplicação proposta, e os resultados obtidos através de testes com essa aplicação são apresentados e discutidos no capítulo 5; a métrica utilizada para avaliação dos resultados obtidos é explicada nessa mesma seção. Finalmente, o capítulo 6 traz a conclusão e considerações finais do trabalho.

2 Trabalhos Relacionados

Ferramentas para cálculo automático de medidas têm sido utilizadas nas mais diversas áreas, desde robótica até em biologia. As técnicas utilizadas para obtenção de medidas também são bem diversificadas; algumas utilizam lasers, outras utilizam câmeras comuns acompanhadas de técnicas de Visão Computacional. Algumas dessas ferramentas e técnicas são apresentadas neste capítulo.

Uma das formas mais comuns de obter distâncias ou saber o tamanho de objetos em imagens utiliza marcadores de referência. No método apresentado por [Rosebrock \(2015\)](#), um objeto de tamanho conhecido é colocado na imagem, para que o tamanho de outros componentes da cena possam ser estimados a partir da distância focal da câmera. Com a obtenção da distância focal, mesmo que o objeto mude de posição nas fotos ou *frames* seguintes, é possível obter a distância do objeto para câmera através da Similaridade de Triângulos. Esse método usa técnicas de Processamento de Imagens e Visão Computacional como detecção de contornos para identificar o objeto desejado na cena. Uma vez que o objeto de referência foi encontrado, pode-se utilizar a distância focal calculada para obter a medida de outros objetos na imagem.

Em um trabalho que também visa automatizar a coleta de informações para a construção de registros *As-Built*, [Liu, Eybpoosh e Akinci \(2012\)](#) usam uma câmera e um *scanner* a laser. Nesse trabalho, o *scanner* a laser produz uma nuvem de pontos, e a partir dela, todas as informações geométricas necessárias para fazer os registros *As-Built* são extraídas. Para manter referência da área analisada pelo *scanner*, imagens do ambiente são obtidas através de uma câmera comum. Posteriormente, os dados obtidos pelo *scanner*, da situação atual da obra ou área analisada, são combinados com as imagens da câmera e com os próprios registros de *As-Built* de antes do início da obra, para efeito de comparação. Portanto, a aplicação proposta pelos autores não funciona em tempo real, e necessita de pós-processamento para extração de dados e para a combinação dos dados obtidos pelos dois dispositivos utilizados.

Quanto às aplicações móveis em RA para obtenção automática de medidas, seu uso tem se popularizado com a chegada do ARKit ([APPLE, 2017](#)) e Tango ([GOOGLE, 2015](#)). O aplicativo *Air Measure App Demo* ([LABS, 2017](#)), disponível na *App Store*, funciona como uma fita métrica virtual e utiliza a plataforma para realidade aumentada da Apple para obter informações do ambiente e assim calcular distâncias. O aplicativo também permite adicionar móveis em 3D e traçar o contorno de ambientes, a fim de obter a planta baixa do local. Em relação às aplicações de RA utilizando a plataforma Tango, o aplicativo *Measure* ([GOOGLE, 2017a](#)), disponível na Play Store, realiza medidas simples permitindo

que usuários apontem o dispositivo para o que eles gostariam de medir e usem o cursor para colocar pontos. As medidas são exibidas em uma visualização de realidade aumentada. A aplicação *Easy Build* (WOSOMTECH, 2017) também faz medidas em 3D, além de permitir a geração de um *mesh* em 3D do ambiente escaneado, bem como a adição de objetos em 3D no ambiente.

Em relação à análise dos dados obtidas por dispositivo habilitado com a plataforma, como medidas e escaneamentos, o trabalho apresentado por Froehlich, Azhar e Vanture (2017) fez uma avaliação dos resultados obtidos por alguns aplicativos desenvolvidos com a plataforma Tango logo que a mesma foi lançada, em meados de 2015.

A metodologia desenvolvida neste trabalho não usa objetos de referência como em Rosebrock (2015), ou necessita de pós-processamento para combinar resultados obtidos com dois dispositivos diferentes como em Liu, Eybpoosh e Akinci (2012). A aplicação desenvolvida funciona em tempo real e em um dispositivo de custo mais baixo que o apresentado em Labs (2017), e diferente do aplicativo disponível em Google (2017a), além de informar distâncias, calcula área em tempo real. Os aplicativos *Measure Google* (2017a) e *Easy Build Wosomtech* (2017), encontram-se apenas disponíveis na Play Store, sem detalhes de seu funcionamento ou desenvolvimento. Em contrapartida, este trabalho apresenta a metodologia do desenvolvimento de uma aplicação em RA para cálculo automático de medidas, como também a análise dos resultados obtidos por essa aplicação a fim de determinar a viabilidade de seu uso na obtenção de dados *As-Built*.

3 Fundamentação Teórica

Nesta seção, apresentam-se os conceitos e técnicas que embasaram o desenvolvimento deste projeto. As seções e subseções a seguir explicam cada tema abordado na metodologia proposta neste trabalho.

3.1 Realidade Aumentada

De acordo com (AZUMA, 1997), Realidade Aumentada é uma variação de Ambientes Virtuais em que o usuário consegue ver o mundo real com objetos virtuais superimpostos ou compondo o ambiente. Em contraste com aplicações de Realidade Virtual, que imergem completamente o usuário num ambiente simulado e impedem que o mesmo veja o ambiente ao seu redor, aplicações de realidade aumentada complementam o mundo real, em vez de completamente substituí-lo. A figura 1 mostra o *Continuum* de realidade-virtualidade em que a realidade aumentada se encontra entre ambientes completamente sintéticos e ambientes completamente reais (MILGRAM; KISHINO, 1994).

Figura 1 – *Continuum* de realidade-virtualidade.



Fonte – Adaptado de Azuma et al. (2001).

A realidade aumentada permite ampliar ambientes reais com a adição de elementos virtuais (CARDOSO et al., 2007). Em realidade aumentada, informações computacionalmente geradas são associadas ao mundo real de forma que os usuários possam interagir com elementos virtuais, e que pareça que esses elementos fazem parte do ambiente real. Para isso, alguns critérios precisam ser atendidos. De acordo com (AZUMA et al., 2001), para desenvolver aplicações de realidade aumentada é preciso:

- Combinar objetos reais e virtuais num ambiente real;
- Funcionar de forma interativa e em tempo real;
- Registrar ou alinhar objetos virtuais com objetos reais.

Para incluir informações virtuais no ambiente real, as aplicações de realidade aumentada usam marcadores; esses marcadores, que podem ser do tipo fiduciais ou naturais [Gomes Jr. \(2017\)](#). Marcadores fiduciais ou artificiais são imagens distintas que possuem padrões visuais e dimensões bem-definidas. Tais marcadores podem ser identificados através de métodos de Visão Computacional e são utilizados como objetos de referência para superimpor objetos virtuais 3D na cena, conforme explica [Koch et al. \(2014\)](#). O autor ainda pontua que a principal desvantagem desse tipo de marcador é a necessidade de sua pré-instalação na cena, e caso tenha de ser colocado em muitos lugares numa mesma cena, seu uso se torna ineficiente.

Por outro lado, marcadores naturais baseiam-se nas características naturalmente visíveis, como objetos que já existem no ambiente, sem modificação ([AZUMA et al., 2001](#)). Para identificação de marcadores naturais, é necessário o processamento do ambiente observado através de técnicas de Processamento de Imagens, mas informações adicionais vindas de giroscópios, sensores térmicos e outros tipos de sensores também podem ser utilizadas. Ao utilizar marcadores naturais para adicionar objetos computacionalmente gerados, é preciso calcular o alinhamento do sistema de coordenadas do objeto virtual com o do mundo real para posicionar os objetos corretamente ([Gomes Jr., 2017](#)).

Para construir aplicações em realidade aumentada são necessários dispositivos para mesclar o ambiente real com o virtual. De acordo com [Azuma et al. \(2001\)](#), esses dispositivos podem ser categorizados em:

- **Head-worn displays:** Esse tipo de dispositivo é usado na cabeça do usuário, projetando as imagens na frente de seus olhos. O display visual é acoplado a cabeça, e possui dois tipos de visualização; a ótica e a visualização através de vídeo. No meio ótico, a informação virtual é projetada em um display transparente, enquanto que a segunda forma usa a câmera de vídeo acoplada na cabeça como um plano de fundo para a sobreposição das informações em RA.
- **Handheld displays:** Alguns sistemas usam dispositivos portáteis, como displays LCD com câmera acoplada que funcionam como uma espécie de janela, mostrando objetos reais com informações em RA sobrepostas.
- **Displays de projeção:** Nessa abordagem, a informação virtual é projetada diretamente no objeto físico desejado.

Aplicações em realidade aumentada podem ser extremamente úteis, pois elas têm o poder de aumentar ou melhorar a percepção que o usuário tem do mundo real, bem como a forma de interação com o mesmo. Isso acontece porque objetos virtuais adicionados à cena real oferecem informações que o usuário não consegue detectar automaticamente com seus próprios sentidos ([AZUMA, 1997](#)). Sendo assim, as informações apresentadas em

aplicações de realidade aumentada podem ajudar usuários a realizar tarefas do mundo real.

A aplicação proposta neste trabalho utiliza um *smartphone*, dispositivo do tipo *handheld*, para colocar informações virtuais no ambiente real. Tais informações são superimpostas nos lugares definidos pelo próprio usuário, através de marcadores naturais interativos. O usuário toca a tela de seu dispositivo a fim de definir pontos de interesse que delimitarão a área a ser analisada, bem como onde as informações serão colocadas. Os detalhes desse processo são apresentados no capítulo 4.

3.2 Nuvem de Pontos

Uma série de aplicações nas áreas de manufatura, medicina, geografia, design etc. requerem o escaneamento de objetos tridimensionais complexos a fim de incorporá-los em processos computacionais. Com isso, técnicas para medição e reconstrução de superfícies foram aprimoradas para produzir uma grande quantidade de pontos que representasse a superfície de objetos (LINSEN, 2001).

A reconstrução de superfícies é um processo no qual objetos 3D são inferidos, ou reconstruídos a partir de uma coleção de pontos discretizados que representam a forma observada (BERGER et al., 2017). Tal conjunto de pontos representando a superfície ou as bordas de objetos tridimensionais é chamado de nuvem de pontos; essa estrutura pode ser utilizada para modelagem, reconhecimento e análise do mundo real.

De acordo com Rusu e Cousins (2011), nuvem de pontos é uma estrutura de dados usada para representar uma coleção de pontos multidimensionais, quando usada para representar dados tridimensionais, cada ponto presente na nuvem representa uma coordenada geométrica X, Y e Z da superfície amostrada. Nuvem de pontos podem ser obtidas através de sensores como câmeras *stereo*, *scanners* 3D, e câmeras do tipo *Time of Flight* (RUSU; COUSINS, 2011). Na metodologia proposta por esse trabalho, nuvem de pontos foi a forma utilizada para representação de dados de profundidade e para reconstrução de superfícies. A forma de obtenção desses dados será explicada na seção seguinte, e sua utilização será apresentada no capítulo 4.

3.3 Plataforma Tango

Tango é um projeto promovido pela Google para desenvolvimento de aplicações de realidade aumentada em dispositivos móveis, com o auxílio de hardware especializado (GOOGLE, 2017b). A plataforma Tango faz uso de técnicas de Visão Computacional para permitir que dispositivos possam saber sua posição em relação ao ambiente ao redor, rastrear movimentação, reconhecer espaços e escanear ambientes em 3D. Para isso, Tango

faz uso de 3 tipos de tecnologia: *Motion Tracking*, *Area Learning* e *Depth Perception*. Na metodologia proposta neste trabalho, dois desses módulos foram utilizados. Cada um deles será explicado nas subseções a seguir.

3.3.1 Motion Tracking

O módulo de rastreamento de movimentos, do inglês *Motion Tracking* permite que o dispositivo utilizado saiba sua posição à medida em que se move em um ambiente (TANGO, 2017b). A plataforma Tango implementa o rastreamento de movimentos usando Odometria Visual-Inercial (VIO) para estimar onde o dispositivo se encontra em relação a onde ele estava inicialmente.

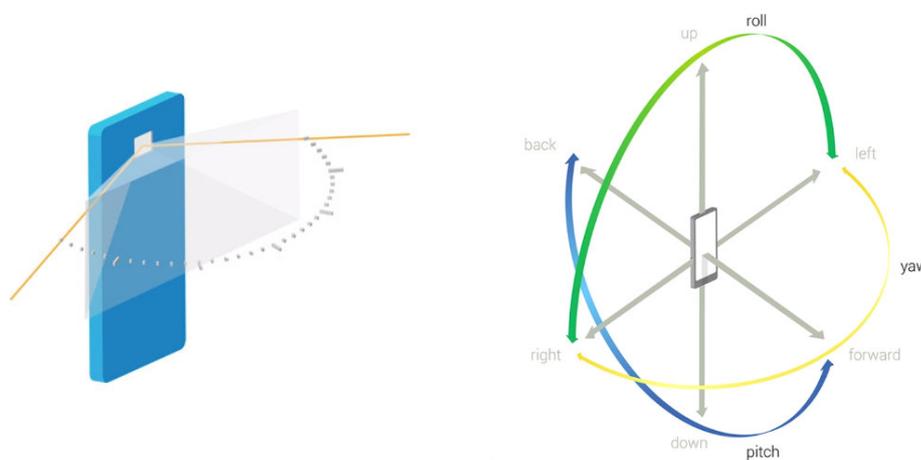
A Odometria Visual (VO) padrão usa imagens da câmera para determinar mudanças na posição, observando a posição relativa de diferentes recursos nessas imagens (TANGO, 2017b). Nenhum conhecimento prévio da cena nem do movimento é necessário; as estimativas são obtidas baseando-se apenas em informações visuais. Entretanto, VO também pode ser usada em conjunto com informações de outras fontes, como GPS e sensores inerciais (NISTÉR; NARODITSKY; BERGEN, 2004). Daí surge o conceito de Odometria Visual-Inercial utilizado nos dispositivos Tango. A VIO complementa a odometria visual com sensores de movimento inerciais capazes de rastrear a rotação e aceleração de um dispositivo. Dessa forma, dispositivos integrados com a plataforma Tango podem estimar sua orientação e movimentação dentro de um espaço 3D com uma precisão ainda maior. Ao contrário do GPS, o rastreamento de movimentos usando VIO funciona em ambientes fechados (TANGO, 2017b).

Para proporcionar a VIO, a plataforma Tango coleta dados de uma câmera do tipo *fish-eye* (olho de peixe), mostrada na Figura 2a e de uma Inertial Measurement Unit (IMU). A imagem da câmera é utilizada para identificar características como cantos, e manter registro do quanto esses cantos de moveram de um *frame* pra outro, a fim de calcular a distância percorrida. A plataforma Tango usa lentes do tipo olho de peixe de ângulo amplo com 160 graus, para capturar os detalhes do ambiente. A IMU usa um acelerômetro e um giroscópio pra verificar quão rápido o dispositivo se move e em qual direção ele está indo. As imagens se juntam com os dados obtidos pelos sensores IMU pra calcular quanto o dispositivo se moveu. Isso acontece em tempo real, cerca de 100 vezes por segundo, o que faz com que o módulo de *Motion Tracking* aconteça de forma sutil, dando ao olho humano um senso de movimento contínuo (TANGO, 2017b).

A API do Tango informa a posição e orientação do dispositivo através de poses, que são a combinação da posição e orientação do dispositivo nas direções mostradas na Figura 2b . A API provê duas formas de obter dados de pose: através de *callbacks* para obter as atualizações de pose mais recentes e através de funções para obter uma estimativa de pose em um momento específico. Os dados são informados em duas partes: um vetor

Figura 2 – Câmera usada em Motion Tracking, e possíveis orientação e direção.

- (a) Exemplo de uma câmera de ângulo amplo usada nos dispositivos Tango. (b) Direções e eixos de rotação representadas na pose.



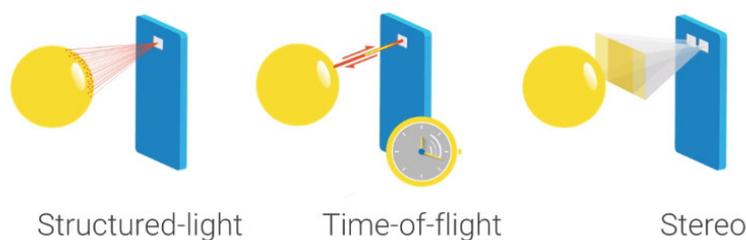
Fonte – (TANGO, 2017b)

em metros representando a translação e um *quaternion* para rotação (TANGO, 2017b).

3.3.2 Depth Perception

O módulo de Percepção de Profundidade, do inglês *Depth Perception* permite que uma aplicação possa entender as distâncias entre objetos no mundo real. Para recuperação desse tipo de informação, os dispositivos equipados com a plataforma Tango podem fazer uso de tecnologias como *Structured Light* (Luz estruturada), *Time of Flight* (Tempo de Voo) e Estéreo (TANGO, 2017a), conforme mostra a Figura 3.

Figura 3 – Exemplo de tecnologias usadas na captura de dados de profundidade.



Fonte – (TANGO, 2017a)

Sistemas de Luz estruturada e Tempo de Voo fazem uso de sensores e projetores infravermelho (IR), enquanto que sistemas Estéreos não fazem uso deles. Sistemas de

Luz Estruturada enviam padrões de pontos infravermelhos para iluminar o contorno do ambiente. O dispositivo então mede o tamanho dos pontos infravermelhos e quanto maior o ponto, mais distante ele está. Sistemas de Tempo de Voo usam sensores especiais para medir a profundidade baseando-se no tempo que leva pra transmitir um raio infravermelho e captar seu reflexo. Para distâncias pequenas, o raio leva apenas alguns nanosegundos para fazer a viagem de ida e volta. Há ainda sistemas Estéreos, que funcionam de forma análoga ao olho humano, e capturam ambientes utilizando duas câmeras separadas por uma curta distância, e inferem dados de profundidade usando triangulação. Os sensores baseados em infravermelho como Luz estruturada e Tempo de Voo funcionam particularmente bem em ambientes internos, com iluminação controlada (TANGO, 2017a).

Os dispositivos equipados com a plataforma Tango retornam dados de profundidade em forma de nuvem de pontos, uma representação não tratada de como os dados são, como mostra a Figura 4. Esse formato fornece coordenadas do tipo (X, Y, Z, C) para a maior quantidade de pontos que for possível obter numa determinada cena. X, Y, Z são coordenadas dadas em metros, enquanto que C é uma medida de confiança que varia entre $[0,1]$, onde 1 significa confiança total. Z aponta na direção do eixo ótico da câmera, perpendicular ao plano da câmera. X aponta para a direita do usuário e Y aponta para a parte inferior da tela. A origem é o centro focal da câmera de profundidade (TANGO, 2016).

O dispositivo utilizado no desenvolvimento deste projeto faz uso de sensores e projetores (IR) para a percepção de profundidade (VELAZCO, 2016), utilizando a tecnologia *Time of Flight* (CHENG, 2016). Mais informações sobre esse dispositivo são apresentadas na seção 4.

3.4 Triangulação de Delaunay

Triangulação é um dos tópicos mais importantes em Geometria Computacional, sendo utilizada nas mais diversas áreas, como em Computação Gráfica, visualização científica, robótica, Visão Computacional, síntese de imagens, bem como matemática e ciências naturais (CIGNONI; MONTANI; SCOPIGNO, 1998). Uma triangulação particiona um polígono ou conjunto de pontos em triângulos, o que permite facilitar o cálculo da área total de um polígono ou representar um conjunto de pontos. De acordo com Gärtner e Hoffmann (2013), calcular a área de um polígono não ocorre de forma direta; geralmente se descreve o polígono em termos de objetos geométricos mais simples, para o qual o cálculo da área é fácil. Um bom candidato para tal forma é o triângulo.

O conceito de triangulação pode ser definido a partir de um grafo planar, conforme mostra Anglada (1997). Seja V um conjunto finito de n vértices no \mathbb{R}^2 e seja A um conjunto de arestas determinadas pelos vértices V . Um grafo planar é o par $G = (V, A)$ em que:

Figura 4 – Exemplo de dados de profundidade em forma de nuvem de pontos obtido por um dispositivo equipado com a plataforma Tango.



Fonte – (TANGO, 2017a)

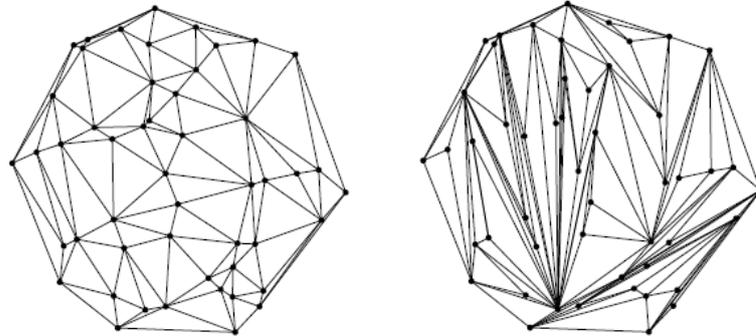
- para cada aresta $ab \in A$, $ab \cap V = \emptyset$
- para cada par de arestas ab, cd onde $ab \neq cd$ em A , $ab \cap cd = \emptyset$

Ou seja, arestas diferentes não se interceptam e além de seus vértices constituintes, nenhum outro vértice intercepta uma aresta. Sendo assim, uma triangulação é um grafo planar $T = (V, A)$ em que A é maximal, e as arestas de A incluem o fecho convexo de V e se subdividem em faces triangulares que não se sobrepõem, formando assim os triângulos de T .

Dentre as triangulações mais conhecidas, encontram-se a Triangulação de Delaunay, representada na Figura 5a e Triangulação de Scan, representada na Figura 5b. A triangulação do tipo Scan costuma apresentar triângulos deformados e não regulares. Em contrapartida, o método de Delaunay produz uma triangulação que maximiza o ângulo mínimo de cada triângulo, garantindo uma triangulação o mais equiangular possível e isso faz com que os triângulos tenham mais regularidade (GÄRTNER; HOFFMANN, 2013). Conforme pontua Piteri e JUNIOR (2007), a garantia de regularidade local e global dos triângulos produzidos pelo método de Delaunay é de extrema importância na interpolação numérica de funções de duas variáveis, na geração de malhas de elementos finitos e na modelagem digital de terrenos. Portanto, a triangulação usada nesse trabalho foi a do tipo Delaunay.

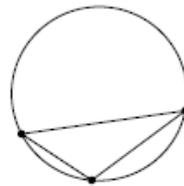
Figura 5 – Tipos de triangulação.

(a) Triangulação do tipo Delaunay. (b) Triangulação do tipo Scan.



Fonte – (GÄRTNER; HOFFMANN, 2013)

Figura 6 – Condição do circuncírculo vazio.



Fonte – (GÄRTNER; HOFFMANN, 2013)

Cada triângulo gerado pela triangulação de Delaunay deve respeitar o critério de circuncírculo vazio. De acordo com Gärtner e Hoffmann (2013), uma triangulação de um conjunto de pontos finitos $P \subset \mathbb{R}^2$ é chamada de uma triangulação de Delaunay, se o círculo que contém os três pontos de cada triângulo estiver vazio, isto é, se não há nenhum ponto de P em seu interior, todos os pontos do triângulo devem estar na circunferência. As Figuras 6 e 7 mostram exemplos de triângulos que satisfazem ou não essa condição.

Figura 7 – Etapas da Triangulação de Delaunay.



Fonte – (NOGUEIRA, 2013)

Há três principais classes de algoritmos usadas na implementação da triangulação de Delaunay, sendo elas: paradigma incremental, paradigma de divisão e conquista, e paradigma de varredura de linhas. O método escolhido para o desenvolvimento deste

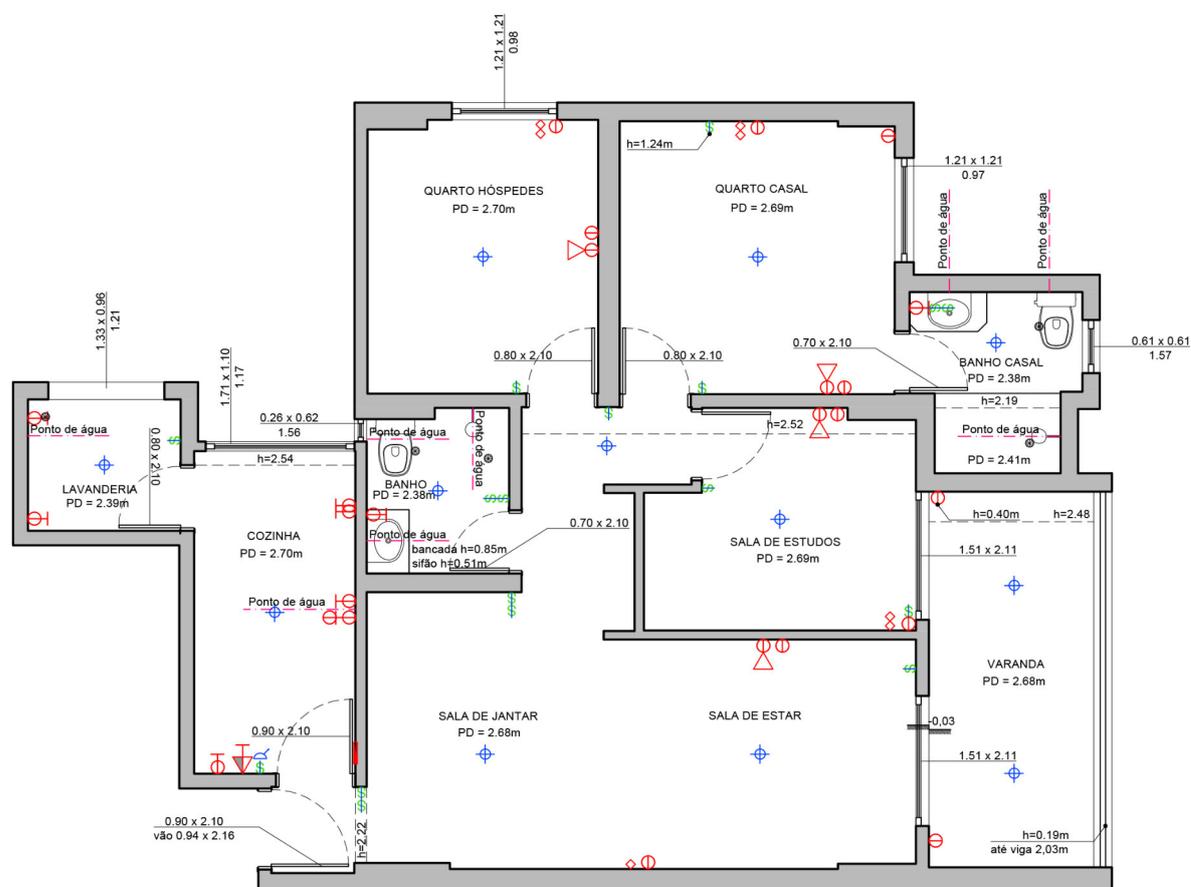
trabalho foi o paradigma incremental; detalhes de sua implementação bem como análise de sua complexidade são apresentados no capítulo 4. No método proposto, escolheu-se triangulação porque no futuro, pretende-se estender o uso da aplicação para poliedros também.

3.5 Medições *As-Built*

Um termo muito utilizado nas áreas de AEC é o *As-Built*, uma expressão inglesa que significa "como construído". *As-Built* se refere ao levantamento métrico dos elementos e estruturas existentes em um ambiente, e os dados obtidos são usados em inspeções, em comparações com a área planejada e na construção de registros de ambiente. Registros *As-Built* são desenhos técnicos que representam a atual situação dos componentes de um ambiente. Eles podem ser complexos, incluindo informações estruturais, elétricas e hidráulicas de toda uma área, ou podem ser simples, representando as informações de um móvel, por exemplo, (PINHAL, 2014). Dados e registros *As-Built* são usados para catalogar o progresso de uma obra, na elaboração de estudos e projetos, em inspeções, e auxiliam na tomada de decisões e em futuras intervenções, pois reúnem todas as informações métricas do ambiente (VECTOR, 2017).

O método proposto nesse trabalho visa auxiliar a obtenção dados *As-Built* por meio de uma aplicação móvel onde as informações obtidas são superimpostas no ambiente observado através de realidade aumentada. As imagens do ambiente analisado com as medidas obtidas pelo aplicativo podem auxiliar o processo de construção de registros *As-Built*. A figura 8 mostra um exemplo de desenho de *As-Built*.

Figura 8 – Registro de *As-Built* de uma residência.



PLANTA DE LEVANTAMENTO
ESCALA 1:50



Fonte – (MITSUSE, 2014)

4 Materiais e Métodos

Neste capítulo apresenta-se a sequência de etapas adotada para a obtenção da nuvem de pontos, captura dos pontos demarcados pelo usuário, geração da malha de triângulos e cálculo da área total do polígono definido pelo usuário. Primeiramente, apresentam-se os softwares e hardware utilizados no desenvolvimento deste trabalho, em seguida, cada fase constituinte desta metodologia será descrita nas seções seguintes.

4.1 Ferramentas

Para desenvolvimento da aplicação móvel proposta neste trabalho, foi utilizada a *engine* Unity; uma plataforma para desenvolvimento de jogos e aplicações em 2D e 3D para uso em *desktops*, consoles e dispositivos móveis. A aplicação desenvolvida foi testada no dispositivo Android Lenovo Phab 2 Pro e a linguagem de programação utilizada foi C#.

Figura 9 – Lenovo Phab 2 Pro.



Fonte – (CHENG, 2016)

Phab 2 Pro é um *smartphone* de 6.4 polegadas desenvolvido pela Lenovo que possui 4 câmeras e dá suporte a plataforma Tango. Na parte de trás do *smartphone* há uma câmera RGB de 16 MP, uma câmera de profundidade e uma câmera do tipo olho-de-peixe de ângulo amplo, enquanto que na frente há uma câmera de 8 MP; seu processador é o Qualcomm Snapdragon, que foi otimizado para funcionar de forma eficiente com a plataforma Tango (LENOVO, 2017). A Figura 9 mostra o *smartphone* utilizado.

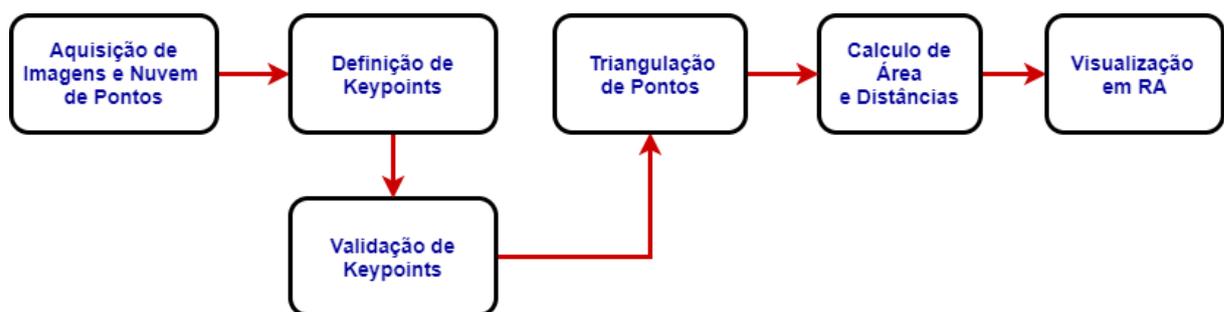
O dispositivo apresentado foi projetado para funcionar bem em ambientes internos a distâncias entre 0.5 a 4 metros, pois essa configuração proporciona dados de profundidade confiáveis e permite que o *smartphone* equilibre bem a iluminação IR e processamento

de profundidade (TANGO, 2017a). É importante lembrar que Tango e Android não são sistemas fortemente de tempo real, isto acontece porque o Kernel Linux do Android não pode fornecer fortes garantias sobre o tempo de execução do software que está sendo executado no dispositivo. Como resultado, o Projeto Tango é considerado um sistema levemente de tempo real (TANGO, 2017b).

4.2 Método Proposto

Esta seção apresenta cada etapa do método proposto para desenvolvimento de aplicação móvel para mensuração semiautomática de ambientes e superfícies utilizando técnicas de Realidade Aumentada e Computação Gráfica em tempo real. A Figura 10 apresenta a sequência de etapas envolvidas na progressão deste trabalho de uma forma geral. Cada etapa da metodologia será explicada neste capítulo, e uma análise dos resultados obtidos por essa metodologia será apresentada no capítulo 5.

Figura 10 – Etapas da Metodologia Proposta.



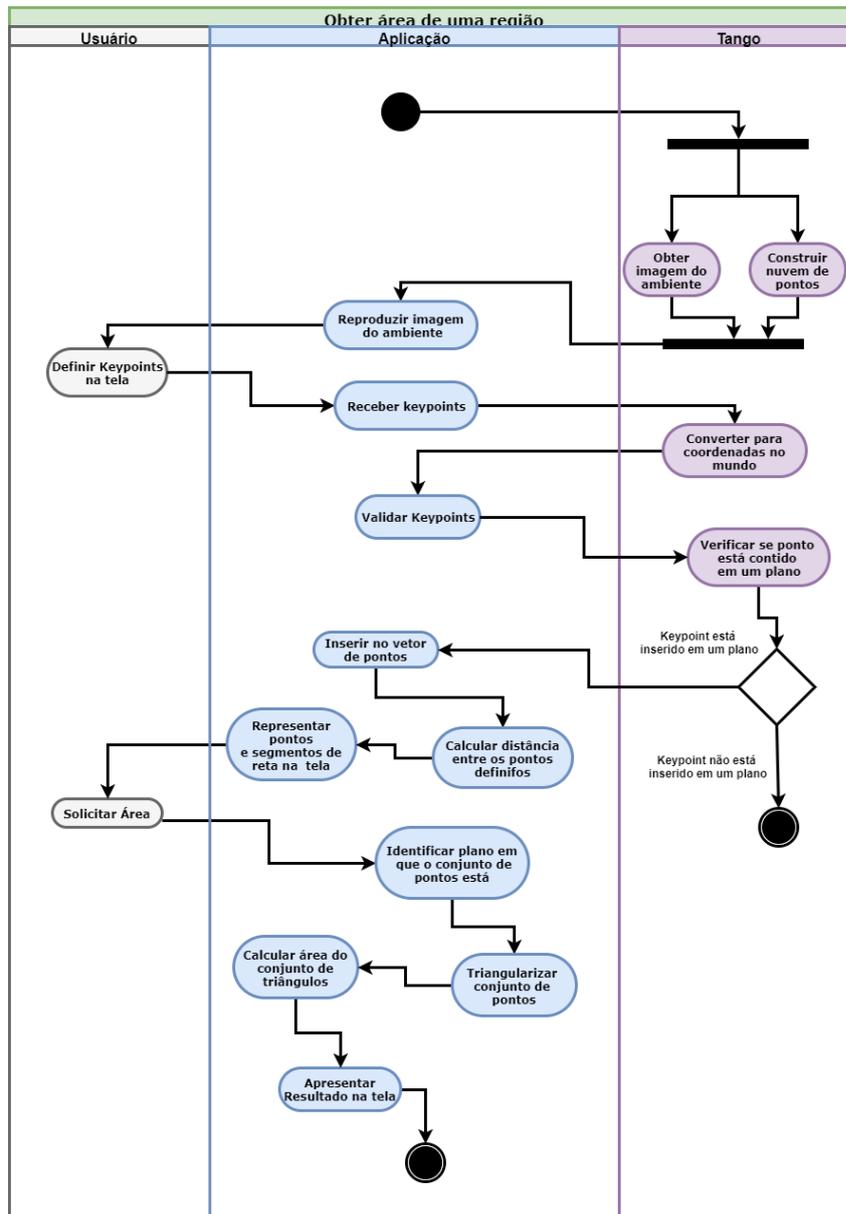
Fonte – Acervo da autora.

De uma forma mais detalhada, a Figura 11 apresenta o diagrama de atividades que descreve os passos a serem executados para o desempenho da principal operação da aplicação proposta, a obtenção de medidas e de área de uma região definida pelo usuário. O diagrama apresenta as ações que serão executadas, o que elas fazem, a sequência em que elas ocorrem e quem as executa. As subseções a seguir explicam em detalhes cada uma dessas ações.

4.2.1 Aquisição de Imagens e Nuvem de Pontos

Nessa etapa, a imagem do ambiente em que o usuário se encontra é capturada pela câmera RGB do *smartphone* utilizado e é apresentada no aplicativo, sendo atualizada a cada novo *frame* em um determinado *framerate*. Concorrentemente, a nuvem de pontos do ambiente observado é construída utilizando o sensor de profundidade do dispositivo móvel, através da tecnologia de captura de dados de profundidade *Time of Flight*. Dados de profundidade são obtidos desde a posição em que o usuário começou a utilizar o aplicativo

Figura 11 – Diagrama de atividades da principal funcionalidade da aplicação.



Fonte – Acervo da autora.

(origem), até uma distância de quatro metros da origem. Para habilitar a captura e processamento desses dados, o SDK da plataforma Tango para Unity foi utilizado.

Para dar início ao funcionamento da aplicação, um objeto do tipo *TangoApplication* é instanciado e assim que *smartphone* se conectar ao serviço do Tango, o *rate* da câmera de profundidade é definido. Em seguida, um objeto do tipo *TangoUnityDepth* é criado e recebe informações vindas da câmera de profundidade, enquanto que *TangoPointCloud* é o objeto referente à nuvem de pontos da cena. Todas as operações com nuvem de pontos são feitas a partir deste objeto, por exemplo, dado uma posição na tela do celular encontrar o ponto correspondente na nuvem de pontos. Essa funcionalidade é utilizada na próxima etapa do método proposto.

4.2.2 Definição e Validação de Pontos de Interesse

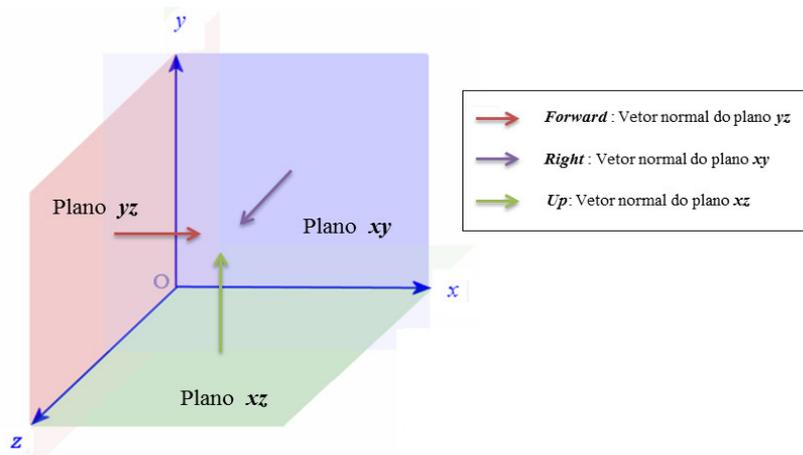
Como mencionado brevemente no capítulo 3, a aplicação desenvolvida usa marcadores naturais interativos, ou seja, o próprio usuário define os pontos referentes à parte do ambiente visualizado que deseja analisar. Os pontos são obtidos através de toques na tela do *smartphone* e pontos são usados como referência para o local de inclusão de informações virtuais no ambiente real. Ao tocar na tela, o usuário define um ponto de interesse, que na aplicação é representado por uma esfera simples; a cada dois pontos um segmento de reta é traçado entre esses pontos e a distância entre eles é apresentada.

Para que as informações virtuais sejam apresentadas corretamente no mundo real, é preciso que os pontos definidos pelo usuário sejam válidos, caso contrário o usuário terá a impressão de que os objetos virtuais estão flutuando no ar. Um ponto de interesse válido pertence a um plano; para verificar se essa condição se satisfaz é preciso converter as coordenadas do ponto (X, Y) referente à posição da tela em que o usuário clicou para coordenadas do mundo real. Em seguida, verifica-se se esse ponto pertence a um plano, se pertencer, o ponto de interesse é adicionado ao vetor de pontos e um objeto virtual do tipo esfera é colocado no lugar correspondente no mundo real. Tanto o método para converter coordenadas de tela para coordenadas na nuvem de pontos, quando o método para verificar se um ponto em coordenadas de tela pertence a um plano no mundo real são fornecidos pelo SDK da plataforma Tango, a partir dos métodos da classe *TangoPointCloud*.

Além de verificar se cada ponto definido pelo usuário pertence a um plano, é preciso identificar o plano ao qual um conjunto de pontos pertence, a fim de calcular a área do polígono formado por esses pontos. A área é dada como quantidade de espaço bidimensional, mas as coordenadas no mundo real são dadas em três dimensões. Portanto, verifica-se a qual dos três planos apresentados na Figura 12 o conjunto de pontos pertence. Para isso, é necessário que os pontos definidos pelo usuário sejam coplanares, e o plano a que esses pontos pertencem (Plano XY , XZ ou YZ) é dado pelo vetor normal do plano.

O plano XY apresenta vetor normal do tipo *Right* (Direita) de coordenadas $(1, 0, 0)$, ou o vetor *Left* (Esquerda) com coordenadas $(-1, 0, 0)$, de mesma direção que *Right* mas de sentido oposto. O plano XZ por sua vez, apresenta vetor normal do tipo *Up* (Para cima) de coordenadas $(0, 1, 0)$, ou o vetor *Down* (Para baixo) com coordenadas $(0, -1, 0)$. Já o plano YZ apresenta vetor normal do tipo *Forward* (Para frente) de coordenadas $(0, 0, 1)$, ou o vetor *Back* (Para trás) com coordenadas $(0, 0, -1)$. Compara-se o vetor normal do plano ao qual os pontos definidos pelo usuário pertencem com os vetores mencionados acima. Após determinar o plano, caso o usuário selecione a opção referente ao cálculo de área na interface da aplicação, a próxima etapa da metodologia proposta é acionada, recebendo como parâmetro os pontos de interesse definidos nessa etapa convertidos em coordenados do plano ao qual eles pertencem.

Figura 12 – Tipos de planos e seus vetores normais.



Fonte – Acervo Pessoal

4.2.3 Triangulação de Pontos

Após definir os pontos de interesse referentes à área analisada, o usuário pode verificar a área total do polígono formado por esses pontos ativando a opção correspondente na interface. Ao fazer isso, o conjunto de pontos passa pelo algoritmo de Triangulação de Delaunay, resultando em um conjunto de triângulos que são usados para cálculo de área. Essa técnica é utilizada para que a área possa ser calculada da mesma maneira pra todos os polígonos, independente do formato dos mesmos.

O algoritmo escolhido para triangulação de Delaunay foi do tipo incremental devido à sua simplicidade e robustez. O algoritmo foi proposto por [Guibas, Knuth e Sharir \(1992\)](#) e apresenta complexidade $O(n \log n)$; o pseudocódigo de seu funcionamento é mostrado em *Algorithm 1*.

O algoritmo recebe como entrada o conjunto de pontos de interesse obtido na etapa anterior da metodologia e procura um triângulo grande o suficiente para conter todos os pontos do conjunto informado. A partir daí, cada ponto é inserido um por um na triangulação e procura-se qual triângulo contém aquele ponto. Para isso, usa-se a condição do circuncírculo vazio para saber se ponto inserido é interno a algum dos triângulos já existentes. Tal condição é verificada pelo cálculo do determinante, que se baseia nas coordenadas dos pontos P_a, P_b, P_c do triângulo em análise e do ponto P_i recém adicionado.

$$Det(P_a, P_b, P_c, P_i) = \begin{vmatrix} x_{P_a} & y_{P_a} & x_{P_a}^2 + y_{P_a}^2 & 1 \\ x_{P_b} & y_{P_b} & x_{P_b}^2 + y_{P_b}^2 & 1 \\ x_{P_c} & y_{P_c} & x_{P_c}^2 + y_{P_c}^2 & 1 \\ x_{P_i} & y_{P_i} & x_{P_i}^2 + y_{P_i}^2 & 1 \end{vmatrix} \quad (4.1)$$

Se o determinante for menor que 0, o ponto P_i está fora do circuncírculo do

Algorithm 1 Triangulação de Delaunay**input** : Conjunto de P de n pontos no plano**output** : Conjunto T_p de triângulos

1. Inicialize T_p com o triângulo $P_1P_2P_3$ que contém todos os pontos de P ;
- for** $i \leftarrow 1$ **to** n **do**
 2. insira P_i em T_p ;
 3. encontre o triângulo $P_aP_bP_c$ que contém P_i ;
 - if** P_i está dentro de $P_aP_bP_c$ **then**
 4. insira arestas de P_i para os vértices P_a, P_b, P_c , resultando em três triângulos;
 5. verifique se cada novo triângulo é valido;
 6. insira os novos triângulos em T_p ;
 - else if** P_i está em uma das arestas de $P_aP_bP_c$ **then**
 7. remove essa aresta do triângulo;
 8. insira uma aresta entre P_i e o vértice que não faz parte da aresta removida, resultando em dois triângulos;
 9. insira o vértice P_i no triângulo adjacente ao triângulo P_a, P_b, P_c , resultando em dois triângulos;
 5. verifique se cada novo triângulo é valido;
 6. insira os novos triângulos em T_p ;
- end**
10. apague o triângulo $P_1P_2P_3$ e suas arestas que incidem em T_p ;

triângulo observado e a busca continua. Se for maior que 0, o ponto P_i está dentro do circuncírculo que contém o $\Delta P_aP_bP_c$; a partir daí três novos triângulos são gerados e a condição do circuncírculo vazio é novamente verificada. Caso não seja satisfeita, as arestas formadas pelos vértices P_a, P_b, P_c e P_i são trocadas até que os triângulos gerados satisfaçam a condição. Em outro caso, quando o determinante é igual que 0, o ponto P_i pertence a circunferência que contém o triângulo, estando em uma de suas arestas. Nesse caso procura-se a aresta em que P_i está contido e procura-se o triângulo adjacente que também contém essa aresta, gerando quatro novos triângulos. Os triângulos são analisados e mudados, se necessário, e o resultado do algoritmo é um conjunto de triângulos em que o ângulo mínimo é maximizado. Esse conjunto é passado para próxima etapa do método proposto.

4.2.4 Obtenção de medidas e Cálculo de Área

A área total do polígono formado pelos pontos de interesse definidos pelo usuário é calculada a partir do conjunto de triângulos obtidos na etapa anterior, calculando a área de cada triângulo e somando as áreas de todos eles. Para cálculo da área do triângulo utilizou-se a fórmula de Heron, por sua praticidade e porque ela pode ser aplicada para todo tipo de triângulo. Como mostra [Weisstein \(2003\)](#), a fórmula de Heron utiliza apenas a medida de cada lado do triângulo e dispensa cálculos adicionais como obtenção de ângulos. Primeiro calcula-se o semiperímetro do triângulo de vértices A, B, C , dado pela fórmula

4.2, e em seguida usa-se a fórmula 4.3 para obter o valor da área total do triângulo.

$$S = \frac{\overline{AB} + \overline{AC} + \overline{BC}}{2} \quad (4.2)$$

$$A = \sqrt{S(S - \overline{AB})(S - \overline{AC})(S - \overline{BC})} \quad (4.3)$$

Para obter o tamanho das arestas \overline{AB} , \overline{AC} , \overline{BC} de cada triângulo gerado, usa-se a fórmula da distância euclidiana para dois pontos no plano. Por exemplo, dado os pontos p e q no plano de coordenadas $p = (a, b)$ e $q = (c, d)$, a distância entre eles é dada por :

$$D(p, q) = \sqrt{(a - c)^2 + (b - d)^2} \quad (4.4)$$

Em espaços *n-dimensionais*, a fórmula 4.5 é usada:

$$D(p, q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (4.5)$$

Na fase de aquisição de pontos de interesse, a cada dois pontos definidos pelo usuário, a distância euclidiana entre esses pontos também é calculada e seu resultado é mostrado na aplicação.

4.2.5 Aplicativo e Visualização em Realidade Aumentada

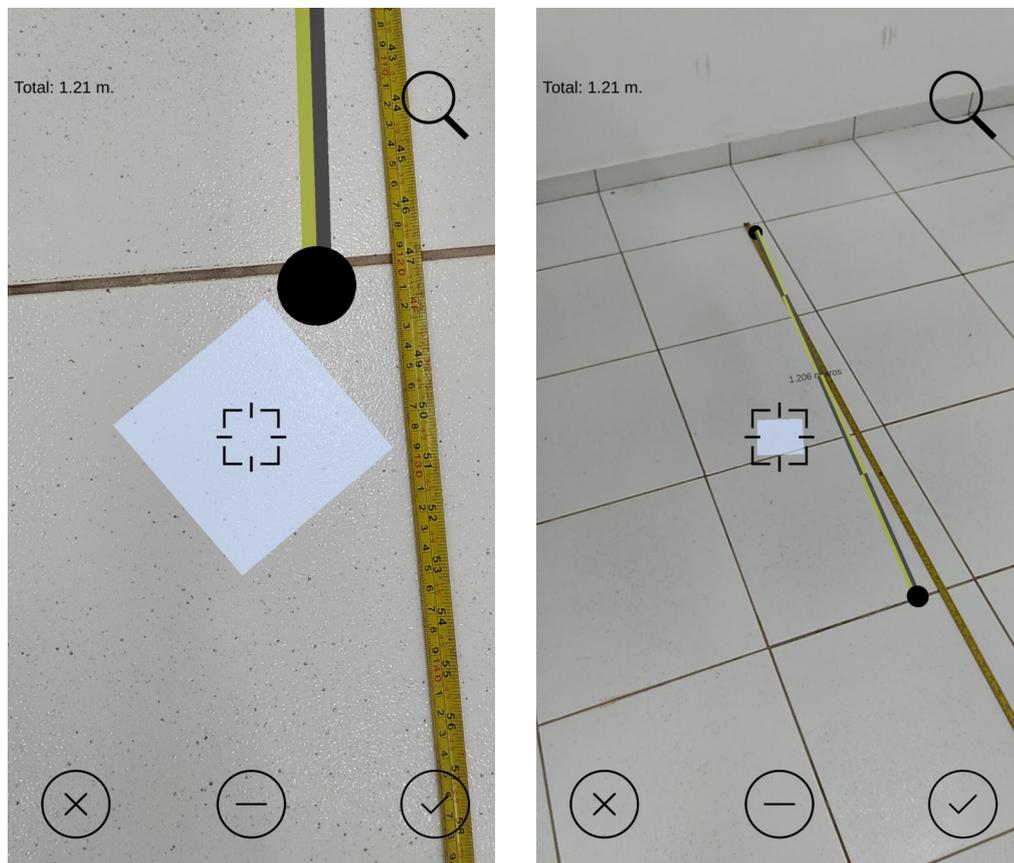
O resultado do método proposto é um aplicativo para dispositivo móvel, no qual as informações obtidas nas etapas anteriores são apresentadas na interface da aplicação, seguindo os conceitos apresentados na seção 3.1 para proporcionar uma experiência completa em realidade aumentada. Além da identificação de planos apresentada na seção 4.2.2, algo fundamental para visualização correta em realidade aumentada é o módulo de *Motion Tracking* da plataforma Tango; ele permite que o usuário se mova livremente pelo ambiente sem que isso afete a visualização dos objetos virtuais colocados na cena, que permanecem no lugar em que foram colocados inicialmente ainda que o usuário se mova. Além disso, mesmo que o usuário mude a perspectiva da visualização, é importante que as informações adicionadas na cena mudem de perspectiva também, aumentando de tamanho a medida que o usuário se aproxima, ou diminuindo de tamanho a medida que ele se afasta, como é visto nas Figuras 13a e 13b. Para proporcionar isso, cada objeto virtual sempre olha para a câmera da aplicação.

A interface do aplicativo é formada por 4 botões: o botão (X) apaga todas as informações colocadas na tela, o botão (-) apaga o último ponto de interesse adicionado, o botão (V) informa a área da figura formada pelos pontos definidos, e o botão de lupa no canto superior direito dá informações sobre a aplicação. Para inserir pontos de interesse o usuário só precisa clicar em qualquer parte da tela, e o ponto representado pelo foco da

câmera será inserido no vetor de pontos de interesse, se este for válido. A fim de guiar o usuário, se o ponto referente ao foco atual da câmera estiver contido num plano no mundo real, um plano virtual é apresentado na tela, se o plano não aparecer, o foco da câmera está fora do alcance representado na nuvem de pontos ou o ponto não está em um plano, como mostra a Figura 14b. Esferas representam os pontos de interesse definidos pelo usuário, e um segmento de reta é definido a cada dois pontos, com uma *label* apresentando o tamanho do segmento em metros. No canto superior direito pode-se ver outra *label* que informa o tamanho de todos os segmentos formados, ou o perímetro de um polígono, como mostra a Figura 14a. As Figuras 15a e 15b mostram outros exemplos de execução da aplicação.

Figura 13 – Informação virtual vista de diferentes perspectivas.

- (a) Comparação entre a medida da aplicação e a da trena. (b) Aplicação observada de um outro ângulo.

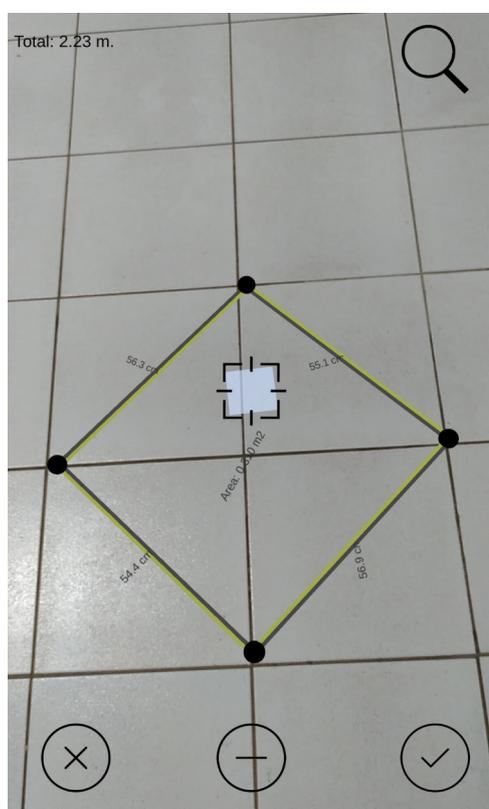


Fonte – Acervo da autora.

Figura 14 – Exemplos de funcionamento da aplicação.

(a) Polígono com a área e perímetro informados, bem como o tamanho de cada aresta.

(b) Exemplo de uma situação em que um usuário foca em um ponto que está fora do alcance.

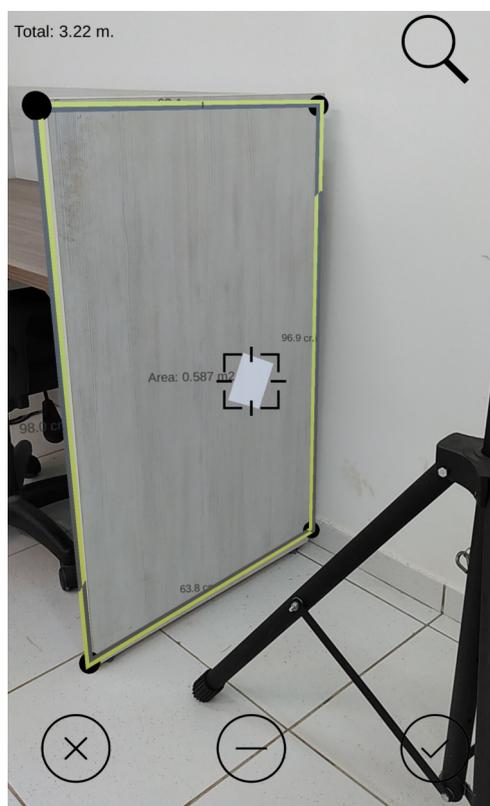


Fonte – Acervo da autora.

Figura 15 – Mais exemplos de funcionamento da aplicação.

(a) Exemplo de medidas obtidas com o aplicativo desenvolvido.

(b) Outro exemplo de medidas obtidas com o aplicativo desenvolvido.



Fonte – Acervo da autora.

5 Resultados e discussão

Neste capítulo são apresentados os resultados obtidos através da metodologia proposta e a métrica utilizada para avaliação desses resultados.

5.1 Coeficiente de Correlação Intraclasse

Uma das métricas mais utilizadas em Estatística para avaliação de confiabilidade de medidas é o Coeficiente de Correlação Intraclasse (ICC), do inglês *Interclass Correlation Coefficient*. O ICC é utilizado para mensurar a homogeneidade de duas ou mais medidas, sendo visto como a medida da proporção da variabilidade total atribuída ao objeto medido (LAUREANO, 2011). Geralmente, o ICC é aplicado em métodos de estimação de medidas, a fim de comparar a medida estimada com a medida real.

A definição de ICC apresentada por Bergamaschi (1999) diz que ao considerar a existência de dois grupos de fatores que causam variação, a correlação intraclasse quantifica a importância relativa de um dos fatores. O ICC é obtido através da divisão da variância entre o indivíduos, σ_x^2 , pela variância total, σ_T^2 , que inclui a variação entre indivíduos e a variação não pretendida σ_ε^2 , ou erro, conforme mostra a fórmula 5.1.

$$\rho = \frac{\sigma_x^2}{\sigma_x^2 + \sigma_\varepsilon^2} = \frac{\sigma_x^2}{\sigma_T^2} \quad (5.1)$$

O ICC tem como resultado um valor entre 0 a 1, que representa a confiabilidade das medidas observadas. Se o ICC for igual a 0, há grande variabilidade intra-observador mas não há variabilidade inter-observador, o que significa que os grupos de medidas não são homogêneos ou confiáveis. Se o valor do ICC for 1, não há variabilidade intra-observador mas há uma grande variabilidade inter-observador, ou seja, as medidas obtidas são altamente confiáveis e homogêneas entre si. Para os valores entre esse intervalo, a tabela apresentada por Pinto et al. (2017) deve ser consultada:

Tabela 1 – Interpretação dos valores do ICC

Valores do ICC	Interpretação
ICC < 0.4	Pobre
$0.4 \leq ICC < 0.75$	Satisfatório
ICC ≥ 0.75	Excelente

Neste trabalho, as medidas obtidas pelo aplicativo desenvolvido são representadas pela variância entre os indivíduos - σ_x^2 , e as medidas obtidas por métodos manuais, ou as medidas reais são representadas pela variância total, σ_T^2 .

5.2 Testes e Resultados

A fim de avaliar as medidas obtidas com o aplicativo desenvolvido, três tipos de testes foram verificados, levando em conta a especificação de que o dispositivo foi feito para operar em distâncias de 40 cm a 4 metros. Sendo assim, verificou-se a precisão de medidas de até 1 metros, até 2 metros e até 3 metros, a fim de determinar se o aumento da distância prejudica a precisão. Para cada grupo, dez mensurações foram feitas a partir de elementos do Núcleo de Computação Aplicada, como paredes, piso e móveis. Cada elemento foi medido três vezes, uma com uma trena comum, outra com uma trena a *laser* e outra com a aplicação desenvolvida. A trena comum e a trena a *laser* foram usadas como forma de comparação por serem os métodos mais utilizados por profissionais de AEC e por usuários comuns. As tabelas 2, 3 e 4 mostram as medidas obtidas com cada método e o erro entre cada grupo de medida. T_c representa as medidas obtidas com a trena comum, T_l com a trena a *laser* e M_p representa as medidas obtidas com o método proposto.

Tabela 2 – Medidas de até 1 metro.

(T_c)	(T_l)	(M_p)	$T_c - T_l$	$T_c - M_p$	$T_l - M_p$
33.4 cm	33.5 cm	32.5 cm	-1 mm	9 mm	1 cm
40 cm	40.1 cm	40.4 cm	-1 mm	-4 mm	-3 mm
55.9 cm	56.1 cm	52.5 cm	-2 mm	3.4 cm	3.6 cm
59 cm	58.8 cm	55.6 cm	2 mm	3.4 cm	3.2 cm
60 cm	59.6 cm	63.8 cm	4 mm	-3.8 cm	-4.2 cm
72.6 cm	72.2 cm	70.9 cm	4 mm	1.7 cm	1.3 cm
80 cm	79.9 cm	80.9 cm	1 mm	-9 mm	-1 cm
82.5 cm	83.2 cm	82.3 cm	-7 mm	2 mm	9 mm
1 m	99.8 cm	96 cm	2 mm	4 cm	3.8 cm
1 m	1.003 m	98 cm	-3 mm	2 cm	2.3 cm

Tabela 3 – Medidas de até 2 metros.

(T_c)	(T_l)	(M_p)	$T_c - T_l$	$T_c - M_p$	$T_l - M_p$
1.055 m	1.06 m	1.078 m	-5 mm	-2.3 cm	-1.8 cm
1.101 m	1.10 m	1.111 m	1 mm	-1 cm	-1.1 cm
1.17 m	1.168 m	1.178 m	2 mm	-8 mm	-1 cm
1.20 m	1.189 m	1.224 m	1.1 cm	-2.4 cm	-3.5 cm
1.37 m	1.367 m	1.371 m	3 mm	-1 mm	-4 mm
1.413 m	1.41 m	1.38 m	3 mm	3.3 cm	3 cm
1.54 m	1.538 m	1.557 m	2 mm	-1.7 cm	-1.9 cm
1.565 m	1.56 m	1.54 m	5 mm	2.5 cm	2 cm
1.92 m	1.93 m	1.954 m	-1 cm	-3.4 cm	-2.4 cm
2 m	2.001 m	2.012 m	-1 mm	-1.2 cm	-1.1 cm

A fim de verificar se o método proposto para cálculo de área apresentou resultados consistentes, alguns polígonos de área definida foram representados na aplicação através

Tabela 4 – Medidas de até 3 metros.

(T_c)	(T_l)	(M_p)	$T_c - T_l$	$T_c - M_p$	$T_l - M_p$
2.11 m	2.107 m	2.109 m	3 mm	1 mm	-2 mm
2.11 m	2.109 m	2.101 m	1 mm	9 mm	8 mm
2.16 m	2.152 m	2.125 m	8 mm	3.5 cm	2.7 cm
2.28 m	2.28 m	2.30 m	0	-2 cm	-2 cm
2.30 m	2.307 m	2.300 m	-7 mm	0	7 mm
2.53 m	2.529 m	2.534 m	1 mm	-4 mm	-5 mm
2.63 m	2.631 m	2.74 m	-1 mm	-11 cm	-10.9 cm
2.75 m	2.749 m	2.77 m	1 mm	-2 cm	-2.1 cm
2.80 m	2.80 m	2.812 m	0	-1.2 cm	-1.2 cm
2.985 m	2.996 m	2.96 m	-1.1 cm	-2.5 cm	3.6 cm

de pontos de interesse e o resultado da área fornecida pelo aplicativo foi comparado com o resultado obtido pelas fórmulas abaixo. As Figuras 16a e 16b, mostram dois exemplos de polígonos, um trapézio e um hexágono. A fórmula do trapézio é dada por 5.2 e a do hexágono é dada por 5.3:

$$A_{\text{Trapézio}} = \frac{B + b}{2} \times h \quad (5.2)$$

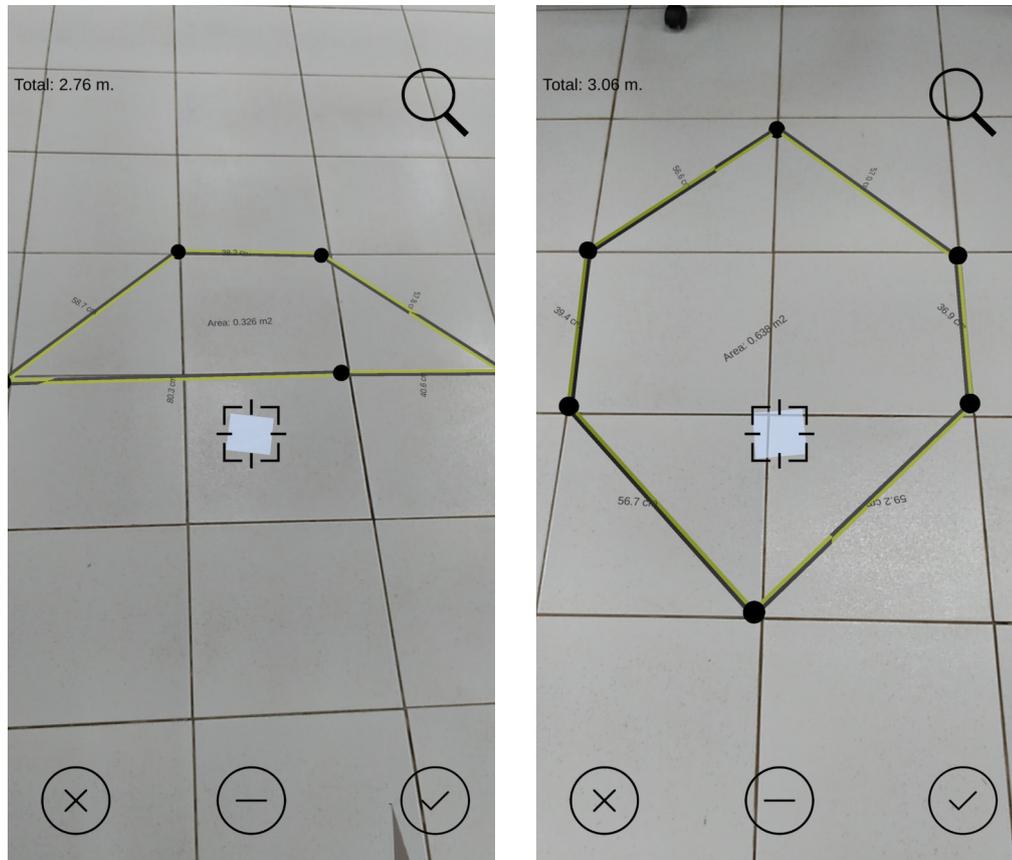
$$A_{\text{Hexágono}} = \frac{\text{Perímetro} \times \text{Apótema}}{2} \quad (5.3)$$

Usando as medidas apresentadas na Figura 16a, B representa a maior base do trapézio e mede 1.20 metros, b representa a menor base do trapézio e mede 38.2 centímetros, a altura h não está representada na figura, mas pode ser obtida a partir das outras arestas, apresentando valor aproximado de 41.2 centímetros. Aplicando a fórmula 5.2, a área do trapézio é de $0.327m^2$ metros quadrados, bem semelhante ao valor obtido pela metodologia proposta, de $0.326m^2$. Já a Figura 16b, mostra um hexágono de perímetro 3.06 metros e apótema de aproximadamente 41 centímetros, com área total de $0.627m^2$, também semelhante à área de $0.638m^2$ apresentada na Figura 16b.

5.3 Discussões

Os valores obtidos em cada grupo foram usados para calcular o ICC, enquanto que os erros entre cada grupo de medidas foram usados para calcular o Erro Médio Quadrático, do inglês *Mean Squared Error* (MSE), dado pela fórmula 5.4, onde n representa o número total de medidas, M representa o primeiro conjunto de medidas e N , o segundo. Enquanto o ICC determina a confiabilidade das medidas obtidas com o método proposto, o MSE fornece uma noção melhor da diferença entre a medida real e a medida estimada. As

Figura 16 – Obtenção de área com a ferramenta desenvolvida.

(a) Trapézio de área $0.326m^2$.(b) Hexágono de área $0.638m^2$.

Fonte – Acervo da autora.

tabelas 5 e 6 apresentam os resultados desses testes.

$$MSE = \frac{1}{n} \sum_{i=1}^n (M_i - N_i)^2 \quad (5.4)$$

Tabela 5 – ICC de cada grupo de medidas.

ICC	M_p e T_c	M_p e T_l	T_c e T_l
1 metro	0.99466	0.99441	0.99987
2 metros	0.99792	0.99822	0.99988
3 metros	0.99307	0.99279	0.99990

O ICC mostra que as medidas obtidas com o aplicativo apresentam alta confiabilidade, tanto em comparação com as medidas obtidas com a trena comum quanto com a trena a *laser*. Para todos os grupos de medida, o ICC obtido foi superior a 99%, sendo que o grupo de medidas de até 2 metros apresentou a maior confiabilidade. Apesar disso, com o MSE percebe-se a real diferença entre as medidas obtidas com o método proposto e as medidas

Tabela 6 – MSE de cada grupo de medidas.

MSE	M_p e T_c	M_p e T_l	T_c e T_l
1 metro	6.227 cm	6.516 cm	0.105 cm
2 metros	4.573 cm	4.144cm	0.299 cm
3 metros	14.992 cm	15.033 cm	0.247 cm

obtidas com os métodos usuais. No grupo de medidas de até 1 metro, o erro máximo foi de -4.2 cm, enquanto que o MSE foi de 6.2 cm para trena comum e 6.5 em comparação com a trena a *laser*. As medidas de até 2 metros apresentaram os melhores resultados dentre os três grupos, com erro máximo de 3.5 cm e MSE de 4.57 cm para trena comum e 4.1 cm para trena a *laser*. Quanto ao grupo de medidas de até 3 metros, aconteceu um caso que destoa de todas as outras medidas obtidas. Uma das medidas apresentou um erro de -11 cm, enquanto que dentre as outras 9 medidas, 3.6 cm foi o erro máximo. Por apresentar um resultado tão discrepante, verificou-se a distância do *smartphone* para o alvo analisado, que foi maior que 3.7 metros, ou seja, bem próximo do limite alcançado pela câmera de profundidade. Executou-se a medida novamente, dessa vez a uma distância mais próxima do alvo, e o valor apresentado foi bem próximo do valor real. Acredita-se que esse erro aconteceu porque o usuário estava muito longe do ponto de interesse; em casos como esse, recomenda-se se aproximar um pouco mais do alvo. A Tabela 7 mostra que com exceção desse caso, a MSE de medidas de até 3 metros foi de 3.89 cm para trena comum e 3.1 em para trena a *laser*, analisando todas as medidas do grupo, o MSE foi de 14.9 cm e 15 cm, respectivamente.

Tabela 7 – MSE do terceiro grupo de medidas sem o pior caso.

MSE	M_p e T_c	M_p e T_l	T_c e T_l
3 metros	2.89 cm	3.152 cm	0.247 cm

Com exceção desse caso discordante, as medidas de até 1 metro apresentaram a maior diferença entre os grupos de medidas, mas ainda assim, as medidas obtidas com o método proposto se aproximam bastante das medidas reais. Com os resultados obtidos, sugere-se que o usuário use o aplicativo desenvolvido para obter estimativas de medidas e para fazer inspeções, a fim de conferir se a área construída corresponde ou se aproxima da área planejada, já que essa é uma atividade que não requer precisão extrema. A aplicação também é apropriada em casos de verificação de área construída para efetuar pagamentos, pois nessa situação, o pagamento do funcionário é determinado pela quantidade de metros quadrados construídos, e diferenças de 4 a 6 centímetros não vão causar tanta interferência. A aplicação é adequada principalmente quando se quer ter uma noção das medidas de um elemento, entretanto, na fase de planejamento de ambiente, sugere-se utilizar os métodos tradicionais, já que a precisão extrema é indispensável.

6 Conclusão

Este trabalho apresentou um método para automatização da obtenção de medidas de superfícies e cálculo de áreas. O método proposto utilizou a plataforma Tango em conjunto com conceitos de realidade aumentada e técnicas de Computação Gráfica para o desenvolvimento de uma ferramenta para dispositivos móveis em que o usuário define os pontos referentes à área que deseja analisar pela tela de seu *smartphone*, eliminando o contato direto com meio físico, o que poupa tempo, esforço e pode ser útil em ambientes insalubres. A ferramenta desenvolvida visa auxiliar usuários na obtenção de dados *As-Built*, que podem ser usados para verificação de área construída, em inspeções, na obtenção de dados para construção de registros *As-Built*, ou simplesmente para obtenção de informações.

A fim de determinar a viabilidade do uso da aplicação desenvolvida, foi feita uma análise dos dados obtidos pela ferramenta. O resultado do teste de ICC mostrou que as medidas obtidas pela ferramenta desenvolvida são comparáveis às obtidas pelos métodos usuais, com confiabilidade de 99%. Entretanto, os resultados obtidos com a MSE para medidas de até 1 metro mostraram que em média, a diferença da medida obtida com o método proposto pode chegar até 6cm em relação à medida real, enquanto que para medidas de até 2 metros pode chegar até 4cm e para medidas de até 3 metros, pode variar de 3cm até 15cm nos piores casos. Tais resultados limitam o uso da aplicação em atividades que exigem extrema precisão.

Como trabalhos futuros, sugere-se incluir o cálculo de volumes e fazer a reconstrução 3D do ambiente analisado para que o arquivo gerado possa ser usado como Building Information Modeling (BIM). Por fim, seria de grande utilidade fazer um estudo da usabilidade do aplicativo com profissionais das áreas de AEC, a fim de efetuar melhorias na aplicação.

Referências

- AMANN, M.-C.; BOSCH, T.; LESCURE, M.; MYLLYLÄ, R.; RIOUX, M. Laser ranging: a critical review of usual techniques for distance measurement. *Optical engineering*, International Society for Optics and Photonics, v. 40, n. 1, p. 10–19, 2001. Citado na página 14.
- ANGLADA, M. V. An improved incremental algorithm for constructing restricted delaunay triangulations. *Computers & Graphics*, Elsevier, v. 21, n. 2, p. 215–223, 1997. Citado na página 24.
- APPLE. *Introducing ARKit*. 2017. Disponível em: <<https://developer.apple.com/arkit/>>. Acessado em 27 de Dezembro de 2017. Citado na página 17.
- AZUMA, R.; BAILLOT, Y.; BEHRINGER, R.; FEINER, S.; JULIER, S.; MACINTYRE, B. Recent advances in augmented reality. *IEEE computer graphics and applications*, IEEE, v. 21, n. 6, p. 34–47, 2001. Citado 3 vezes nas páginas 13, 19 e 20.
- AZUMA, R. T. A survey of augmented reality. *Presence: Teleoperators and virtual environments*, MIT Press, v. 6, n. 4, p. 355–385, 1997. Citado 3 vezes nas páginas 13, 19 e 20.
- BAE, H.; GOLPARVAR-FARD, M.; WHITE, J. High-precision vision-based mobile augmented reality system for context-aware architectural, engineering, construction and facility management (aec/fm) applications. *Visualization in Engineering*, Springer International Publishing, v. 1, n. 1, p. 3, 2013. Citado na página 13.
- BERGAMASCHI, D. P. *Correlação intraclasse de Pearson para pares repetidos: comparação entre dois estimadores*. Tese (Doutorado) — Universidade de São Paulo, 1999. Citado na página 39.
- BERGER, M.; TAGLIASACCHI, A.; SEVERSKY, L. M.; ALLIEZ, P.; GUENNEBAUD, G.; LEVINE, J. A.; SHARF, A.; SILVA, C. T. A survey of surface reconstruction from point clouds. In: WILEY ONLINE LIBRARY. *Computer Graphics Forum*. [S.l.], 2017. v. 36, n. 1, p. 301–329. Citado na página 21.
- CARDOSO, A.; KIRNER, C.; JÚNIOR, E. L.; KELNER, J. Tecnologias e ferramentas para o desenvolvimento de sistemas de realidade virtual e aumentada. *Editora Universitária UFPE*, p. 1–19, 2007. Citado na página 19.
- CHENG, H. *Behind Our Phab 2 Pro, the World's 1st Tango-Enabled Smartphone*. 2016. Disponível em: <<http://blog.lenovo.com/en/blog/behind-the-lenovo-phab-2-pro-the-worlds-first-tango-enabled-smartphone>>. Acessado em 27 de Dezembro de 2017. Citado 2 vezes nas páginas 24 e 29.
- CHI, H.-L.; KANG, S.-C.; WANG, X. Research trends and opportunities of augmented reality applications in architecture, engineering, and construction. *Automation in construction*, Elsevier, v. 33, p. 116–122, 2013. Citado na página 13.

- CIGNONI, P.; MONTANI, C.; SCOPIGNO, R. Dwall: A fast divide and conquer delaunay triangulation algorithm in ed. *Computer-Aided Design*, Elsevier, v. 30, n. 5, p. 333–341, 1998. Citado na página 24.
- FROEHLICH, M.; AZHAR, S.; VANTURE, M. An investigation of google tango® tablet for low cost 3d scanning. In: VILNIUS GEDIMINAS TECHNICAL UNIVERSITY, DEPARTMENT OF CONSTRUCTION ECONOMICS & PROPERTY. *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*. [S.l.], 2017. v. 34. Citado na página 18.
- GOLPARVAR-FARD, M.; PEÑA-MORA, F.; ARBOLEDA, C. A.; LEE, S. Visualization of construction progress monitoring with 4d simulation model overlaid on time-lapsed photographs. *Journal of Computing in Civil Engineering*, American Society of Civil Engineers, v. 23, n. 6, p. 391–404, 2009. Citado na página 14.
- Gomes Jr., D. L. *Metodologia para construção de aplicações de RV e RA com marcadores naturais em cenários industriais*. Tese (Doutorado) — Universidade Federal do Maranhão, 2017. Citado 3 vezes nas páginas 13, 14 e 20.
- GOOGLE. *Tango: Start Here*. 2015. Disponível em: <<https://developers.google.com/tango/>>. Acessado em 27 de Dezembro de 2017. Citado 4 vezes nas páginas 14, 15, 16 e 17.
- GOOGLE. *Measure*. 2017. Disponível em: <<https://play.google.com/store/apps/details?id=com.google.tango.measure&hl=en>>. Acessado em 27 de Dezembro de 2017. Citado 2 vezes nas páginas 17 e 18.
- GOOGLE. *Tango Concepts*. 2017. Disponível em: <<https://developers.google.com/tango/overview/concepts>>. Acessado em 18 de Dezembro de 2017. Citado na página 21.
- GUIBAS, L. J.; KNUTH, D. E.; SHARIR, M. Randomized incremental construction of delaunay and voronoi diagrams. *Algorithmica*, Springer, v. 7, n. 1, p. 381–413, 1992. Citado na página 33.
- GÄRTNER, B.; HOFFMANN, M. *Computational Geometry Lecture Notes*. 2013. Disponível em: <<https://www.ti.inf.ethz.ch/ew/Lehre/CG13/lecture/cg-2013.pdf>>. Acessado em 29 de Dezembro de 2017. Citado 3 vezes nas páginas 24, 25 e 26.
- KOCH, C.; NEGES, M.; KÖNIG, M.; ABRAMOVICI, M. Natural markers for augmented reality-based indoor navigation and facility maintenance. *Automation in Construction*, Elsevier, v. 48, p. 18–30, 2014. Citado na página 20.
- LABS, L. *AirMeasure - AR Tape Ruler*. 2017. Disponível em: <<https://itunes.apple.com/us/app/airmeasure-ar/id1251282152?ls=1&mt=8>>. Acessado em 27 de Dezembro de 2017. Citado 2 vezes nas páginas 17 e 18.
- LAUREANO, G. H. d. C. Coeficiente de correlação intraclasse: comparação entre métodos de estimação clássico e bayesianos. 2011. Citado na página 39.
- LÁZARO, J. L.; CANO, A. E.; FERNÁNDEZ, P. R.; POMPA, Y. Sensor for distance measurement using pixel grey-level information. *Sensors*, Molecular Diversity Preservation International, v. 9, n. 11, p. 8896–8906, 2009. Citado na página 14.

- LENOVO. *Lenovo Phab 2 Pro Features and Tech Specs*. 2017. Disponível em: <<https://www3.lenovo.com/us/en/smart-devices/-lenovo-smartphones/phab-series/Lenovo-Phab-2-Pro/p/WMD00000220>>. Acessado em 19 de Dezembro de 2017. Citado na página 29.
- LINSEN, L. *Point cloud representation*. [S.l.]: Univ., Fak. für Informatik, Bibliothek, 2001. Citado na página 21.
- LIU, X.; EYBPOOSH, M.; AKINCI, B. Developing as-built building information model using construction process history captured by a laser scanner and a camera. In: *Construction Research Congress 2012: Construction Challenges in a Flat World*. [S.l.: s.n.], 2012. p. 1232–1241. Citado 2 vezes nas páginas 17 e 18.
- MILGRAM, P.; KISHINO, F. A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems*, The Institute of Electronics, Information and Communication Engineers, v. 77, n. 12, p. 1321–1329, 1994. Citado na página 19.
- MITSUSE, B. *Levantamento de Imóvel - As Built*. 2014. Disponível em: <<https://ateliereformawordpress.com/2014/07/11/levantamento-de-imovel-as-built/>>. Acessado em 09 de Janeiro de 2018. Citado na página 28.
- NISTÉR, D.; NARODITSKY, O.; BERGEN, J. Visual odometry. In: IEEE. *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*. [S.l.], 2004. v. 1, p. I–I. Citado na página 22.
- NOGUEIRA, J. R. Uma revisão da triangulação de delaunay com malhas geradas pelo algoritmo de green-sibson. 2013. Citado na página 26.
- PINHAL. *Terminologias Arquitetônicas*. 2014. Disponível em: <<http://www.colegiodearquitetos.com.br/dicionario/2014/03/as-built/>>. Acessado em 17 de Outubro de 2017. Citado 2 vezes nas páginas 15 e 27.
- PINTO, J. S.; LOPES, J. M.; OLIVEIRA, J. V. d.; AMARO, J. P.; COSTA, L. D. d. *Métodos para Estimção de Reprodutividade de Medidas : Coeficiente de correlação Intraclasse*. 2017. Disponível em: <<https://users.med.up.pt/~joakim/intromed/coeficientecorrelacaointraclasse.htm>>. Acessado em 03 de Janeiro de 2018. Citado na página 39.
- PITERI, M. A.; JUNIOR, M. M. Triangulação de delaunay e o princípio de inserção randomizado. *II Simpósio Brasileiro de Geomática-V Colóquio Brasileiro de Ciências Geodésicas, Presidente Prudente-SP*, p. 9, 2007. Citado na página 25.
- ROSEBROCK, A. *Find distance from camera to object/marker using Python and OpenCV*. 2015. Disponível em: <<https://www.pyimagesearch.com/2015/01/19/find-distance-camera-objectmarker-using-python-opencv/>>. Acessado em 27 de Dezembro de 2017. Citado 2 vezes nas páginas 17 e 18.
- RUSU, R. B.; COUSINS, S. 3D is here: Point Cloud Library (PCL). In: *IEEE International Conference on Robotics and Automation (ICRA)*. Shanghai, China: [s.n.], 2011. Citado na página 21.
- SERON, S.; OLIVEIRA, D. S. de; SANTOS, A. A. Modulação automática de edifícios em alvenaria estrutural com a utilização da plataforma cad: Programa alvmod. 2009. Citado na página 15.

- TANGO, P. *Tango Point Cloud*. 2016. Disponível em: <https://developers.google.com/tango/apis/c/reference/struct/tango-point-cloud#struct_tango_point_cloud>. Acessado em 19 de Dezembro de 2017. Citado na página 24.
- TANGO, P. *Depth Perception*. 2017. Disponível em: <<https://developers.google.com/tango/overview/depth-perception>>. Acessado em 18 de Dezembro de 2017. Citado 4 vezes nas páginas 23, 24, 25 e 30.
- TANGO, P. *Motion Tracking*. 2017. Disponível em: <<https://developers.google.com/tango/overview/motion-tracking>>. Acessado em 19 de Dezembro de 2017. Citado 3 vezes nas páginas 22, 23 e 30.
- VECTOR, G. D. *As Built na Arquitetura com topografia e geodésia*. 2017. Disponível em: <<http://www.vector.agr.br/produto/as-built-na-arquitetura-com-topografia-e-geodesia/>>. Acessado em 09 de Janeiro de 2018. Citado na página 27.
- VELAZCO, C. *Lenovo Phab 2 Pro review: Stumbling out of the gate*. 2016. Disponível em: <<https://www.engadget.com/2016/11/23/lenovo-phab-2-pro-review/>>. Acessado em 19 de Dezembro de 2017. Citado na página 24.
- WANG, X.; TRUIJENS, M.; HOU, L.; WANG, Y.; ZHOU, Y. Integrating augmented reality with building information modeling: Onsite construction process controlling for liquefied natural gas industry. *Automation in Construction*, Elsevier, v. 40, p. 96–105, 2014. Citado na página 13.
- WEISSTEIN, E. W. *Heron's formula*. 2003. Disponível em: <<http://mathworld.wolfram.com/HeronsFormula.html>>. Acessado em 06 de Janeiro de 2018. Citado na página 34.
- WOSOMTECH. *Easy Build*. 2017. Disponível em: <<https://play.google.com/store/apps/details?id=com.Wosomtech.EasyBuild>>. Acessado em 27 de Dezembro de 2017. Citado na página 18.