

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
COORDENAÇÃO DE ENGENHARIA ELETRÔNICA
CURSO DE ENGENHARIA ELETRÔNICA

GERMANO AUGUSTO WINSCH
PEDRO HENRIQUE SARAIVA SANTOS

**DESENVOLVIMENTO DE UM ESCÂNER
TRIDIMENSIONAL POR TRIANGULAÇÃO BASEADO
EM SISTEMA ÓTICO E FEIXE DE LASER**

TRABALHO DE CONCLUSÃO DE CURSO

TOLEDO
FEVEREIRO - 2014

GERMANO AUGUSTO WINSCH
PEDRO HENRIQUE SARAIVA SANTOS

DESENVOLVIMENTO DE UM ESCÂNER TRIDIMENSIONAL POR TRIANGULAÇÃO BASEADO EM SISTEMA ÓTICO E FEIXE DE LASER

Trabalho de conclusão de curso de graduação apresentado a disciplina de Trabalho de Conclusão de Curso II, do Curso Superior de Engenharia Eletrônica - COELE - da Universidade Tecnológica Federal do Paraná Campus Toledo como requisito parcial para obtenção do título de Bacharel.

Orientador: Prof. Dr. Paulo de Tarso Neves Jr.

TOLEDO
FEVEREIRO - 2014



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Toledo
Coordenação do Curso de Engenharia Eletrônica



TERMO DE APROVAÇÃO

Título do Trabalho de Conclusão de Curso N° 002

Desenvolvimento de um escâner tridimensional baseado em sistema ótico e feixe de laser

por

Germano Augusto Winsch
Pedro Henrique Saraiva Santos

Esse Trabalho de Conclusão de Curso foi apresentado às 8:20h do dia **21 de fevereiro de 2014** como requisito parcial para a obtenção do título **Bacharel em Engenharia Eletrônica**. Após deliberação da Banca Examinadora, composta pelos professores abaixo assinados, o trabalho foi considerado **APROVADO**.

Prof. Dr. Alberto Yoshihiro Nakano
(UTFPR-TD)

Prof. M. José Dolores Vergara Dietrich
(UTFPR-TD)

Prof. Dr. Paulo de Tarso Neves Junior
(UTFPR-TD)
Orientador

Visto da Coordenação

Prof. M. Rodrigo da Ponte Caun
Coordenador da COELE

AGRADECIMENTOS

Primeiramente, gostaríamos de agradecer às nossas famílias que sempre estiveram ao nosso lado, dando-nos apoio e conforto durante toda essa caminhada sempre compreendendo nossos momentos de ausências.

Às nossas namoradas, Camila e Ana Claudia, pelo apoio constante e por toda ajuda e paciência, principalmente nos finais de semestre.

À empresa parceira, Microbras®, em especial ao Maureu J. Benvenuto que sempre esteve disposto, auxiliando e viabilizando a execução do projeto.

Aos nossos amigos e colegas de turma que estiveram conosco todo este tempo.

“A mente que se abre a uma nova ideia jamais voltará ao seu tamanho original” (Albert Einstein).

RESUMO

SANTOS, Pedro H. S. e WINSCH, Germano A. Desenvolvimento de um escâner tridimensional baseado em sistema ótico e feixe de laser. 2014. 65 f. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Eletrônica), Universidade Tecnológica Federal do Paraná. Toledo, 2014.

Este trabalho apresenta os conceitos e métodos para o desenvolvimento e funcionamento de um escâner tridimensional. O escâner é capaz de digitalizar informações referentes a um modelo real, e gera um arquivo de nuvem de pontos, passível de alterações por softwares específicos. Todos os aspectos construtivos como eletrônica, mecânica e software são abordados. Através da utilização da triangulação por câmera e feixe de laser, busca-se um produto capaz de competir com similares de mercado, a fim de atender uma necessidade da empresa parceira do projeto.

Palavras-chave: Triangulação. Escâner 3D. Digitalização.

ABSTRACT

SANTOS, Pedro H. S. e WINSCH, Germano A. Development of a 3D scanner based on optic system and laser beam. 2014. 65 f. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Eletrônica), Federal University of Technology - Paraná. Toledo, 2014

This study presents the methods and concepts to the development and operation of a 3D scanner. This scanner is capable of digitalizing information from a real model, generating a point cloud file, which can be modified by some specific softwares. All the constructive aspects such as electronic, mechanics and software are discussed. By using triangulation with laser and camera, this project seeks to create a product capable of competing with similar ones in the market, in order to fulfill a need from the partner company.

Keywords: Triangulation. Scanner 3D. Digitalizer.

LISTA DE SÍMBOLOS

(X, Y, Z)	Coordenadas tridimensionais não homogêneas
(X_1, X_2, X_3, X_4)	Coordenadas tridimensionais homogêneas
(x, y)	Coordenadas no sistema de imagem
f	Comprimento focal
f_x, f_y	Comprimentos focais horizontal e vertical
λ	Fator de escala homogêneo da coordenada
(o_x, o_y)	Centro da imagem ou ponto principal
\mathbf{P}	$(X, Y, Z)^T$
\mathbf{M}	Matriz dos parâmetros intrínsecos da câmera
r	Raio com relação ao centro da imagem
a_0, a_1, a_2	Coeficientes do polinômio derivado da série de Taylor
k_1, k_2, k_3	Coeficientes do polinômio para correção da distorção radial
p_1, p_2	Coeficientes da equação de correção da distorção tangencial
P_c	Ponto no sistema de coordenadas da câmera (homogênea)
p_c	Ponto no sistema de coordenadas da câmera (não-homogênea)
P_0	Ponto no sistema de coordenadas espaciais
\mathbf{R}	Matriz de rotação
\mathbf{t}	Vetor de translocação

LISTA DE ABREVIATURA

ARM	Advanced RISC Machine
CAD	<i>Computer Aided Design</i> (Desenho auxiliado por computador)
CAE	<i>Computer Aided Engineering</i> (Engenharia auxiliada por computador)
CAM	<i>Computer Aided Manufacturing</i> (Fabricação auxiliada por computador)
CT	<i>Computed Tomography</i>
CMOS	Complementary metal-oxide-semiconductor
CPU	Central Processing Units
CSI	Camera Serial Interface
DSI	Display Serial Interface
GPIO	General Purpose Input/Output
GUI	Graphical User Interface
ISO	International Organization for Standardization
IDE	Integrated Development Environment
I2C	Inter-Integrated Circuit
LCD	Liquid Crystal Display
PLA	Ácido Polilático
PC	Personal Computer
SoC	System on Chip
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver/Transmitter
XML	Extensible Markup Language

LISTA DE FIGURAS

Figura 1. Partes integrantes do sistema	16
Figura 2. Princípio da Triangulação. Com laser e câmera Fonte: Bernardini e Rushmeier, 2002.....	19
Figura 3 - Dois trilhos paralelos se encontram no ponto em que desaparecem no horizonte Fonte: Morvan, 2009.....	21
Figura 4. Modelo básico de câmera Pin Hole. Fonte: Adaptado de Bradski e Kaehler, (2008).....	24
Figura 5. Descrição da relação entre um ponto 3D (X,Y,Z) e sua correspondente projeção (u,v) no plano de imagem. Fonte: Adaptado de Morvan, (2009).....	25
Figura 6. Distorção radial aplicada a um mapa de pixels. (a) Imagem sem distorção (b) Imagem com distorção Fonte: Morvan, (2009).....	27
Figura 7. (a) Imagem com distorção radial (b) Imagem com distorção corrigida através das equações x. Fonte: Morvan, (2009).....	28
Figura 8. Ilustração de distorção Tangencial Fonte: Adaptado de Bradski e Kaehler, (2008).	28
Figura 9. Transformação de coordenadas Fonte: Bradski e Kaehler, (2008).....	30
Figura 10. Determinação do ponto de intersecção entre plano e vetor. Fonte: SUNDAY	32
Figura 11. Imagem do projeto do protótipo.	35
Figura 12. Projeto do suporte da câmera.	36
Figura 13. Projeto do suporte do laser.	36
Figura 14. Projeto das chapas metálicas.	37
Figura 15. Comparação dos dois feixes de laser. Lado direito laser grosso e lado esquerdo laser fino.	37
Figura 16. Projeto da mesa giratória.	38
Figura 17. Projeto da engrenagem de fixação na mesa.	38
Figura 18. Engrenagem do motor.	39
Figura 19. Projeto do eixo.	39
Figura 20. Modelo do Raspberry Pi. Fonte: Adaptado de < http://www.raspberrypi.org/faqs >.	41
Figura 21. Esquemático das GPIO's do Raspberry Pi. Fonte: Esquemático do Raspberry Pi.	42
Figura 22. Câmera RaspiCam utilizada.	44
Figura 23. Esquemático da placa auxiliar.	46
Figura 24. Comportamento da corrente na bobina do motor para uma configuração de 1/8 micro-passos. Fonte: Datasheet alegre A44988	49
Figura 25. Imagens para o processo de calibração.....	55
Figura 27. Objeto padrão de calibração.....	57
Figura 28. Imagem da interface do software.	58
Figura 31. Comparação dos espaços de cores.....	62
Figura 32. Aplicação dos filtros na Figura 30.....	63
Figura 44. Objeto 4. a) Vista lateral. b) Vista posterior. c) Vista Frontal.....	75
Figura 45. a) Nuvem de pontos do objeto 4. b) Vista lateral da malha 4. c) Vista posterior da malha 4.....	75
Figura 48. Possível adição de um laser.....	76

Figura 51. Laser não ortogonal ao plano da mesa.	80
Figura 52. Exemplo de estreitamento no topo.....	80
Figura 53. Vortex da Figura 52.....	81
Figura 54. Medida do diâmetro do Objeto 3.....	82
Figura 55. Medida da base do Objeto 2.....	82

SUMÁRIO

1	INTRODUÇÃO	12
1.1	OBJETIVOS DO TRABALHO	14
1.2	VISÃO GERAL	15
1.3	ESTRUTURA DO TRABALHO	16
2	SELEÇÃO DO MÉTODO DE ESCANEAMENTO	17
2.1	MÉTODOS DE ESCANEAMENTO	17
2.2	SELEÇÃO DO MÉTODO UTILIZADO	18
2.3	<i>SLIT SCANNER</i>	19
3	TEORIA E CONCEITOS	21
3.1	GEOMETRIA PROJETIVA	21
3.2.1	Modelo da Câmera Pinhole	23
3.2.2	Distorções na Câmera	26
3.2.3	Calibração e Obtenção de Parâmetros Intrínsecos	29
3.2.4	Parâmetros Extrínsecos	30
3.2.5	Modelo Completo do Sistema Ótico	31
3.3	INTERSECÇÃO ENTRE LINHAS E PLANOS	32
4	MATERIAIS E MÉTODOS	34
4.1	PROJETO MECÂNICO	34
4.2	SISTEMA EMBARCADO / ELETRÔNICA	40
4.2.1	Raspberry Pi	40
4.2.2	RaspCam	43
4.2.3	Placa Eletrônica Auxiliar	45
4.3	SOFTWARE	50
4.3.1	Linguagem, Compiladores e IDE's	51
4.3.2	WiringPi	52
4.3.3	Comunicação Entre Softwares	52
4.3.4	Software no Raspberry Pi	53
4.3.5	OpenCv	54
4.3.8.1	Interface Visual	58
4.3.8.2	Acesso às Imagens	61
4.3.8.3	Filtro e Processamento	61
4.3.8.4	Triangulação	64
4.3.8.5	Arquivo Gerado	66

5	RESULTADOS	67
5.1	PROTÓTIPO	67
5.2	CALIBRAÇÃO	69
5.3.1	Posicionamento do Laser	73
5.3.2	FAIXA DE DETECÇÃO DO FILTRO	77
5.4	ANÁLISES DE ERROS	78
5.4.1	Plano do Laser	78
5.4.2	DIMENSIONAL	81
6.	CONCLUSÃO	85
7.	REFERÊNCIAS	86
8.	ANEXO I	89

1 INTRODUÇÃO

O uso e acesso às tecnologias de imagens digitais se tornou algo comum, principalmente com a popularização das câmeras fotográficas digitais, com tamanho e preços reduzidos e mídias com capacidade para armazenar todas estas informações. No entanto, com internet de alta velocidade e computadores com grande capacidade de processamento, surge uma nova tendência para visualização de objetos: a visualização tridimensional.

Lojas virtuais com modelos em três dimensões estão se tornando comuns. Fabricantes, em todas as áreas, estão disponibilizando desenhos de seus produtos em formatos 3D, facilitando e atendendo necessidades diversas dos consumidores.

Em geral, a criação de modelos tridimensionais a partir de objetos é feita através de sistemas CAD (Computer Aided Design)/CAM (Computer Aided Engineering)/CAE (Computer Aided Manufacturing) (engenharia reversa), porém, quando existe uma morfologia complexa no modelo, a criação através destes sistemas se torna complexa e é difícil obter uma representação fiel do modelo real.

Em meio a esse mercado, o escâner 3D surge como facilitador na transferência de modelos reais para modelos digitais. Árduas tarefas de modelagem e cansativas medições podem ser poupadas no intuito de acelerar o processo de desenvolvimento, análise e documentação de produtos. As aplicações são inúmeras, seja na área comercial, engenharia ou médica entre outras. Ao converter modelos em informações digitais, o escâner possibilitará que um profissional de saúde, por exemplo, possa enviar por meio eletrônico a um colega distante uma cópia digital de um molde ou prótese, para que o mesmo analise digitalmente ou imprima a mesma em uma impressora 3D, ou então, que um artista plástico possa modelar um objeto em qualquer matéria-prima e o converta em arquivo digital.

O desenvolvimento de um produto, tal como um escâner 3D, requer conhecimentos que vão além da simples tecnologia usada para o escaneamento tridimensional. São necessários também projetos para a estruturação e invólucro do produto, utilizando conceitos mecânicos. É necessária a interação entre usuário e o escâner através de uma interface por *software* (programa). E também um sistema embarcado para controle do produto, e comunicação com o PC (*Personal Computer*).

A parceria entre os acadêmicos e a Microbras¹ permitiu que o projeto mecânico e sua execução fossem possíveis devido ao auxílio prestado pelos profissionais capacitados da empresa. O fato de a empresa produzir e fornecer impressoras 3D, também tornou possível a rápida prototipagem de peças utilizadas na montagem do escâner.

No caso do sistema embarcado, a necessidade de uma unidade que possui processamento rápido, interfaces para conexão de periféricos e comunicação, além de um tamanho compatível com o produto, fez com que o *kit* (conjunto) *Raspberry Pi* fosse selecionado para o projeto.

A placa microprocessada *Raspberry Pi* foi criada com o intuito de incentivar o estudo da ciência da computação em escolas, mas devido ao seu baixo custo, pequeno dimensional (8,6 cm de comprimento por 5,4 cm de largura) e baixo consumo (aproximadamente 3,5 W) aliados ao fato dela possuir todos os componentes soldados em uma única placa, popularizou seu uso entre hobistas em diversos projetos de hardware e software livres (About us | Raspberry Pi, 2014).

Lançada em 2012, utiliza um processador baseado em ARM (*Advanced RISC Machine*) de 700 Mhz, 512 MB de *RAM* (*Random Access Memory*) além de uma *GPU* (*Graphics Processing Unit*) integrada ao processador, construído com a tecnologia *System on Chip*. Esta tecnologia faz com que *CPU* (*Central Processing Unit*), *RAM* e *GPU* sejam construídas em um mesmo circuito integrado, maximizando desempenho e reduzindo consumo. (FAQs | Raspberry Pi, 2014).

A placa possui armazenamento não volátil através de cartão *SD*, e portas *USB*, *Ethernet* e *HDMI*, dentre outras, facilitando a integração com outros hardwares. Ou seja, a *Raspberry Pi*, é um computador de baixo custo com grande poder de processamento e integração, permitindo inclusive, a utilização de sistema operacional, como o *Raspbian*, que é o sistema recomendado derivado da distribuição *Debian* do *Linux*.

Outro fator decisivo na escolha da *Raspberry Pi*, foi a disponibilidade do módulo *Raspicam*. Fabricado pela OmniVision, este módulo contém uma câmera modelo OV5647 com apenas 8,5 x 8,5 x 5 mm. Com uma resolução de 5 MP (*Mega Pixels*) e foco fixo, a *Raspicam* permite vídeos em *HD* (*High Definition*), permitindo uma taxa de 30 quadros por segundo a uma resolução de 1080p (1080

¹ Microbras LTDA. Empresa parceira do projeto.

linhas de resolução vertical com *progressive scan*) ou uma taxa de 60 quadros por segundo a uma resolução de 720p.

Também, em função do modelo de escâner proposto, uma análise dos aspectos da geometria projetiva se faz necessária para a constituição do modelo matemático da parte ótica do sistema, responsável por converter informações obtidas em coordenadas de imagens (pixels) para coordenadas reais e tridimensionais que sejam compatíveis com o objeto escaneado.

Para o desenvolvimento de interfaces no PC, o Qt oferece uma proposta interessante. Qt é uma aplicação multi-plataforma com suporte à linguagem C++. Desenvolvida pela *Digia*, permite a criação de softwares com GUI (*Graphical User Interface*), e está disponível sobre licença comercial sem restrições (Qt Licensing | Qt Project, 2014). Através dele, será possível a criação de uma interface eficiente e simples para que o usuário controle e utilize as funcionalidades do escâner.

Além da seleção de sistemas utilizados no projeto, como mecânico, eletrônico e software computacional, a forma e tecnologia utilizada para o escaneamento será discutida e comparada com as técnicas existentes para obter uma conclusão do método mais interessante e viável para este trabalho. Com a seleção dos métodos e materiais este trabalho deve prover uma solução prática que será então, submetida a uma análise criteriosa de seu desempenho, permitindo futuras comparações e melhorias.

1.1 OBJETIVOS DO TRABALHO

A proposta de trabalho busca viabilizar a criação de um escâner 3D que venha a atender necessidades de mercado como custo, qualidade e robustez, além de complementar a linha de produtos da empresa parceira do projeto. Através da análise e estudos das técnicas e produtos existentes, buscar-se-á uma solução que atenda aos requisitos básicos de funcionamento de um escâner 3D, incluindo qualidade e custo.

1.2 VISÃO GERAL

O projeto contempla a construção de um protótipo capaz de digitalizar tridimensionalmente modelos reais. Para isto, métodos e materiais envolvidos são inicialmente descritos e analisados para adequada compreensão do produto criado.

O sistema final é composto de partes específicas, derivadas de diferentes áreas da engenharia eletrônica e, também, da mecânica. Abaixo, são descritos de forma simplificada estes componentes.

- Sistema mecânico para estruturação, fechamento e partes móveis do sistema.
- Sistema embarcado para controle dos periféricos do escâner: câmera, motor de passo e interface de comunicação com o PC.
- Sistema óptico constituído de uma câmera digital para aquisição das imagens do modelo.
- Laser com feixe em forma de linha utilizado para o método de aquisição escolhido.
- Fonte de alimentação do sistema.
- *Software* (programa) responsável pela interface e controle do escâner através do PC.

Uma visão completa do sistema descrito é ilustrada na Figura 1.

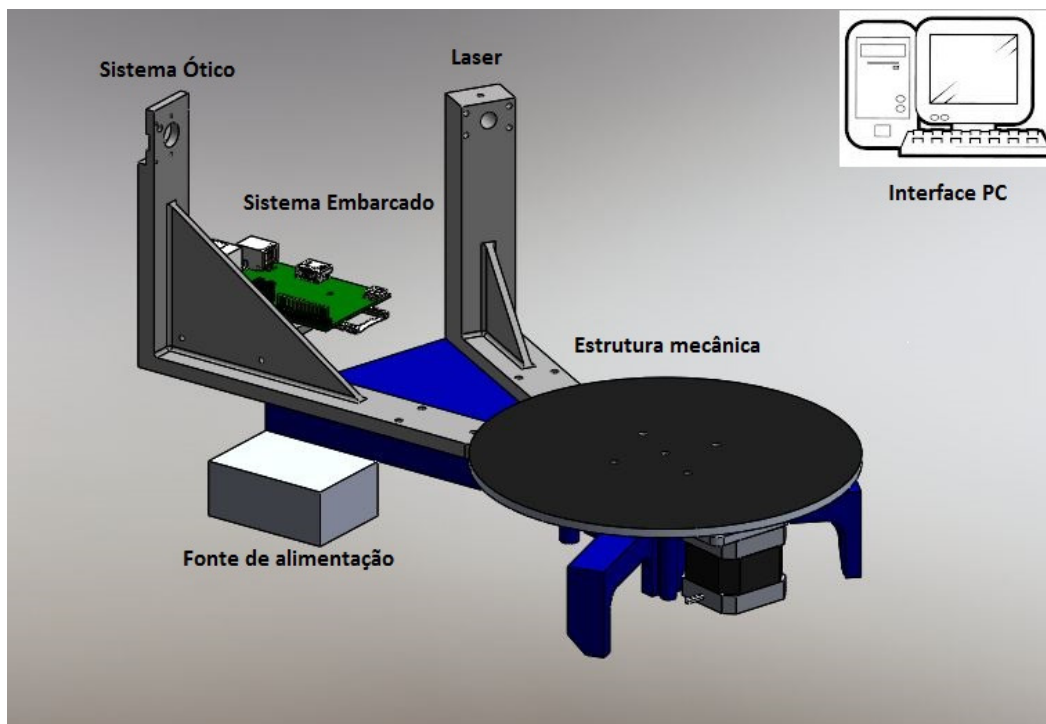


Figura 1. Partes integrantes do sistema

1.3 ESTRUTURA DO TRABALHO

O desenvolvimento do trabalho está dividido em 3 capítulos (2,3, e 4), o Capítulo 2 descreve os métodos de escaneamentos existentes, e apresenta as razões que motivaram a escolha do método utilizado no projeto. O Capítulo 3 traz a introdução dos assuntos teóricos como a geometria projetiva, conceitos relacionados a câmera, distorção em imagens, calibração de câmeras e equações trigonométricas.

Os materiais e métodos utilizados são mostrados no capítulo 4, onde são descritos os projetos mecânicos, eletrônico, software computacional e todos seus pré-requisitos.

O Capítulo 5 é utilizado para análise de resultados e conclusões.

2 SELEÇÃO DO MÉTODO DE ESCANEAMENTO

2.1 MÉTODOS DE ESCANEAMENTO

A forma de objetos em 3D pode ser adquirida através de diversas técnicas com diferentes potenciais e custo. No topo estão os CT scanners (*computed tomography*) que utilizam raios X para construir detalhados objetos. Aliadas ao baixo custo estão técnicas que utilizam câmeras de vídeo e fazem a reconstrução de um modelo tridimensional através de análise de movimento e ou contornos das imagens (BERNARDINI; RUSHMEIER, 2002).

De acordo com Lanman e Taubin (2009), escâneres sem contato e óticos, podem ser divididos em: aquisição através de métodos passivos e aquisição através de métodos ativos.

Métodos passivos não requerem nenhum tipo de controle de iluminação, dependendo apenas da iluminação ambiente. Estereoscopia, por exemplo, é um dos métodos passivos, onde uma estimativa de localização tridimensional é realizada através do uso de duas imagens capturadas de posições diferentes. Um problema recorrente do método passivo é a correspondência entre as imagens, que devem ser conseguidas através de algoritmos que ainda hoje são desafiadores.

Métodos ativos são, então, capazes de superar o problema da complexidade da correspondência de imagens. Neste caso, uma das formas de captura de imagem pode ser substituída por uma fonte de luz controlada, um feixe de laser, por exemplo. Então, um processo de triangulação pode ser utilizado para definir a posição espacial dos pontos da superfície de um objeto.

Uma extensão do modelo simples de triangulação por um ponto de laser e uma câmera é o método por feixe de laser onde várias triangulações podem ser realizadas com uma única imagem.

O slit escâner é amplamente utilizado com sistema baseado na triangulação 3D, com feixe de laser e câmera porque possui simplicidade ótica, mecânica e de custo. O slit escâner é uma extensão da detecção linear de um ponto, permitindo a projeção de um feixe de laser contínuo, capturando os pontos através de um frame de um vídeo (Traduzido de BLAIS,2004).

Para pequenos campos de visão, com ângulos entre 20° e 30°, os *slit scanners* são eficientes e simples. Para maiores ângulos, uma análise anamórfica deve ser realizada e, então, seu uso se torna mais complexo e dispendioso. Anamorfismos em câmeras ocorrem devido as suas lentes, onde a ampliação vertical é diferente da horizontal, deformando a imagem. Outro inconveniente desta técnica é a baixa imunidade à luz ambiente que pode ser melhorada com uma iluminação controlada ou ambiente fechado (BLAIS, 2004).

2.2 SELEÇÃO DO MÉTODO UTILIZADO

Para o desenvolvimento deste trabalho, optou-se pelo sistema de aquisição ativo através da triangulação por feixe de laser e câmera, ou também chamado *slit scanner*, devido aos seguintes fatores:

- Sua simplicidade ótica e mecânica permitiu que o trabalho pudesse ser realizado e concluído em tempo hábil à apresentação deste estudo.
- O protótipo possui um custo reduzido e requer componentes que foram de fácil e rápida aquisição.
- Permite o escaneamento de objetos que podem ser reproduzidos através de impressoras 3D sem maiores dificuldades (principal objetivo do produto para a empresa).
- Para atender ao tamanho da área de impressão da 3D Cloner², permite-se que seja criado um escâner de mesa, com tamanho reduzido e de fácil transporte e utilização.
- Permite a utilização de algumas soluções em sistemas embarcados, como a placa microprocessada *Raspberry Pi* e seu módulo de câmera, *Raspicam*, para uma fácil integração de hardware.
- Rápidos escaneamentos e com controle na densidade de pontos para descrição do modelo real, permitindo modelos com pouca informação e ou modelos com grande quantidade de informação. Quanto mais pontos digitais

² Impressora 3D fabricada pela empresa Microbras, parceira do projeto.

representarem um objeto, mais preciso este será e mais próximo do real será sua reconstrução.

- Pode alcançar uma boa precisão (na faixa de décimos de milímetros) através da utilização de componentes de qualidade e robustez no software empregado.
- É o sistema utilizado em escâners comerciais com mesma finalidade proposta: a aquisição de modelos passíveis de impressão. Um exemplo deste é o *MakerBot Digitizer*³, lançado inclusive, no período de desenvolvimento deste trabalho.

2.3 SLIT SCANNER

O princípio utilizado neste tipo de escâner pode ser analisado através da Figura 2.

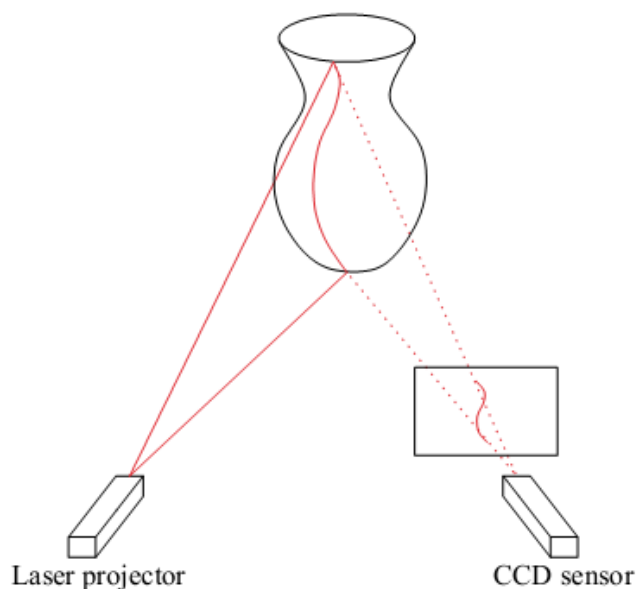


Figura 2. Princípio da Triangulação. Com laser e câmera

Fonte: Bernardini e Rushmeier, 2002

³ Escâner fabricado pela MakerBot disponível em: <http://store.makerbot.com/digitizer>

Um laser em formato de linha é projetado sobre a superfície do objeto, e a câmera captura uma imagem contendo este feixe. Através da posição do pixel na imagem e conhecendo-se a posição da câmera, pode-se estimar um vetor que passa sobre o pixel correspondente na imagem e o ponto de interesse no objeto.

Conhecendo-se a posição do laser, um plano definido pela projeção de sua luz pode ser equacionado, e, a intersecção deste plano com o vetor anterior, descreverá o ponto triangulado que conterá a informação espacial do objeto.

Para a completa triangulação da linha de laser projetada sobre o objeto, cada pixel correspondente deverá ser triangulado. Já para um escaneamento completo do objeto, é possível utilizar uma mesa giratória. No qual, através de sua rotação, o sistema será capaz de detectar e triangular informações relativas à toda sua superfície exterior.

Uma limitação fundamental do que pode ser escaneado através da triangulação é ter uma vista livre para a câmera, sobre o ponto da superfície escaneada no momento que contém o laser refletido em sua superfície (BERNARDINI; RUSHMEIER, 2002).

3 TEORIA E CONCEITOS

Apresentam-se aqui, conceitos relacionados às técnicas utilizadas no desenvolvimento do trabalho, bem como características referentes ao material utilizado.

3.1 GEOMETRIA PROJETIVA

De acordo com Morvan (2009), a geometria projetiva consiste de um arranjo matemático para imagem de múltiplas vistas em 3D e computação gráfica tridimensional. É utilizada como modelo no processo de formação de imagem, geração de imagens sintéticas ou reconstrução de objetos tridimensionais a partir de múltiplas imagens.

Na modelagem de linhas, planos e pontos em um espaço tridimensional se utiliza geralmente a geometria Euclidiana. No entanto, pontos no infinito não podem ser modelados e são considerados casos especiais.

Este caso especial pode ser representado por um desenho perspectivo como o da Figura 3.

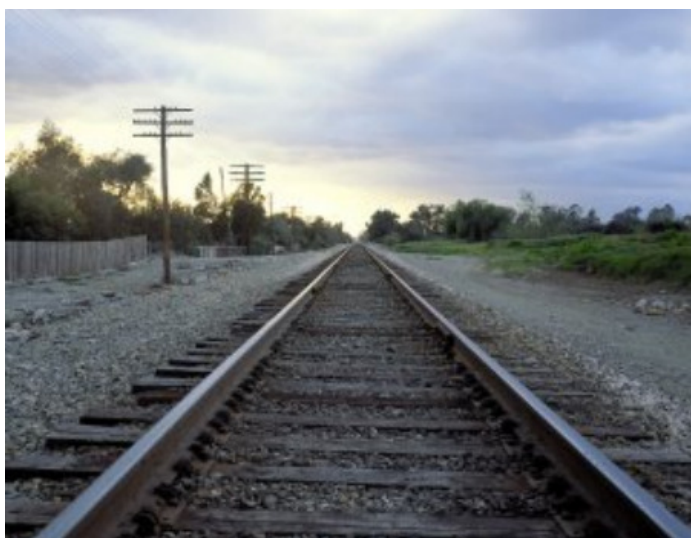


Figura 3 - Dois trilhos paralelos se encontram no ponto em que desaparecem no horizonte
Fonte: Morvan, 2009.

Outra desvantagem do modelo Euclidiano é que para projeções de pontos tridimensionais em planos de imagem requer-se uma operação de escalonamento por perspectiva, resultando em uma operação não-linear.

A geometria projetiva constitui um arranjo atrativo para contornar estas desvantagens. No espaço Euclidiano, um ponto definido em três dimensões é representado por um vetor de três elementos $(X, Y, Z)^T$ ((X, Y, Z) transposto). No espaço projetivo, este mesmo ponto é descrito utilizando um vetor de quatro elementos (X_1, X_2, X_3, X_4) de forma que (MORVAN, 2009):

$$X = X_1/X_4, \quad Y = X_2/X_4, \quad Z = X_3/X_4 \quad \text{sendo } X_4 \neq 0$$

O conjunto $(X, Y, Z)^T$ é normalmente chamado de coordenadas não-homogêneas e o conjunto (X_1, X_2, X_3, X_4) , é chamado de coordenadas homogêneas. Estes conceitos serão utilizados na composição do modelo da câmera, apresentados na seção 3.2.

3.2 CÂMERA

Imagens geradas por câmeras são obtidas através da detecção de luz. A luz gerada por alguma fonte (sol, lâmpada) é irradiada até que seja incidida sobre um objeto onde terá parte de sua irradiação absorvida e, o restante refletido poderá ser percebido na forma de cor, podendo então ser captado por um sensor de imagem de uma câmera.

O entendimento da geometria envolvida neste processo é importante para aplicações de visão computacional como a utilizada no desenvolvimento deste trabalho.

Um modelo simples, mas útil, de como isso acontece é o *pinhole camera model*. Um *pinhole* é uma parede imaginária com um pequeno furo em seu centro que bloqueia todos os raios de luz com exceção daqueles que passam pela abertura no centro (BRADSKI E KAEHLER, 2008).

Infelizmente, este modelo aplicado na prática depende de um tempo de exposição elevado para que seja coletada uma quantidade de luz suficiente para uma boa imagem. Para isto, são inseridas lentes na câmera, que fazem com que uma quantidade maior de raios converjam para sensor de imagem, acelerando o processo de captura.

O problema de se incluir lentes em câmeras é a introdução de distorções, gerando um modelo matemático mais complexo. Neste caso, um processo chamado calibração da câmera é importante para criar uma relação entre medidas na câmera (em unidades de *pixels*) com medidas tridimensionais reais (polegadas, milímetros, metros, entre outros).

O processo de calibração da câmera provê tanto o modelo da geometria da câmera como também o modelo da distorção das lentes. Estas duas informações definem os parâmetros intrínsecos da câmera que serão utilizados para interpretar uma cena real (BRADSKI E KAEHLER, 2008).

3.2.1 Modelo da Câmera Pinhole

Após a passagem da luz pelo orifício do modelo *pinhole*, esta é projetada em um plano de imagem. E o resultado disto é uma imagem definida neste plano sempre em foco. A distância do objeto com relação ao tamanho da imagem será definida pelo seu comprimento focal (f). No modelo *pinhole*, esta distância é precisamente a distância entre o orifício e o plano da imagem formada, conforme a Figura 4.

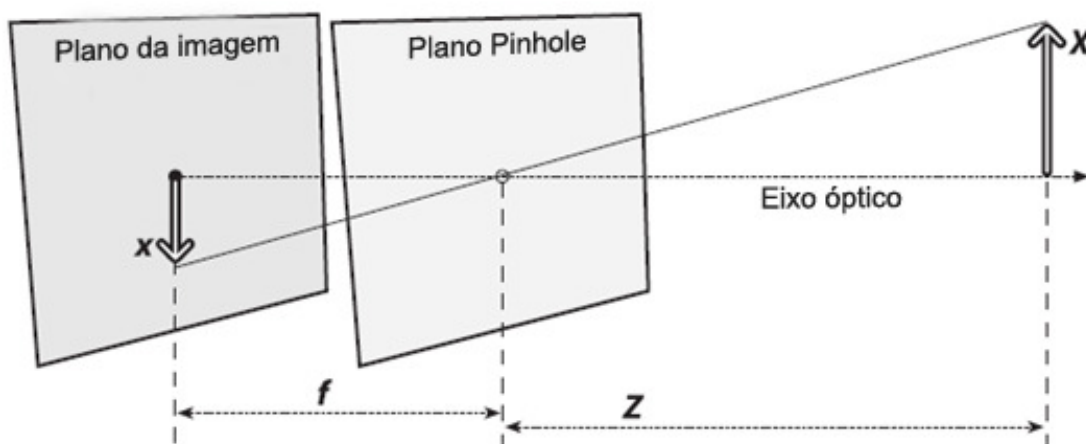


Figura 4. Modelo básico de câmera Pin Hole.
 Fonte: Adaptado de Bradski e Kaehler, (2008).

Através da semelhança de triângulos, pode-se chegar a seguinte definição:

$$-x = f \frac{X}{Z}$$

E através dessa definição, a projeção de um ponto tridimensional real $(X, Y, Z)^T$ projetado sobre um plano de imagem com coordenadas em pixels $(x, y)^T$, pode ser escrita como:

$$x = f \frac{X}{Z} \quad y = f \frac{Y}{Z}$$

O sinal negativo agora deixa de existir, pois o plano da imagem não é mais considerado como estando de ponta cabeça, a Figura 5 mostra esta transformação.

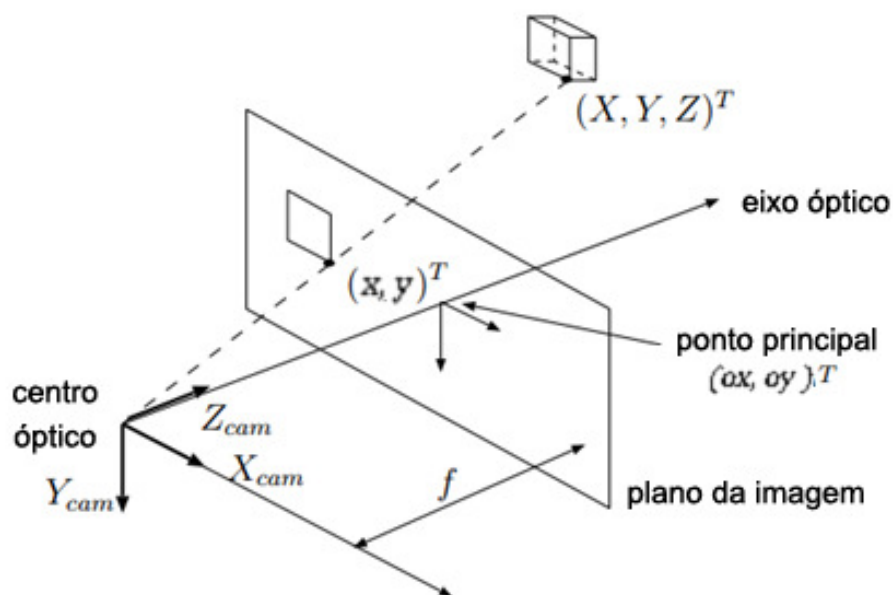


Figura 5. Descrição da relação entre um ponto 3D (X, Y, Z) e sua correspondente projeção (u, v) no plano de imagem.
 Fonte: Adaptado de Morvan, (2009).

Para evitar operações não lineares uma reformulação das equações anteriores utilizando-se da geometria projetiva pode ser realizada, tal que:

$$(\lambda x, \lambda y, \lambda)^T = (Xf, Yf, Z)^T$$

$\lambda = Z$ que representa o fator de escala homogêneo da coordenada.

O modelo pode ser expresso na forma matricial como segue:

$$\lambda \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Para uma relação adequada com as câmeras existentes, o valor de f pode ser descrito separadamente para o eixo vertical e horizontal do sistema de imagens, pois diferentes comprimentos focais podem existir para ambos os eixos em certas câmeras. Portanto, utiliza-se f_x para a equação aplicada ao valor de X e f_y para Y .

Além disso, imagens tem sua origem, (coordenada (0,0) dada em pixels), geralmente no canto superior esquerdo, mas a origem do sistema de coordenadas da câmera se refere ao centro da imagem ou *ponto principal* $(o_x, o_y)^T$ como mostrado na Figura 5.

Portanto, é necessária uma transformação de coordenadas. Utilizando-se do sistema de coordenadas homogêneas, pode-se reescrever a matriz de projeção da equação anterior da seguinte forma:

$$\lambda \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & o_x & 0 \\ 0 & f_y & o_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = [\mathbf{M} \quad \mathbf{0}_3] \mathbf{P} \quad (3.1)$$

Considerando $\mathbf{P} = (X, Y, Z, 1)^T$, \mathbf{M} pode ser definida como a matriz que contém os parâmetros intrínsecos da câmera e seus valores poderão ser obtidos através do processo de calibração da câmera.

3.2.2 Distorções na Câmera

Existem dois principais tipos de distorções em imagens geradas por câmeras. A distorção radial e a distorção tangencial.

Distorções radiais são efeitos causados pela utilização de lentes. Em teoria, é possível criar lentes que não geram distorção, mas na prática e devido aos processos de manufatura utilizados, câmeras comuns possuirão uma pequena quantidade de distorção radial.

Este tipo de distorção é percebido nos pixels próximos da borda do sensor de imagem onde a distância radial é maior, criando um efeito do tipo 'barril' ou 'olho-de-peixe'. Linhas retas serão capturadas e mapeadas como linhas curvas. A Figura 6 ilustra este efeito através de um mapa de pixels.

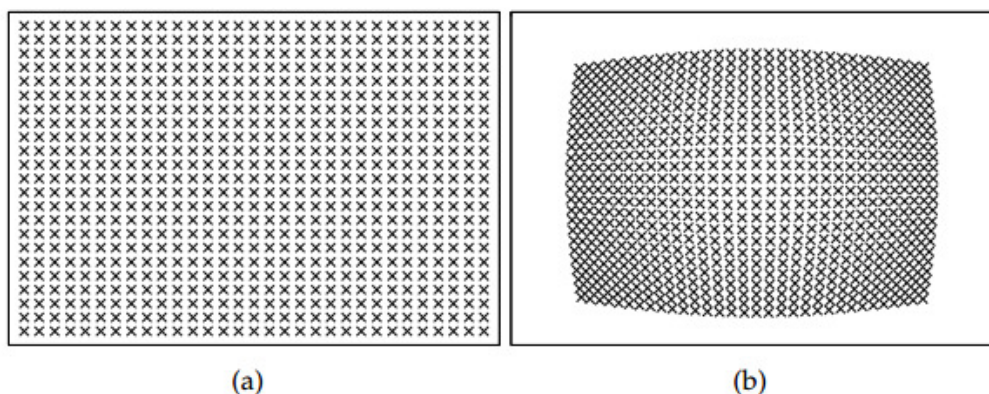


Figura 6. Distorção radial aplicada a um mapa de pixels. (a) Imagem sem distorção (b) Imagem com distorção
Fonte: Morvan, (2009).

Em distorções radiais, a distorção é nula no centro óptico (centro da imagem) e aumenta conforme se move para a borda da imagem. De acordo com Bradski e Kaehler (2008), na prática, esta distorção é pequena e pode ser caracterizada pelos primeiros termos de uma expansão de Taylor em torno do ponto $r = 0$ (centro da imagem). Para câmeras comuns e que podem apresentar considerável distorção, utilizam-se os três primeiros termos não nulos nomeados como k_1, k_2 e k_3 .

A forma do polinômio derivado da série de Taylor pode ser escrito da seguinte forma:

$$f(r) = a_0 + a_1r + a_2r^2 + \dots$$

Neste caso, como $f(r) = 0$ para $r = 0$, $a_0 = 0$. Outra consideração em torno da expansão é de que a função precisa ser simétrica em r . Para isto, apenas os coeficientes com expoentes pares são considerados: r^2, r^4 e r^6 .

E as funções que descrevem a correção do posicionamento dos pixels na imagem poderão ser descritas da seguinte maneira:

$$x_{\text{corrigido}} = x(1 + k_1r^2 + k_2r^4 + k_3r^6)$$

$$y_{\text{corrigido}} = y(1 + k_1r^2 + k_2r^4 + k_3r^6)$$

As Figuras 7(a) e 7(b) representam imagens antes e após a correção da distorção radial.

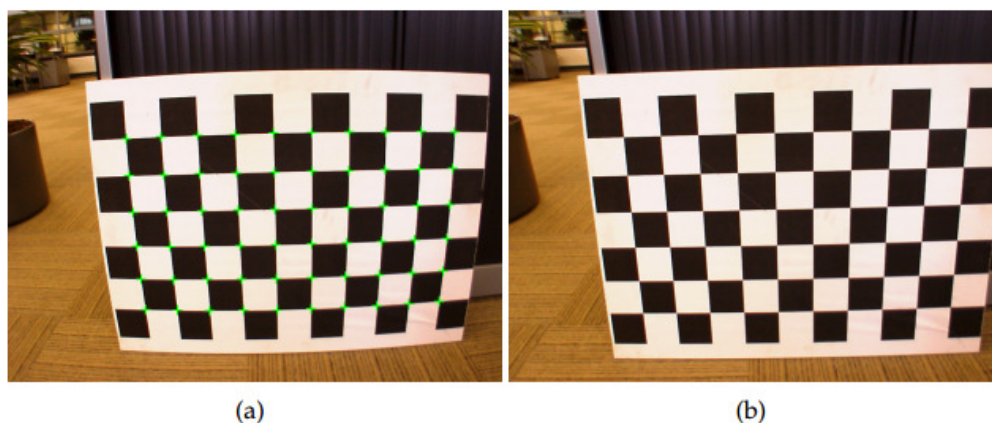


Figura 7. (a) Imagem com distorção radial (b) Imagem com distorção corrigida através das equações x .
Fonte: Morvan, (2009).

A distorção tangencial, por outro lado, é menos comum e ocorre devido à má qualidade da montagem da câmera em si.

Ela pode ocorrer devido ao posicionamento do sensor de imagem (CMOS chip neste caso) não se encontrar paralelo à lente do dispositivo, conforme ilustrado na Figura 8.

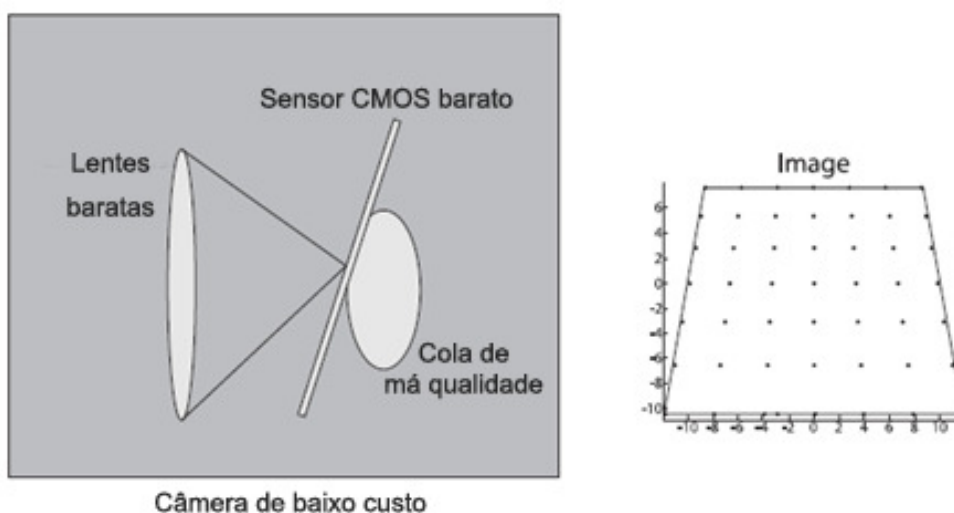


Figura 8. Ilustração de distorção Tangencial
Fonte: Adaptado de Bradski e Kaehler, (2008).

Este tipo de distorção pode ser corrigido através das seguintes equações:

$$x_{\text{corrigido}} = x + [2p_1 + p_2(r^2 + 2x^2)]$$

$$y_{\text{corrigido}} = y + [p_1(r^2 + 2y^2) + 2p_2x]$$

Onde os parâmetros $(x_{\text{corrigido}}, y_{\text{corrigido}})$ são os pontos em pixels com as distorções removidas, (x, y) são os valores em pixels originais contendo distorção. Os termos p_1 e p_2 são os coeficientes para correção e r é o raio em pixels para qual o ponto atual faz parte e que parte do centro da imagem.

As equações para correção deste tipo de distorção serão apenas citadas. Demonstrações a respeito deste tema podem ser encontradas em “Decentering Distortion of Lenses” por Duane C. Brown (1966).

3.2.3 Calibração e Obtenção de Parâmetros Intrínsecos

O conjunto de parâmetros relacionados à distância focal, centro focal e coeficientes de distorção descritos anteriormente são os parâmetros intrínsecos de uma câmera e serão necessários para os cálculos utilizados no processo de triangulação do escâner.

O processo de obtenção destes parâmetros não será demonstrado matematicamente, pois além da complexidade envolvida, existem ferramentas e algoritmos específicos que realizam esta calibração e fornecem estes parâmetros para utilização no campo da visão computacional e reconstrução de modelos.

Dentre os softwares que disponibilizam este tipo de ferramenta, podem ser citados *Matlab®* e a biblioteca *OpenCV*, sendo estes os mais comuns e mais utilizados em aplicações envolvendo este tipo de necessidade.

Neste trabalho, será utilizado a biblioteca *OpenCV* para calibração e funções de filtros para imagem, bem como para correções de distorções.

3.2.4 Parâmetros Extrínsecos

Diferentemente dos parâmetros intrínsecos da câmera, que descrevem componentes relacionadas à própria câmera e suas características, existem os parâmetros extrínsecos. Estes indicam a posição e orientação da câmera no sistema de coordenadas tridimensionais reais.

Para cada imagem tirada com a câmera é possível descrever a posição relativa do objeto com relação ao sistema de coordenadas da câmera em termos de sua rotação e translocação.

Conforme mostra a Figura 9, um ponto P_0 no sistema de coordenadas espaciais está associado a um ponto p_c no sistema de coordenadas da câmera, através dos parâmetros R que seria uma matriz de rotação e t que seria um vetor de translocação. Sendo possível estabelecer a seguinte relação:

$$P_c = R(P_0 - T)$$

Onde P_c é a transformação da coordenada p_c para coordenada homogêneas caracterizando-se por $(x, y, 1)^T$.

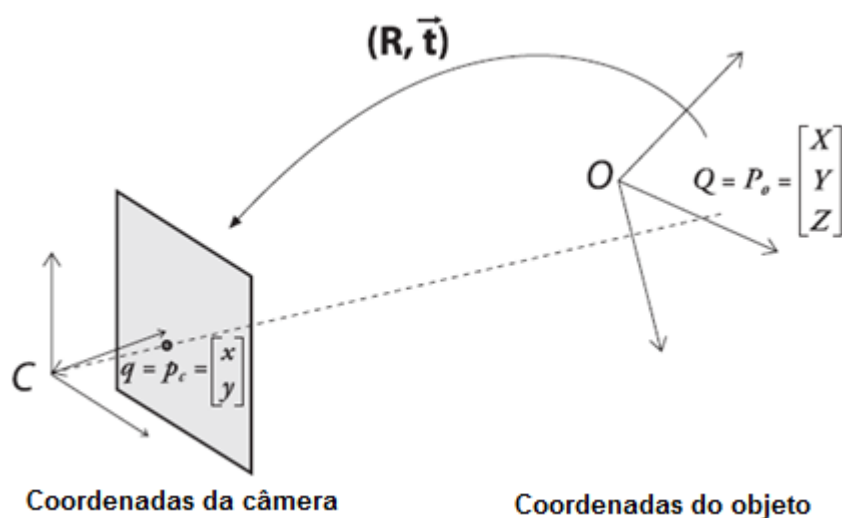


Figura 9. Transformação de coordenadas
Fonte: Bradski e Kaehler, (2008).

Estes parâmetros também podem ser estimados através de algoritmos, inclusive os disponíveis na biblioteca OpenCV utilizada neste projeto. A demonstração detalhada deste processo será descrita na seção Materiais e Métodos.

Para utilização destes parâmetros, ambos podem ser descritos através de uma matriz e vetor dos tipos:

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \text{ e } \mathbf{t} = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}$$

Sendo os r_{ij} os valores da matriz de rotação e t_i os valores do vetor de translocação.

3.2.5 Modelo Completo do Sistema Ótico

A junção do modelo de projeção descrito pela equação 3.1, com os parâmetros extrínsecos representarão o modelo utilizado neste projeto para o auxílio no processo de triangulação dos pontos capturados pelo escâner.

Este modelo permite a conversão de pontos em imagens, descritos em pixels, serem convertidos para uma coordenada tridimensional real e com origem e unidades definidas pelo usuário na hora da definição dos parâmetros extrínsecos.

A junção com suas devidas simplificações será descrita pelo seguinte modelo:

$$\lambda \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \\ \mathbf{1} \end{bmatrix} = \mathbf{M} \left(\mathbf{R} \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \\ \mathbf{Z} \end{bmatrix} + \mathbf{t} \right)$$

O sistema também pode ser escrito como:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = R^{-1} * \left(M^{-1} * \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} - t \right) \quad (3.2)$$

Neste caso, R^{-1} e M^{-1} são as matrizes inversas de R e M , respectivamente.

Este último modelo, simplifica sua utilização no sistema do escâner, pois isola os valores que serão calculados na utilização da equação, sendo estes valores o conjunto de coordenadas (X, Y, Z).

3.3 INTERSECÇÃO ENTRE LINHAS E PLANOS

Na triangulação por feixe de laser, é necessário o cálculo da intersecção da linha, que passa pela câmera e vai em direção ao ponto que se deseja triangular, com o plano formado pelo laser.

A Figura 10 representa pontos, retas e planos que ilustram a teoria para o cálculo deste tipo de intersecção.

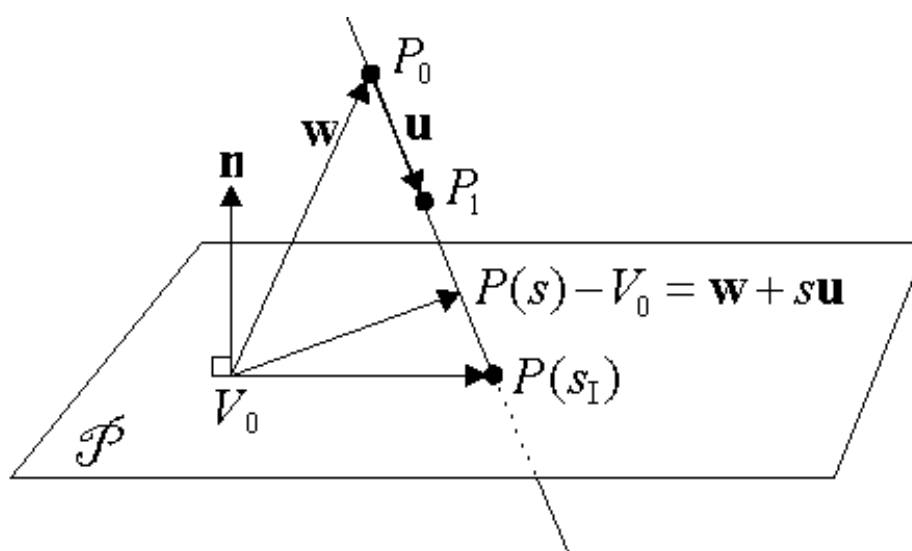


Figura 10. Determinação do ponto de intersecção entre plano e vetor.

Fonte: SUNDAY

Considerando P_0 e P_1 como pontos pertencentes à reta que intersecta com o plano \mathcal{P} , $P(s_I)$ é o ponto desejado, sendo s um fator escalar proporcional ao vetor \mathbf{u} , e s_I é um caso particular deste fator para qual $\mathbf{w} + s_I\mathbf{u}$ será igual ao ponto $P(s_I)$ desejado. V_0 é um ponto pertencente ao plano \mathcal{P} . Já o vetor \mathbf{n} corresponde a um vetor normal ao plano \mathcal{P} . \mathbf{u} é definido por $P_1 - P_0$ e $\mathbf{w} = P_0 - V_0$.

No ponto de intersecção $P(s_I)$ o vetor definido por $P(s) - V_0 = \mathbf{w} + s\mathbf{u}$ será perpendicular à \mathbf{n} . Isto permite que seja definida a seguinte condição:

$$\mathbf{n} \cdot (\mathbf{w} + s\mathbf{u}) = 0$$

Resolvendo s em função dos produtos escalares, obtém-se a solução particular s_I , que será o coeficiente do vetor \mathbf{u} correspondente ao ponto $P(s_I)$

$$s_I = \frac{-\mathbf{n} \cdot \mathbf{w}}{\mathbf{n} \cdot \mathbf{u}} \quad (3.3)$$

O ponto de intersecção será o resultado da aplicação de s_I no vetor:

$$P(s) - V_0 = \mathbf{w} + s_I\mathbf{u} \quad (3.4)$$

4 MATERIAIS E MÉTODOS

Neste capítulo, são descritos os materiais, bibliotecas, softwares e métodos utilizados para o desenvolvimento do escâner.

Inclui-se nesta seção detalhes sobre o projeto mecânico do protótipo desenvolvido, bem como o sistema embarcado e a eletrônica adicional utilizada no controle do escâner e da comunicação. E por fim, o software criado para interface entre escâner e PC.

4.1 PROJETO MECÂNICO

O projeto mecânico foi desenvolvido com o auxílio da empresa parceira do projeto. As medidas, localizações e funcionalidades pertinentes ao funcionamento do escâner foram definidas pelos acadêmicos, ficando a critério da empresa a solução utilizada, os detalhes técnicos e construtivos das partes necessárias e o material envolvido na fabricação.

A divisão do projeto pode ser definida através de quatro importantes partes do sistema:

- Suporte para câmera;
- Suporte para o laser;
- Mesa giratória para colocação do objeto;
- Gabinete de fechamento para controle da iluminação;

A imagem do projeto completo pode ser visualizada na Figura 11. Para o suporte da câmera projetou-se uma peça que coloca a câmera a 100 mm de altura com relação à superfície da mesa giratória. E a distância do centro da mesa até a base do suporte da câmera com aproximadamente 300 mm. Essas medidas permitem que ângulo de visão da câmera englobe toda a área de escaneamento desejada. A Figura 12 mostra o suporte da câmera.

O suporte do laser é muito similar ao da câmera, com a diferença de estar mais próximo do centro da mesa em função de melhorar a sua intensidade quando projetado sobre o objeto. Sua imagem pode ser vista na Figura 13

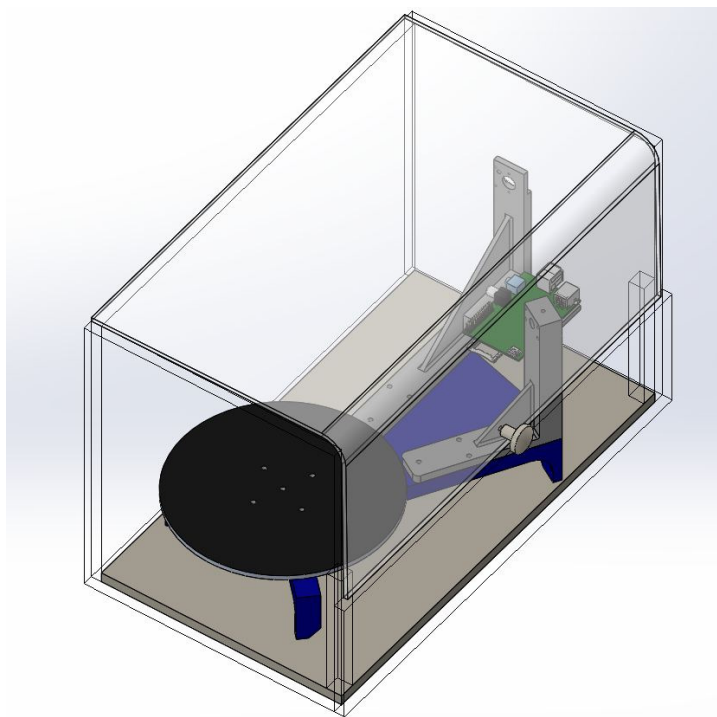


Figura 11. Imagem do projeto do protótipo.

O ângulo entre o suporte da câmera e do laser com relação ao centro da mesa é de extrema importância para o projeto, ele deve ser preciso, pois é considerado nos cálculos de triangulação do escâner.

Baseado na análise dos valores descritos em estudos e nos escâneres existentes utilizou-se um ângulo de 30° que permite pouca obstrução do laser para muitos objetos e permite boa precisão no resultado da triangulação.

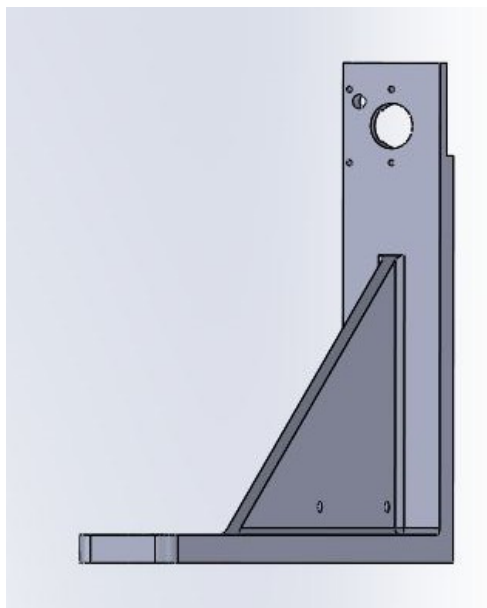


Figura 12. Projeto do suporte da câmera.

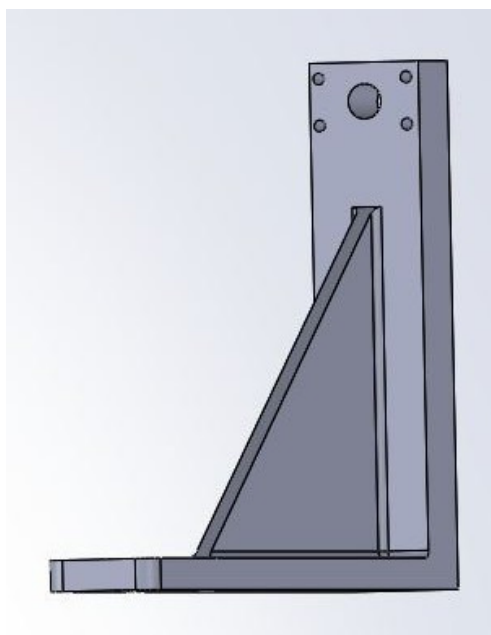


Figura 13. Projeto do suporte do laser.

Outro detalhe pertinente ao suporte do laser é adição de duas chapas metálicas em seu topo para o controle da largura do feixe irradiado até o objeto. Notou-se que sem este artifício o feixe de laser possuía uma largura exagerada e que poderia comprometer a qualidade da detecção e precisão da medição.

A Figura 14 mostra este artifício e logo abaixo, na Figura 15, a diferença do uso ou não destas chapas.

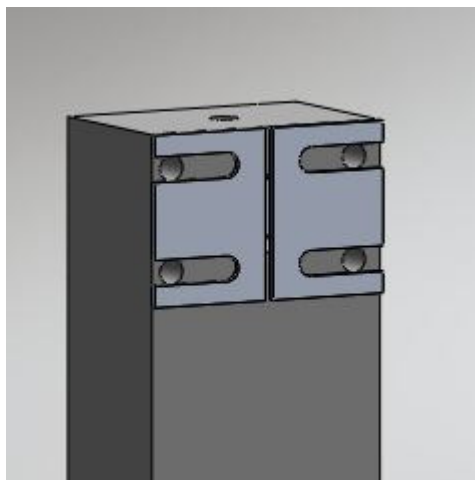


Figura 14. Projeto das chapas metálicas.

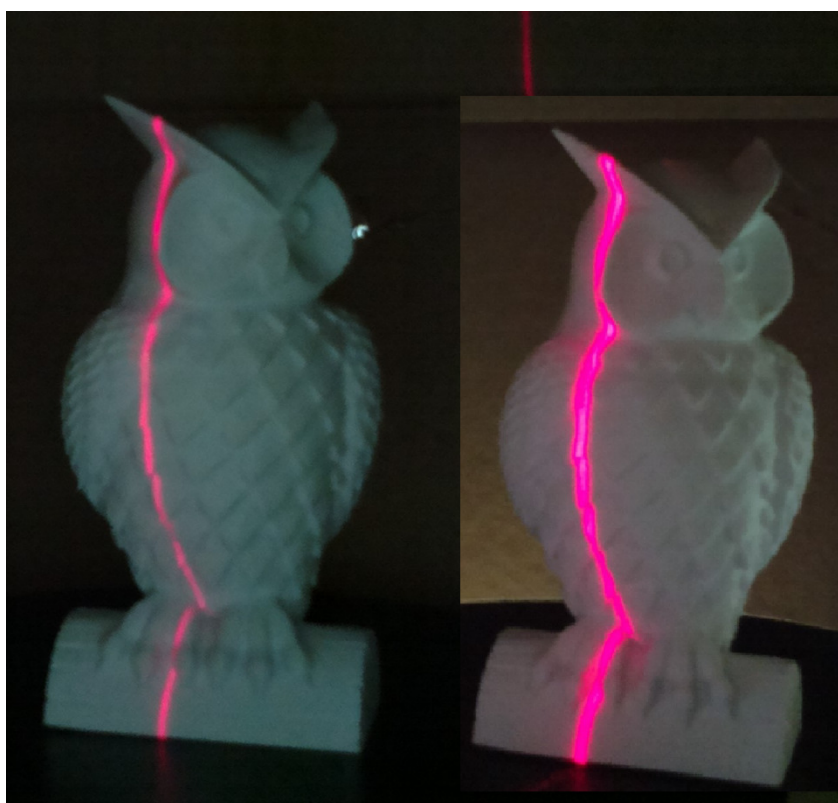


Figura 15. Comparação dos dois feixes de laser. Lado direito laser grosso e lado esquerdo laser fino.

Para a mesa giratória, um sistema de engrenagens conectadas a mesa e ao motor, foi utilizado.

A mesa foi projetada com 200 mm de diâmetro, para compatibilização com o tamanho médio da área de impressão das atuais impressoras 3D, e o sistema de

engrenagens transfere o movimento do eixo do motor para o eixo da mesa giratória. As Figuras de 16 a 19 ilustram os elementos utilizados.

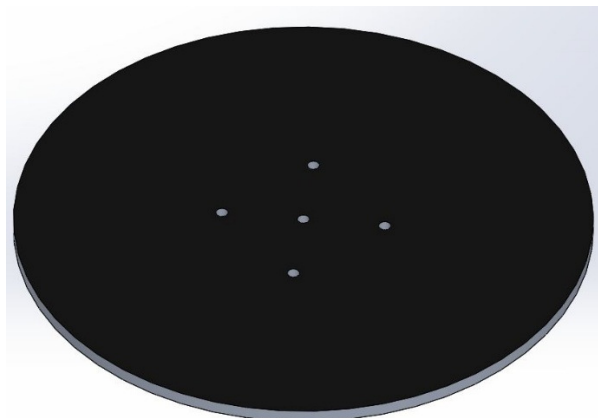


Figura 16. Projeto da mesa giratória.

A Figura 16 representa a mesa com quatro furos para fixação da engrenagem da Figura 17 e seu furo central é utilizado como ponto fiducial.



Figura 17. Projeto da engrenagem de fixação na mesa.

A engrenagem menor da Figura 18 é acoplada diretamente ao eixo do motor e transfere seu movimento para a engrenagem maior parafusada à mesa. A relação do número de dentes da engrenagem e consequentemente no número de voltas é de 1:5. Ou seja, para cada cinco revoluções do motor, ocorrerá uma revolução da mesa giratória. Este detalhe é importante para o controle do motor de passo mostrado posteriormente.

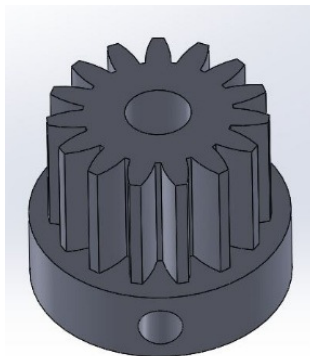


Figura 18. Engrenagem do motor.

Um rolamento também foi adicionado à estrutura fixa do escâner e nele, encaixou-se um eixo usinado conectado à mesa para permitir o seu giro livre de acordo com a movimentação das engrenagens.

A Figura 19 mostra o eixo usinado.

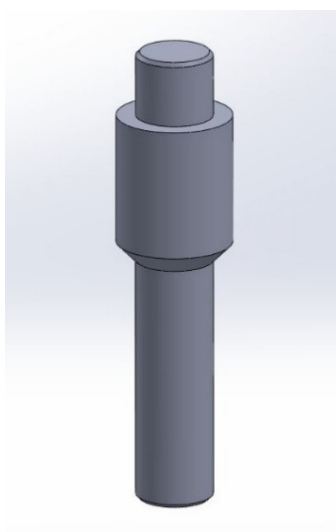


Figura 19. Projeto do eixo.

O gabinete, que pode ser visualizado na Figura 11, foi criado com o intuito de se controlar a luz que incide no ambiente em que se encontra o objeto alvo, sendo este um importante fator para a qualidade do resultado do processo de escaneamento.

Uma vista completa explodida que permite melhor entendimento da montagem completa do escâner está disponível no Anexo I.

4.2 SISTEMA EMBARCADO / ELETRÔNICA

O sistema eletrônico necessário para o escâner requer as seguintes funcionalidades:

- Controle e *driver* para o motor de passo da mesa giratória.
- Interface para câmera utilizada.
- Alimentação para o módulo de laser.
- Processamento adequado para o processamento de imagens e cálculos de triangulação, caso o processamento seja realizado no sistema embarcado.
- Interface de comunicação com o PC para controle do escâner e transferência dos dados relativos ao modelo e ou imagens para processamento.
- Fonte de alimentação com tensões específicas para o conjunto de soluções utilizadas.

4.2.1 Raspberry Pi

Lançado em 2012, o primeiro modelo deste computador reduzido buscavam estimular o ensino da ciência da computação em escolas, retomando o interesse dos jovens pelo estudo da computação básica (About us | Raspberry Pi, 2012). Seu baixo custo e variedade de funções fez com que rapidamente se popularizasse na utilização em pequenos projetos eletrônicos, especialmente entre hobistas. A figura abaixo mostra a placa do *Raspberry Pi* e alguns de seus componentes.

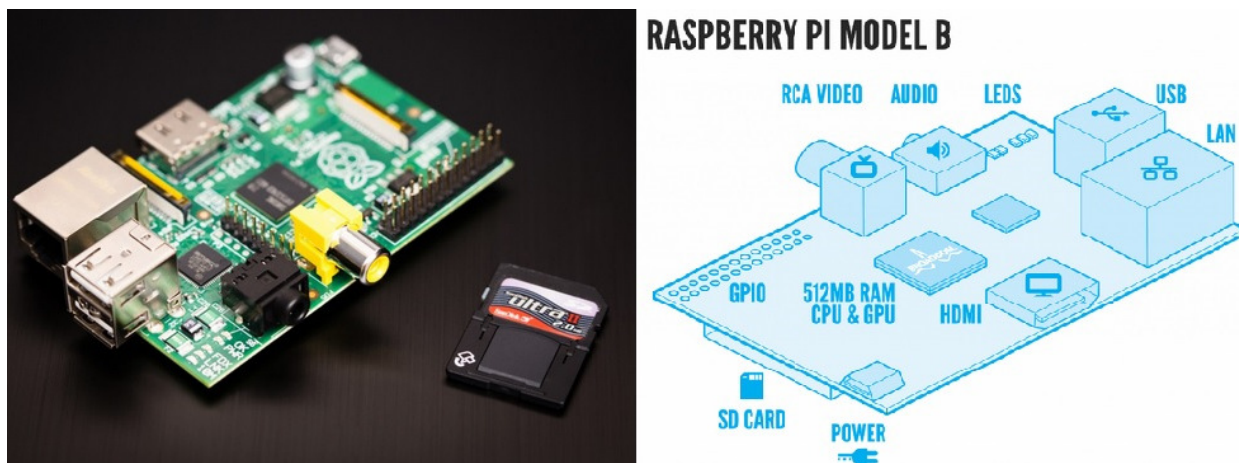


Figura 20. Modelo do Raspberry Pi.

Fonte: Adaptado de <<http://www.raspberrypi.org/faqs>>.

Conforme descrito na documentação do *Raspberry Pi* (FAQs | Raspberry Pi), este computador possui um processador ARM 11 de 700 MHz, uma GPU dedicada e 512 MB de memória RAM incluídas em um único chip: o *Broadcom BCM2835*. Encapsulamento este, possível devido à tecnologia *System on a Chip (SoC)*.

Outras características como a disponibilidade de uma saída HDMI, conexão *Ethernet*, e portas USB, tornaram o *Raspberry Pi* atraente do ponto de vista de projetos eletrônicos para as mais variadas áreas, incluindo robótica, visão computacional, aplicações para *Smart TV's* e até jogos.

A facilidade do uso desta placa é devida ao fato de se poder embarcar nela, sistemas operacionais *Linux*, sendo a versão mais utilizada a *Raspbian*, uma versão não oficial da distribuição *Debian* que é, inclusive, otimizada para o conjunto de instruções *ARMv6* presente na *Raspberry Pi*. (RPi Distributions | 2013).

Os periféricos disponíveis pela *Raspberry Pi* incluem:

- SD Card que atua como forma de armazenamento não volátil do sistema.
- 2 Portas USB 2.0 que podem ser utilizadas para conexão de mouse e teclado.
- Saída de vídeo HDMI que pode ser conectada diretamente a monitores e TV's.
- Saída de vídeo composto.
- Conector Jack para saída de áudio.
- Interface LAN / Ethernet permitindo conexão com internet e acesso a redes.

- DSI (Display Serial Interface) usado principalmente para conexão de displays LCD ou similares.
- CSI (Camera Serial Interface) permite a conexão de módulos de câmera como a *Raspicam*.
- GPIO's (General purpose input/output's) Entradas e saídas genéricas que podem ser utilizadas para aplicações diversas.

A figura abaixo mostra o barramento de GPIO's disponíveis no *Raspberry Pi*. Além de saídas digitais convencionais, o barramento permite ainda o uso de interfaces como UART (Universal Asynchronous Receiver/Transmitter), SPI (Serial Peripheral Interface) e I2C (Inter-Integrated Circuit).

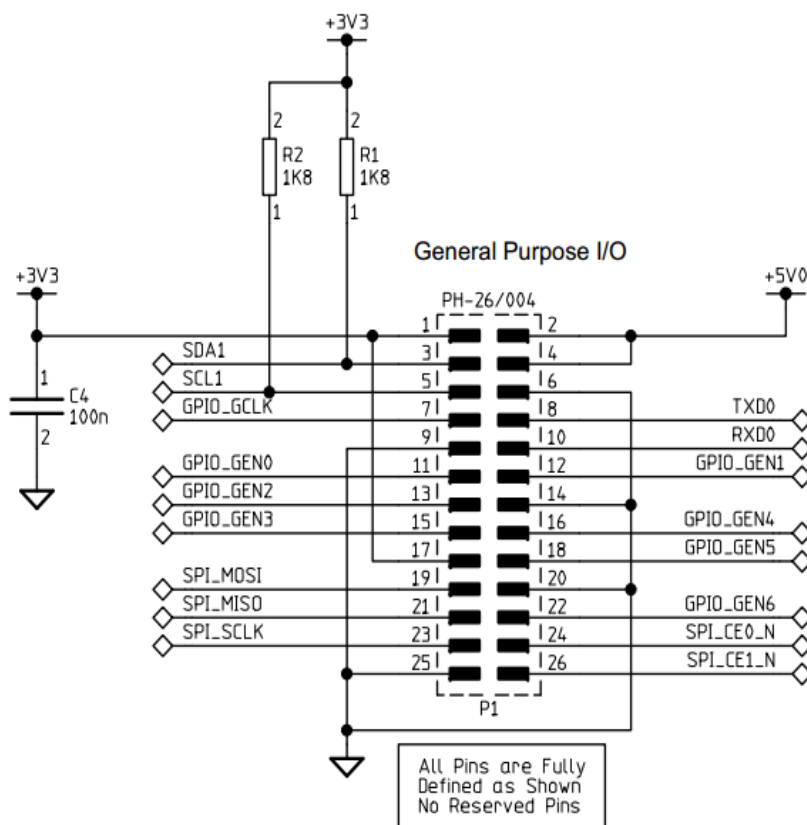


Figura 21. Esquemático das GPIO's do Raspberry Pi.
Fonte: Esquemático do Raspberry Pi.

Estas GPIO's são utilizadas no projeto para controle do driver utilizado para o controle do motor e que será abordado no próximo tópico.

Um cartão de 16 GB de espaço com a versão *Raspbian* do sistema Linux é utilizado como sistema e dispositivo de armazenamento do sistema. É possível obter esta distribuição através de download gratuito na Internet.

Para interface de comunicação entre escâner e PC, é utilizada a interface *Ethernet* disponível no *Raspberry Pi*. Mais precisamente através do compartilhamento e gerenciamentos de arquivos descritos na etapa da criação do software.

O último recurso utilizado e talvez o mais decisivo na escolha da *Raspberry Pi* é o módulo CSI para utilização da câmera *Raspicam*.

4.2.2 RaspiCam

A *Raspicam* é um módulo que pode ser adquirido a parte e adicionado ao *Raspberry Pi* através da interface CSI disponível. Este módulo contém uma câmera modelo OV5647 produzido pela OmniVision, com uma resolução de 2592 x 1944 pixels (5 Mega pixels) para imagens estáticas e um tamanho de apenas 8,5 x 8,5 x 5 mm. Ambos fatores relacionados à resolução e ao tamanho da câmera foram fatores importantes na escolha do módulo.

A câmera suporta formatos como JPEG, PNG, GIF, e BMP e modos não comprimidos em RGB ou YUV. Além de fotos é possível gerar vídeos em 1080p30 (1920x1080x30fps).

Módulos similares encontrados no mercado ou possuíam tamanho não compatível com o projeto, ou eram extremamente caros e, em alguns casos suas interfaces de comunicação demandavam um hardware dedicado não contemplado pela placa utilizada. Portanto, como a *Raspberry Pi* já possui uma interface específica pra esta câmera permitindo a transferência das imagens e o controle da câmera, esta foi a solução adotada para o projeto.

A Figura 22 mostra o modelo de câmera adotado, a *RaspiCam*.

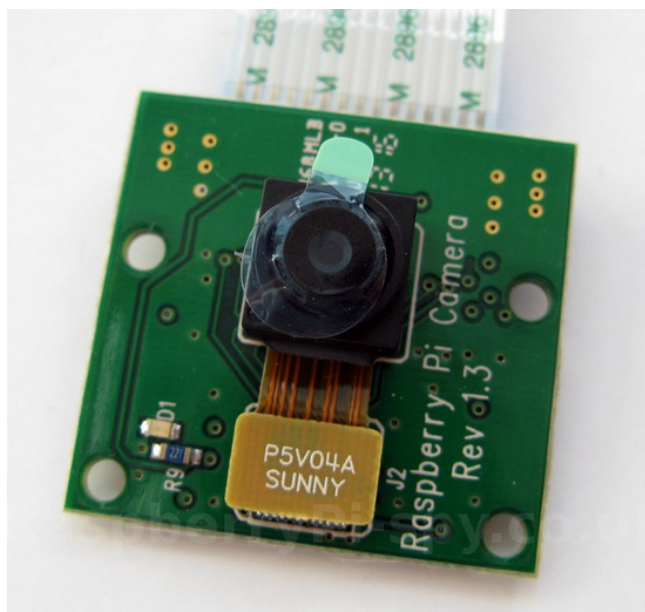


Figura 22. Câmera RaspiCam utilizada.

Um aplicativo para o Linux pode ser instalado e executado através de terminal no Raspberry para a utilização da câmera para captura de imagens. O comando utilizado é nomeado como *raspstill*, que permite diversas configurações, sendo as pertinentes a este trabalho as opções para nome e formato do arquivo e a configuração do tempo de exposição da câmera para melhor qualidade de foto em ambientes escuros como o do escâner.

Para alterar o nome e formato do arquivo basta utilizar a opção *-o*, e para configuração do tempo de exposição, utiliza-se o comando *-t* conforme o exemplo: - *raspistill -o /home/pi/foto.jpg -t 500*: Este comando gravará a imagem capturada no diretório */home/pi* e com o nome *'foto.jpg'*. E a captura da imagem ocorrerá após um tempo de 500 ms de exposição da lente e sensor à luz.

Tempos maiores de exposição permitem a captação de uma quantidade maior de luz pela lente e sensor da câmera e conseqüentemente melhoram fotos realizadas no escuro.

4.2.3 Placa Eletrônica Auxiliar

Com a utilização da placa *Raspberry Pi* e do módulo *Raspicam*, as seguintes necessidades são sanadas:

- Controle para o driver do motor de passo.
- Câmera e interface adequada para captura de imagens.
- Interface de comunicação com o PC através da *Ethernet*.
- Processamento das imagens.

Com isso, ainda são necessários os seguintes recursos:

- Alimentação para o sistema do Raspberry que requer uma tensão de 5 V e uma corrente mínima de 700 mA.
- Alimentação para o laser sendo necessários 3,3 V e 20 mA.
- Driver e alimentação para o motor de passo sendo este um motor de passo modelo SM1.8-A1740CHSE (1.8º NEMA 17) fornecido pela Action Technology e com uma tensão nominal de 4 V(dc) e uma corrente de 1,2 A.

Para atender estes requisitos, o circuito da Figura 23 foi projetado.

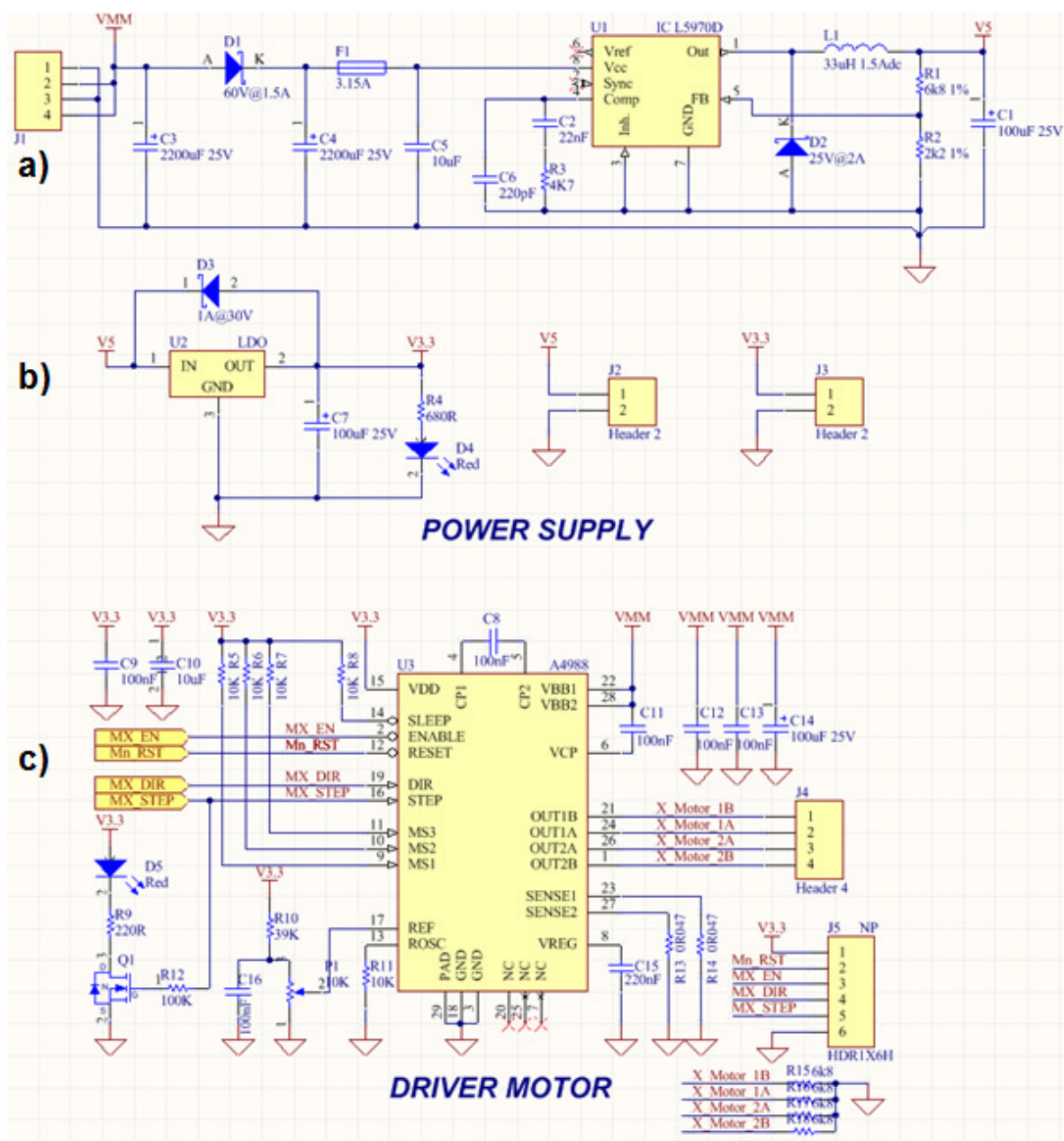


Figura 23. Esquemático da placa auxiliar.

Para a entrada do circuito, considerou-se 2 capacitores eletrolíticos de $2200\mu F$ para estabilidade da tensão devido ao uso do motor de passo e alta corrente consumida pelo circuito de 5 V em função do *Raspberry Pi*. São considerados também, para a entrada do circuito, um diodo D1 para evitar polarização reversa e um fusível F1 para proteção.

J1 representa o conector que receberá a tensão de alimentação do circuito. A alimentação do circuito provém de uma fonte de 24 V por 1,5 A adquirida especificamente para este projeto.

Para a conversão CC de 24 V para 5 V um conversor Buck é utilizado devido a sua melhor eficiência comparada aos conversores lineares. Utilizou-se um circuito integrado L5970D (U1 Figura 23.a) fabricado pela *ST Microelectronics*. Este conversor permite tensões de entrada que podem variar de 4,4 V até 36 V e sua saída é projetada para 1 A podendo chegar a uma corrente máxima típica de 1,87 A. Neste caso, o conversor recebe uma tensão de 24 V e converte para 5 V, sendo a tensão de saída definida pelo divisor resistivo conectado no pino FB do conversor. Este pino possui uma tensão fixa de 1,23 V e a partir deste valor podem ser estimados valores de resistências para a obtenção do valor de tensão de saída desejado.

Neste caso, foram utilizados os resistores R1 e R2 e a tensão de saída será definida por V_{out} , sendo $V_{out} = 5V$ na Figura 23

$$V_{out} = 1,23 * \frac{2,2 K + 6,8 K}{2,2 K} = 5,03 V$$

Os componentes C2, C6 e R3 são componentes recomendados pelo *datasheet* do L5970D para compensação da frequência utilizada no conversor. D2, L1 e C1 são os componentes discretos pertencentes ao conversor e seus valores são recomendados também pelo *datasheet* do L5970D.

Esta tensão de saída de 5 V será responsável por alimentar o *Raspberry Pi* através do conector J2.

Para a alimentação do driver do motor de passo e também para alimentação do laser do escâner, uma tensão de 3,3 V é utilizada. Para isso, um conversor linear TLV1117-33 (U2 Figura 23.b) é utilizado. Optou-se por um conversor linear devido à sua facilidade de uso e baixo custo. Como a queda de tensão nesta conversão é pequena (5 V para 3,3 V) sua eficiência não será tão baixa. O diodo D3 é recomendado pelo *datasheet* para proteger o regulador caso a entrada deste seja curto-circuitada com o GND. O capacitor C7 é utilizado para a estabilização do regulador sendo recomendado um valor mínimo de 50 μF . O Led D4 servirá apenas como sinalização da energização do circuito e o conector J3 disponibilizará esta tensão para alimentação do laser.

Para o controle do motor é utilizado o driver modelo A4988 (Figura 23.c) fabricado pela Allegro Microsystems. Este driver tem uma grande vantagem de

possuir um tradutor interno que transforma um único pulso gerado em sua entrada STEP, em um micropasso na saída para o motor, com controle de corrente, e sem a necessidade da utilização de tabelas de sequência de fase do motor.

Para o controle lógico do driver, utilizam-se as seguintes entradas que respondem a níveis lógicos de 3,3 V ou 5 V. Como os pinos de saída (GPIO's) do Raspberry Pi possuem 3,3 V, esta será a tensão utilizada e portanto, 0 é a tensão de nível lógico baixo e 3,3 V será a tensão de nível lógico alto.

- **ENABLE:** Habilita ou desabilita todas as saídas. Quando está em nível lógico alto todas as saídas são desabilitadas. As ações no STEP, DIRECTION e outros pinos se mantêm configurados e ativos independentemente do estado do ENABLE;
- **RESET:** Desliga todas as saídas. Todos os pulsos no STEP são ignorados até que o RESET esteja configurado em nível lógico alto;
- **DIRECTION:** Determina o sentido de rotação do motor de passo. Só ocorre a mudança de direção quando o STEP seguinte for para nível lógico alto;
- **STEP:** A transição de nível lógico baixo para alto incrementa uma posição no motor. A taxa de acréscimo é determinado pelo estado das entradas MSx.
- **MSx:** São os pinos responsáveis pela configuração de micro-passos para a saída do motor. São definidos pelas entradas MS1, MS2 e MS3. A Figura 24 mostra as possíveis configurações.

Micro-passo é uma técnica utilizada comumente para aproximar a corrente que passam pelas bobinas do motor de uma senóide, permitindo transições mais suaves e menor ressonância nas partes do motor. Quanto maior a divisão de passos, mais suave ocorre essa aproximação.

A Figura 24, retirada do datasheet do driver A4988 mostra a forma da corrente nas bobinas do motor de passo em função dos passos aplicados na entrada lógica do driver descrevendo a aplicação de micro-passos.

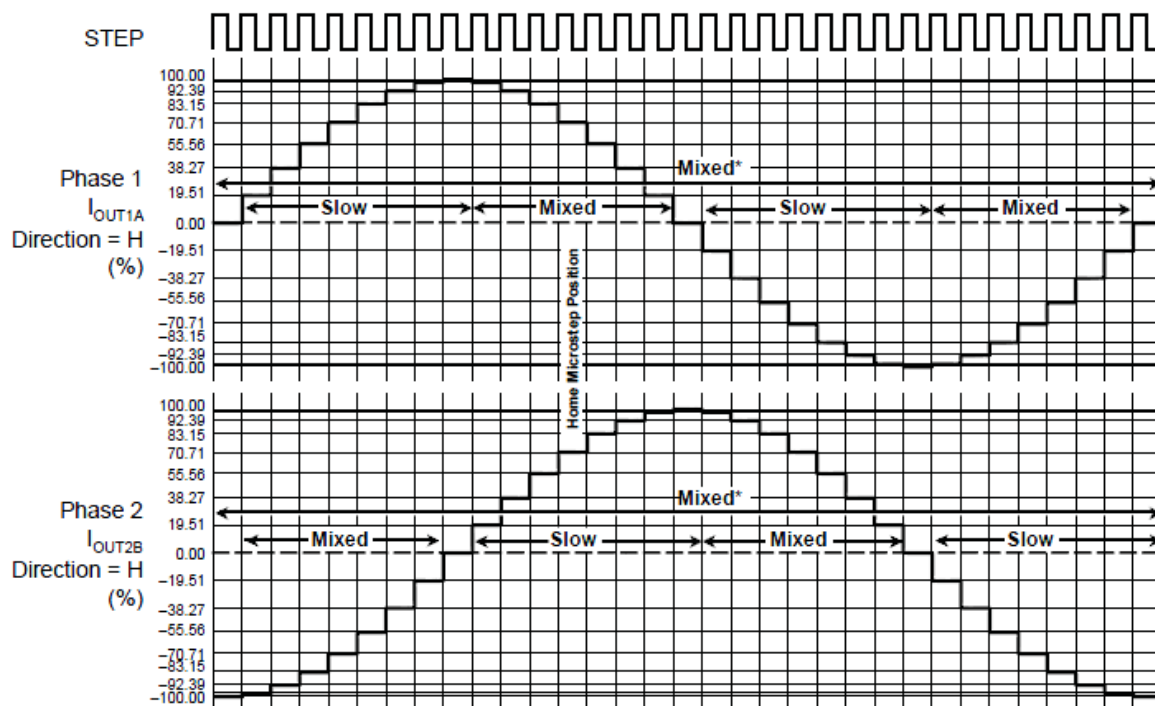


Figura 24. Comportamento da corrente na bobina do motor para uma configuração de 1/8 micro-passos.

Fonte: Datasheet alegra A44988

Para a aplicação do escâner uma configuração de 1/16 passos foi utilizada e os resistores de pull-ups adicionados aos pinos MS1, MS2 e MS3 conforme esquemático e recomendação do Quadro 1.

Quadro 1. Configurações possíveis dos pinos MSx

Fonte: Datasheet alegra A44988

MS1	MS2	MS3	Microstep Resolution
L	L	L	Full Step
H	L	L	Half Step
L	H	L	Quarter Step
H	H	L	Eighth Step
H	H	H	Sixteenth Step

Com esta configuração, o total de passos necessários para uma revolução completa do motor será dado por:

$$N^{\circ} \text{ de passos} = \text{Divisor de micropasso} * \text{Número de passos original do motor}$$

$$N^{\circ} \text{ de passos} = 16 * 200 = 3200 \text{ ppr (passos por revolução)}$$

Considerando a relação de 1:5 das engrenagens do escâner, o número de passos necessários para uma completa revolução da mesa giratória pode ser calculado da seguinte maneira:

$$N^{\circ} \text{ de passos} = 3200 * 5 = 16000 \text{ passos}$$

Portanto, serão necessários 16000 passos aplicados a entrada STEP do driver do motor para se obter uma rotação completa da mesa.

Para o controle de corrente no motor, o conjunto de componentes P1, R10, R13 e R14 é utilizado. Sendo P1 o potenciômetro ajustado de acordo com equação disponível no *datasheet* e permitindo um limite de 1,2 A na saída para o motor que corresponde a corrente nominal do motor.

Para a conexão das entradas e saídas do driver são utilizados os conectores J4 e J5. J4 possui as entradas lógicas provenientes do *Raspberry Pi* para o controle do driver. E o conector J5 será utilizado para a conexão do motor de passo.

Com isso, todas as necessidades do escâner em termos de eletrônica e mecânica estão definidas. Restando agora o software tanto o que será embarcado no *Raspberry Pi* quanto o software para interface no PC.

4.3 SOFTWARE

Inicialmente, cogitou-se a possibilidade de realizar os filtros e cálculos relacionados a triangulação diretamente na placa *Raspberry Pi*. Após alguns testes relacionados a velocidade de execução de algoritmos tanto no *Raspberry Pi* quanto no PC, demonstraram uma velocidade muito superior de execução no PC. Apesar do *Raspberry Pi* ser capaz de realizar os cálculos utilizados no escâner, optou-se pelo processamento no PC para maior velocidade na execução do escaneamento. Assim, ficam definidas a função de cada software da seguinte maneira:

a) Software Raspberry Pi

- Controlar o motor de passo para o posicionamento correto da mesa.
- Acessar a câmera, e salvar as imagens no PC para posterior processamento.
- Interagir com o software no PC atendendo aos comandos de início de escaneamento bem como de parada.

b) Software PC

- Interface visual e acesso aos controles do escâner.
- Acesso às imagens geradas pelo escâner.
- Filtro e processamento das imagens.
- Triangulação.
- Gerenciamento dos arquivos gerados.

Antes da descrição dos mesmos serão descritos alguns métodos e recursos utilizados como requisito e ou parte da criação dos softwares.

4.3.1 Linguagem, Compiladores e IDE's

Todo o software foi escrito em C++, esta opção se deve ao fato de ser a linguagem na qual são disponibilizadas as bibliotecas OpenCV utilizadas no projeto bem como linguagem nativa da aplicação Qt® usada na criação da interface no PC. Além de também, ser uma linguagem a qual os acadêmicos já são bem familiarizados.

Para o software no *Raspberry Pi* foi utilizado a IDE chamada CodeBlocks. Esta é uma IDE simples e que foi possível sua instalação direto no Linux utilizado na placa Raspberry Pi permitindo, inclusive, interface gráfica através do StartX, que é um recurso de terminal gráfico para o Linux. Este software como mencionado, também foi escrito em C++ e compilado com o GCC (GNU Compiler Collection).

Já para o software da interface no PC, foi utilizado Qt®: Uma poderosa aplicação multi-plataforma com suporte a linguagem C++ e que permite a criação de GUI's, ou seja, elementos gráficos para interação com o usuário (janelas, botões, etc).

4.3.2 WiringPi

O acionamento dos pinos do processador no *Raspberry Pi*, quando da utilização de um sistema operacional como o Linux por exemplo, não é direto como no caso de microcontroladores. Para esta função, utilizou-se uma biblioteca gratuita nomeada WiringPi (HENDERSON, 2013). Ela permite que a programação fique mais clara e deixa as configurações de baixo nível dos registradores do microprocessador mais abstratas.

4.3.3 Comunicação Entre Softwares

A solução mais simples encontrada para a comunicação e compartilhamento de informações entre PC e escâner para o teste do protótipo criado, foi à utilização da rede de Ethernet baseada em um sistema de compartilhamento de arquivos.

O PC utilizado possuía sistema do tipo Windows e como este seria o responsável pelo processamento de dados e também conta com maior capacidade de armazenamento, foi criada uma pasta de arquivos compartilhada na raiz da unidade C: do PC com o nome de 'scanner'. Portanto o caminho acessado pelo software no PC é 'C:/scanner/"/>.

Para que o Raspberry Pi pudesse acessar este mesmo diretório com o intuito de armazenar as imagens capturadas pelo scanner, uma aplicação chamada SAMBA que permite uma interoperabilidade entre Linux e Windows possibilitando inclusive este compartilhamento de arquivos.

Foi possível alterar a configuração do sistema Linux de forma que em toda a inicialização, o acesso a esta pasta na rede fosse montado e disponibilizado através do sistema de mídia do Linux. Portanto cada vez que o *Raspiberry Pi*

necessita utilizar o espaço compartilhado, basta acessar o caminho '/media/scanner/' que este terá acesso direto a pasta 'C:/scanner/' no PC.

Isso permitirá que as imagens capturadas através da câmera no *Raspberry Pi* sejam salvas diretamente no PC, agilizando o processo de transferência de arquivos para o processamento.

Além das imagens, é necessária a troca de informações com relação aos comandos a serem executados pelo escâner vindos do PC. Sendo eles basicamente um comando para início de escaneamento e outro para interrupção do escaneamento.

Para solucionar este problema, um arquivo 'conf.cfg' do tipo ASCII, foi criado no diretório compartilhado entre os dispositivos. Assim, o escâner constantemente lê o arquivo à espera de um comando, e o software no PC é responsável por armazenar estes comandos no arquivo.

4.3.4 Software no Raspberry Pi

Com os métodos anteriores, é possível agora descrever como foi escrito o software embarcado no *Raspberry Pi*.

Inicialmente, o software permanece constantemente abrindo o arquivo 'conf.cfg' e aguardando um comando 'Start'. Quando este comando é enviado pelo PC e identificado pelo escâner o software lê a próxima linha do arquivo que contém o número de imagens que serão capturadas no decorrer da revolução completa da mesa. Este número será responsável pelo cálculo da quantidade de passos aplicados ao motor, bem como a numeração das imagens que serão salvas no PC.

Para o cálculo do número de passos utiliza-se o valor total de passos para uma revolução que é de 16000 (conforme calculado no tópico sobre a eletrônica) e divide-se pelo número de imagens. O escâner então executa a movimentação da mesa com a quantidade correta de passos, e salva a imagem no PC através do comando 'raspistill -t 500 -o /media/scanner/img%%%' onde %%% será substituído pelo número da foto atual e terá um intervalo que vai de 0 até o número de imagens passadas pelo arquivo de configuração menos 1. Este processo ocorre

até que todas as imagens sejam salvas ou até que um comando 'Stop' seja recebido e o escâner novamente aguardará por um comando 'Start'.

Antes da aplicação dos passos para rotação da mesa, as saídas do conjunto de GPIO's do Raspberry Pi conectadas aos pinos de ENABLE e RESET do driver do motor são configuradas para 0 e 1 respectivamente. Isso permite com que o driver seja capaz de atuar o motor através dos pulsos na entrada STEP. Após isso, para cada imagem tirada, um número de passos correspondente são enviados para a saída conectada ao pino STEP. Estes pulsos são gerados com uma intervalo de 10 μ s entre si, sendo bastante superiores aos valores mínimos recomendados pelo fabricante do driver possibilitando seu uso sem nenhuma dificuldade e ou perda de passos.

Como nenhum outro tipo de processamento é realizado na placa, o software acabou sendo criado de maneira bastante simples e sem maiores dificuldades.

4.3.5 OpenCV

Para calibração da câmera e tratamentos das imagens foi utilizada uma ferramenta chamada OpenCV.

OpenCV é uma biblioteca multiplataforma de visão computacional, livre para uso educacional e comercial, desenvolvida em C/C++ e possui interfaces para Python, Matlab e outras linguagens. Desenvolvida pela Intel, OpenCV possui vários módulos com diferentes funcionalidades (sub-bibliotecas).

De acordo com Bradski e Kaehler (2008), OpenCV possui mais de 500 funções, as quais abrangem as mais variadas aplicações, incluindo inspeção fabril, imagens de segurança, calibração de câmeras, imagens médicas, vídeo estéreo, obtenção de modelos tridimensionais, produção de nuvens de pontos 3D, reconhecimento facial, classificação de ações humanas e reconhecimento de objetos.

A extração de dados de imagens provenientes de fotos ou vídeos pode ser considerada visão computacional, desde que estes dados sejam utilizados para alguma aplicação (MILANO e HONORATO,2008).

4.3.6 CALIBRAÇÃO DA CÂMERA

O processo de calibração utilizado é realizado em duas etapas e permite a obtenção dos parâmetros intrínsecos e extrínsecos da câmera descritos no Capítulo 3 e necessários na resolução do modelo ótico como requisito para a triangulação do objeto.

A primeira etapa consiste na utilização de um algoritmo fornecido pela OpenCV que calcula os parâmetros intrínsecos da câmera através de um conjunto de imagens que possuem um padrão. Para este tipo de calibração, um plano com um tabuleiro de xadrez pode ser utilizado.

São necessários entrar com os parâmetros referentes a quantidade de linhas e colunas presentes no tabuleiro, bem como a medida de cada quadrado (30x30mm no tabuleiro utilizado). Após isso, é necessária uma quantidade suficiente de imagens para que o processo de calibração seja realizado de forma adequada e apresente maior imunidade a ruído. Foram utilizadas 15 imagens conforme Figura 25.



Figura 255. Imagens para o processo de calibração.

Com as imagens dispostas em posições diversas, o OpenCV é capaz de determinar suas posições e com isso resolver as equações como descritas no tópico referente ao modelo ótico, e então, fornecer os parâmetros intrínsecos.

Estes parâmetros serão salvos em um arquivo do tipo XML pelo próprio software de calibração. Este arquivo conterá os valores referentes a matriz \mathbf{M} e os parâmetros para correção da distorção k_1 , k_2 , k_3 , p_1 e p_2 , que serão posteriormente utilizados na resolução do modelo ótico descrito no Capítulo 3.

Para o cálculo dos parâmetros extrínsecos do escâner R e t , o OpenCV fornece a função **solvePnP** e **solvePnPRansac**, sendo a segunda através de método iterativo e mais resistente a discrepâncias.

A função toma como entrada, os valores dos parâmetros intrínsecos e dois conjuntos de pontos, o primeiro com valores de coordenadas em pixels e o segundo com as coordenadas tridimensionais adotadas como padrão pelo usuário e que possuem correspondência entre si.

Para este processo, definiu-se o sistema de coordenadas utilizados no projeto (Figura 26), com as unidades em milímetros.

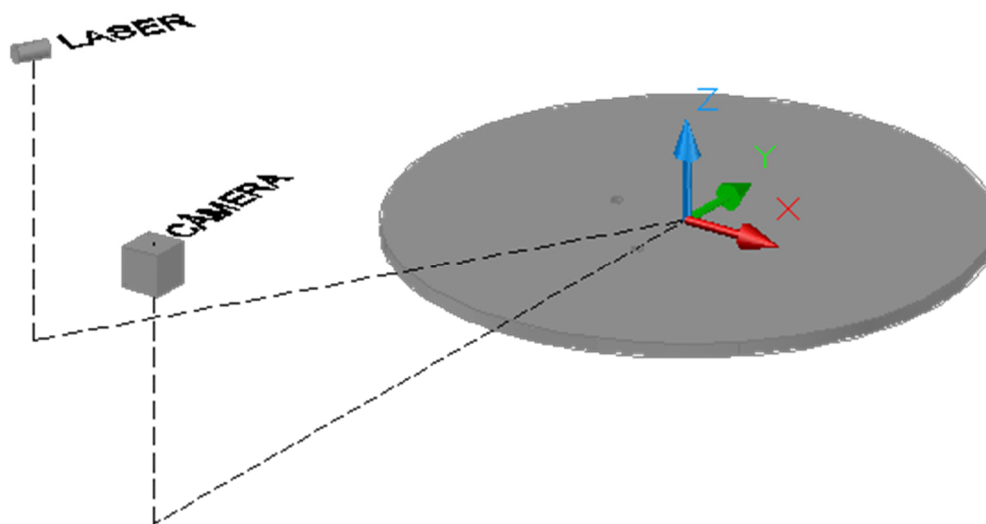


Figura 26. Sistema de coordenadas adotado no projeto com origem no centro da mesa.

O ponto central da mesa possuirá coordenada igual a $(0, 0, 0)$ e os quatro quadrantes da mesa serão dados pelas coordenadas $(-200, 0, 0)$, $(200, 0, 0)$, $(0, -200, 0)$ e $(0, 200, 0)$.

Para a correlação de pontos da imagem e coordenadas reais utilizadas, foi utilizado um objeto padrão com coordenadas reais em milímetros conhecidas (Figura 27).

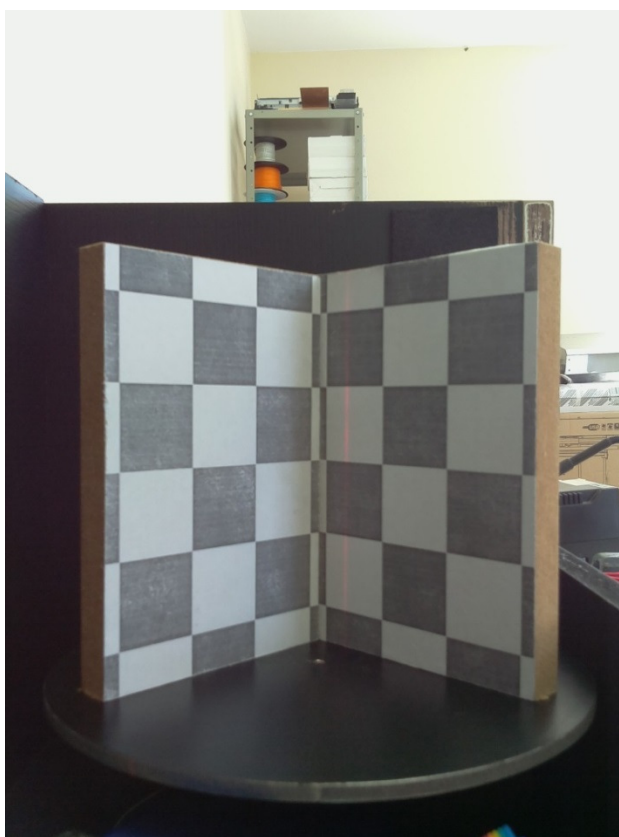


Figura 267. Objeto padrão de calibração

Como todas as bordas da peça utilizada possuem coordenadas reais conhecidas em milímetros, é possível a relação com suas respectivas coordenadas em pixels da imagem. A função retornará os parâmetros R e t que serão utilizados na triangulação.

4.3.8 Software no PC

Abaixo são descritas as etapas criadas e utilizadas no software que disponibiliza a interface no PC para controle do escâner.

4.3.8.1 Interface Visual

Através da utilização das ferramentas QML do QT Studio®, foi gerada uma interface correspondente à Figura 28:

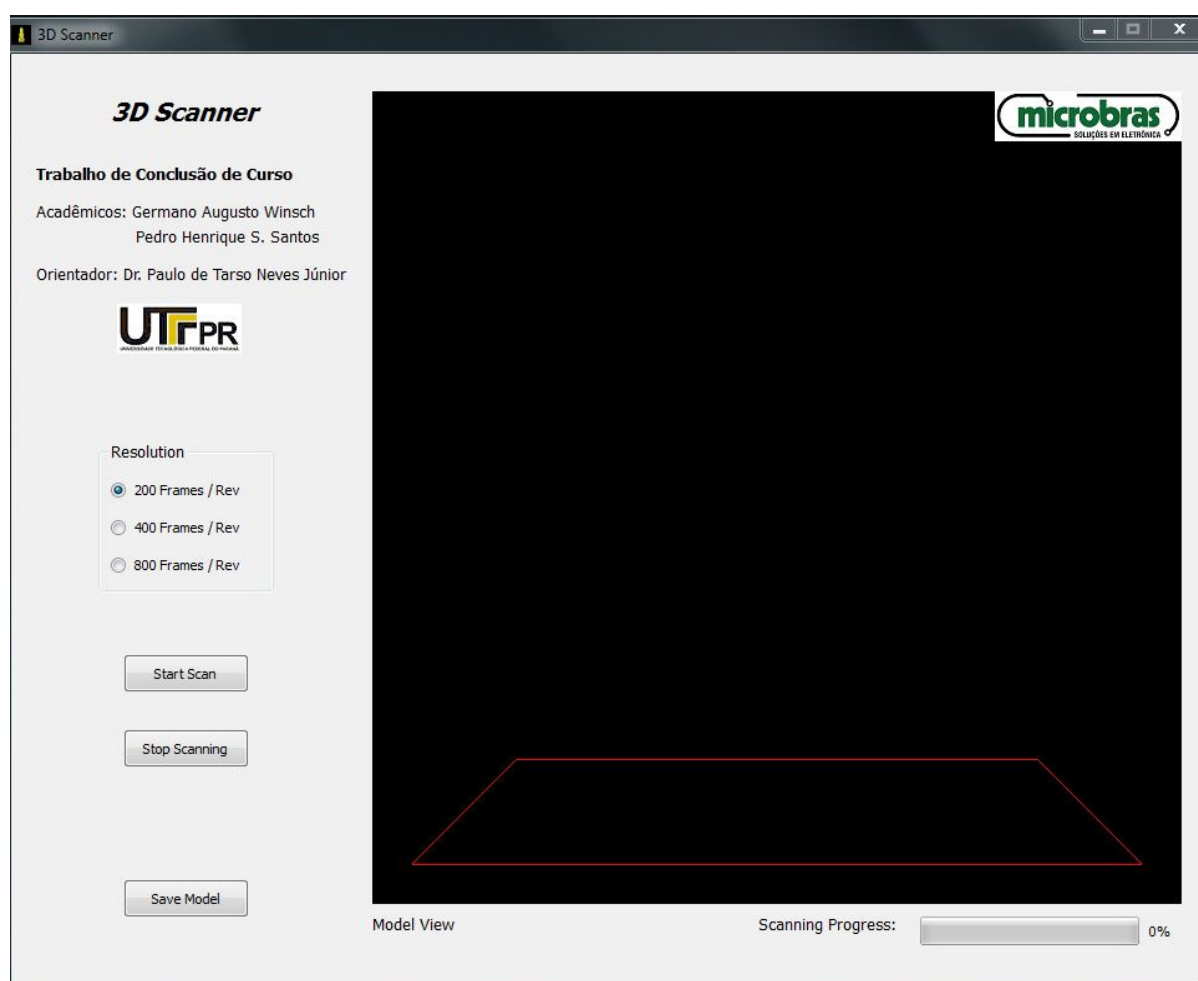


Figura 278. Imagem da interface do software.

Os elementos de controle e visualização são descritos a seguir:

a) Controle de resolução

Permite a seleção do número de imagens que serão capturadas durante o escaneamento da peça.

Foram levados em consideração os testes realizados para se obter uma boa amostragem e tempo de aquisição, além da divisão inteira com relação ao número de passos para cada revolução do motor.

Como são 16.000 passos por revolução da mesa, números como 200, 400 e 800 permitem um número inteiro de passos para cada imagem.

b) Botões

O botão de 'Start' permite o início do escaneamento utilizando a configuração pré-definida de resolução.

Através do arquivo 'conf.cfg' comum entre PC e Scanner ocorre a comunicação através de palavras chaves. Assim, quando pressionado, o botão adiciona uma linha com o comando 'Start' e uma linha com a configuração da resolução '200' (por exemplo). E esta informação será lida e interpretada pelo software no *Raspberry Pi*, que entenderá o início de um escaneamento.

O mesmo ocorre para o botão 'Stop', a informação interromperá o escaneamento no PC e *Raspberry Pi*.

O botão 'Save Model' é utilizado para designar nome e destino do arquivo de nuvem de pontos gerado (*.xyz), permitindo ao usuário gerenciar e organizar seus modelos adquiridos.

c) Visualização do modelo

A janela de visualização permite visualizar os pontos recém-triangulados pelo software do início do escaneamento ao fim. Permitindo o acompanhamento do processo e verificação do desempenho da aquisição para a peça alvo.

Para esta visualização se tornar possível, foram necessários a compreensão e implementação de uma aplicação baseada em OpenGL.

“OpenGL é uma interface de programação de aplicação (API) que é meramente uma biblioteca que permite o acesso às funcionalidades do hardware gráfico. O OpenGL em sua versão 4.3, contém mais de 500 comandos distintos que permitem que você especifique objetos, imagens e operações necessárias para produzir aplicações gráficas computacionais iterativas” (Shreiner, et al., 2013)

Foi utilizada uma janela com suporte para OpenGL integrada ao Qt Studio. Esta permite gerar um espaço com projeção perspectiva que converte os pontos triangulados oriundos do escaneamento em pontos bidimensionais que tornam possível uma simulação da visualização real do objeto.

A Figura 29 mostra a janela, com um modelo escaneado e completamente representado através do OpenGL:

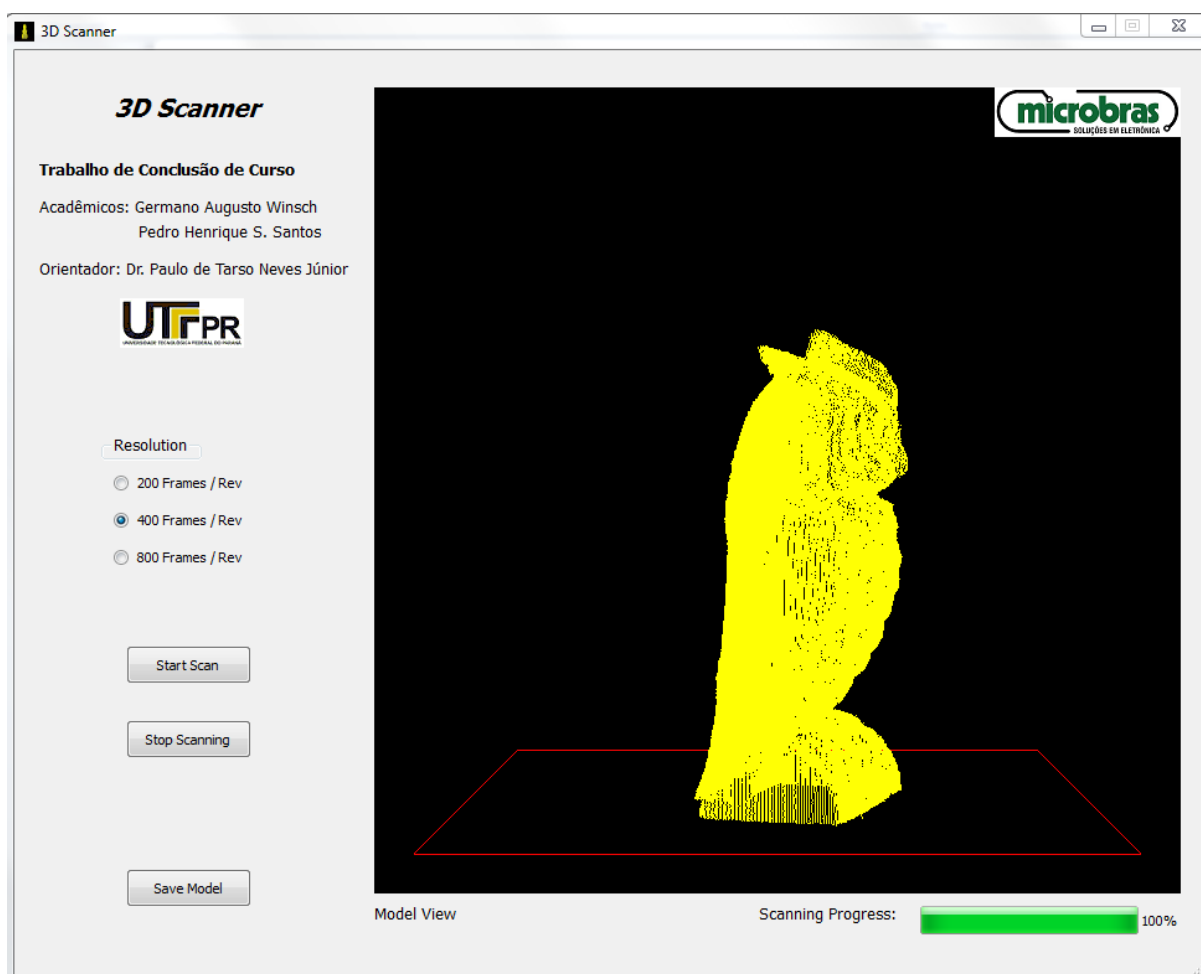


Figura 29. Modelo completamente escaneado.

Além da visualização frontal, é possível rotacionar a janela com o mouse a fim de posicionar a câmera virtual, possibilitando a análise dos diversos ângulos do modelo.

4.3.8.2 Acesso às Imagens

Cada imagem capturada pelo scanner é salva em um diretório específico no PC em que o scanner está conectado, conforme descrito no item referente à comunicação entre os softwares. Cada imagem tem uma numeração em sequência adicionada ao nome do arquivo para permitir a identificação da rotação da mesma com relação à mesa do scanner.

Após o tratamento e coleta das informações necessárias cada imagem é removida. Isto se deve, especialmente, ao fato de cada uma delas possuir um tamanho de aproximadamente 2,7 MB, o que num total de 200, 400 e ou 800 imagens, representam um grande espaço em disco.

4.3.8.3 Filtro e Processamento

Cada imagem capturada no scanner deve ser filtrada e processada de forma a se isolar apenas os pontos correspondentes ao laser projetado na imagem.

A Figura 30, representa parte de uma foto capturada do scanner sem nenhum tipo de tratamento:

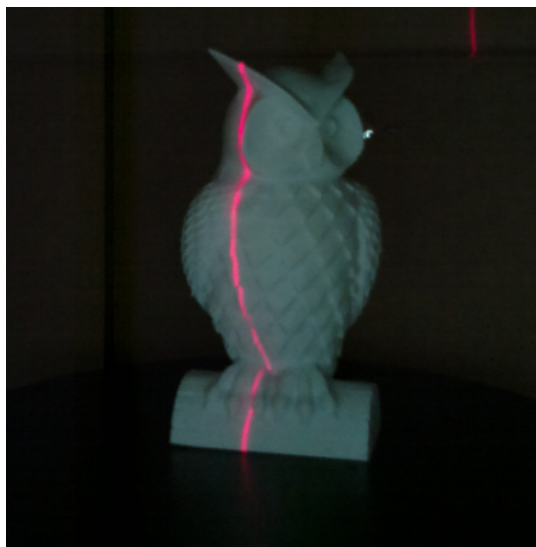


Figura 30. Captura de imagem durante escaneamento.

Utilizando funções disponibilizadas pelo OpenCV, a imagem é aberta no software, passa pelo processo de correção das distorções, utilizando a função **undistort** presente no OpenCV (que recebe como parâmetros os coeficientes k_1 , k_2 , k_3 , p_1 e p_2 obtidos na calibração) e então é convertida de RGB para modo de cores HSV.

As diferenças entre os dois espaços de cores pode ser verificada através da Figura 31:

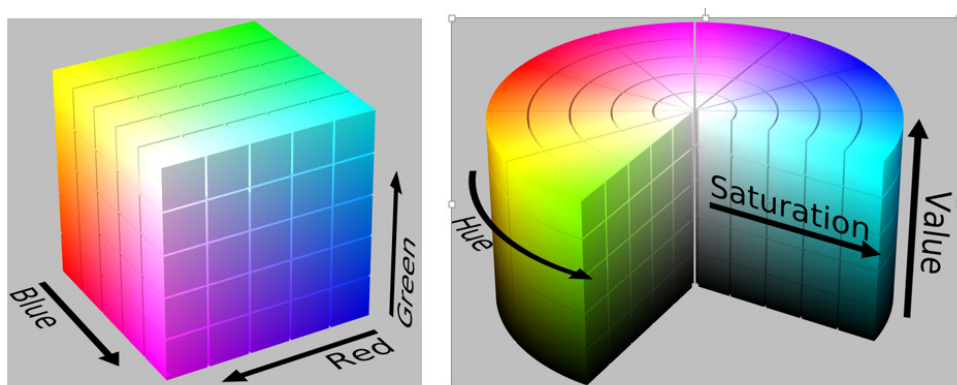


Figura 281. Comparação dos espaços de cores.

A motivação para a utilização do HSV é a facilidade para lidar com limites para filtros, pois seus parâmetros são muito mais intuitivos na utilização do que o padrão RGB.

Hue basicamente é a definição da cor desejada. *Saturation* define a intensidade de cor relacionada ao seu próprio brilho. E *Value* representa o brilho da cor, escura, ou clara.

Baseados nestes princípios, um filtro para o range da cor do laser foi criado. O mesmo converte a imagem capturada pela câmera em uma imagem contendo 1 em toda região de interesse (laser) e 0 nos demais pixels. A imagem a seguir representa este filtro aplicado à Figura 30



Figura 292. Aplicação dos filtros na Figura 30

A próxima etapa filtra a imagem anterior, deixando apenas uma linha central. Utilizando um simples algoritmo que pega o ponto central do laser para cada linha da imagem, tendo-se o resultado mostrado na Figura 33.



Figura 33. Detecção da linha de pontos partindo da Figura 32

Esta é a imagem filtrada final, que conterà os pontos utilizados para a triangulação, sendo este processo repetido para todas as imagens durante o escaneamento.

A imagem é armazenada na memória do software através de uma matriz binária que permite a indexação e localização de cada pixel e seu respectivo valor.

4.3.8.4 Triangulação

Depois de definidos os pontos de interesse e sua localização na imagem, é necessária a conversão destes para pontos tridimensionais correspondentes ao modelo alvo do escaneamento.

Para isto, inicialmente utiliza-se a aplicação do modelo ótico descrito pela equação 3.2.

Os parâmetros $M, R e t$ são conhecidos pois foram obtidos através do processo de calibração.

Os valores de (u, v) são os valores das coordenadas em pixels que contém os pontos do laser, conforme a Figura 33.

Para o cálculo de (X, Y, Z) resta saber o valor do fator de escalonamento λ , mas como uma reta pode ser definida por dois pontos, bastando que estes pertençam à própria reta, o fator de escalonamento será indiferente, pois para qualquer que seja seu valor, o ponto ainda pertencerá à reta que liga a câmera ao ponto que se deseja triangular.

Utiliza-se um fator λ de 1 para simplificar os cálculos e então o ponto chamado de P_1 é calculado (Verificar Figura 10).

O outro ponto pertencente a reta pode ser definido pela posição da câmera, sendo que para isso, utiliza-se a solução particular do modelo ótico para qual as coordenadas da câmera sejam $(0,0,0)^T$. Sendo o resultado, a posição da câmera no sistema de coordenadas reais em milímetros. Este ponto será chamado de P_0 (Verificar Figura 10).

Para a definição do plano do laser são necessários um ponto pertencente ao plano e um vetor normal a este plano. Como o ângulo do laser e sua posição são conhecidos, fica fácil de estimar estes dois valores. Sendo um ponto pertencente a este plano a própria coordenada $(0, 0, 0)$ já que o plano do laser passa pelo centro da mesa.

Como o plano do laser está rotacionado em 30° em relação à câmera, o vetor normal poderá ser definido como:

$$(\cos 30^\circ, \sin 30^\circ, 0) = (0,866025, 0,5, 0)$$

Com estes valores é possível calcular o ponto final triangulado utilizando as equações 3.3 e 3.4.

Este processo deverá ser repetido para cada ponto na imagem referente ao laser. E para todas as imagens escaneadas e filtradas.

Antes de colocar os pontos em um arquivo contendo a nuvem de pontos, é necessário, para cada imagem, rotacionar os pontos calculados para sua posição correspondente a posição atual da mesa.

Isso se deve ao fato de a câmera permanecer estática e o objeto rotacionar junto com a mesa. Então, para cada posição da mesa, aplica-se a equação abaixo, para posicionar os pontos espacialmente de maneira correta.

$$X_{rotacionado} = X * \cos(\theta) - Y * \sin(\theta)$$

$$Y_{rotacionado} = X * \text{sen}(\theta) + Y * \text{cos}(\theta)$$

$$Z_{rotacionado} = Z$$

Sendo θ , o ângulo da mesa correspondente à imagem atual, conforme Figura 34.

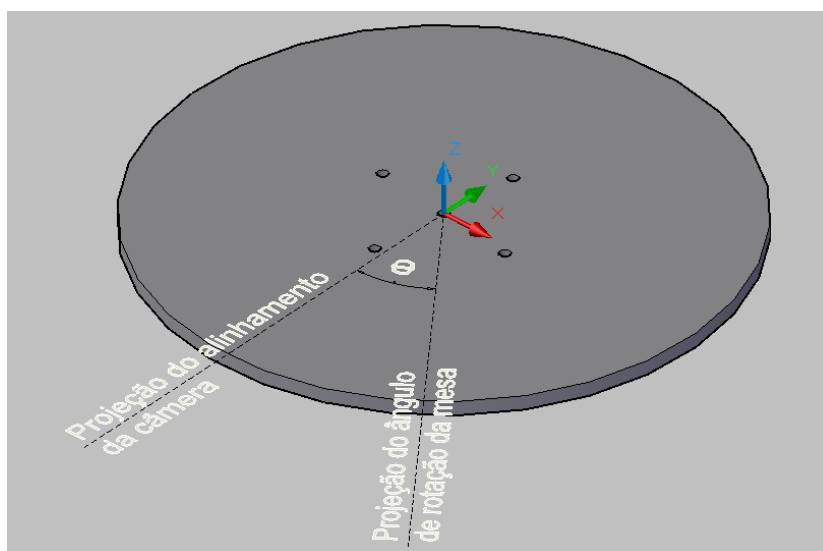


Figura 34. Ângulo de rotação

4.3.8.5 Arquivo Gerado

Todos os pontos calculados pela triangulação, serão adicionados a um arquivo do tipo *.xyz, que permite ser aberto e interpretado pela maioria dos softwares para tratamento de nuvem de pontos.

Este arquivo, nada mais contém, do que todos os pontos triangulados descritos por suas coordenadas (X, Y, Z) separados por vírgulas e cada ponto em uma linha.

O software permitirá ao usuário definir o local e nome do arquivo do modelo gerado para futura edição e ou visualização.

5 RESULTADOS

5.1 PROTÓTIPO

A Figura 35 demonstra o resultado final do protótipo do escâner tridimensional, onde é possível visualizar todos os componentes projetados. Dentre os quais estão a placa auxiliar (Figura 36), o suporte da câmera, o laser, a plataforma giratória, o *Raspberry Pi* e também uma fonte industrial (Figura 37) para realizar a alimentação de 24 V da placa auxiliar.

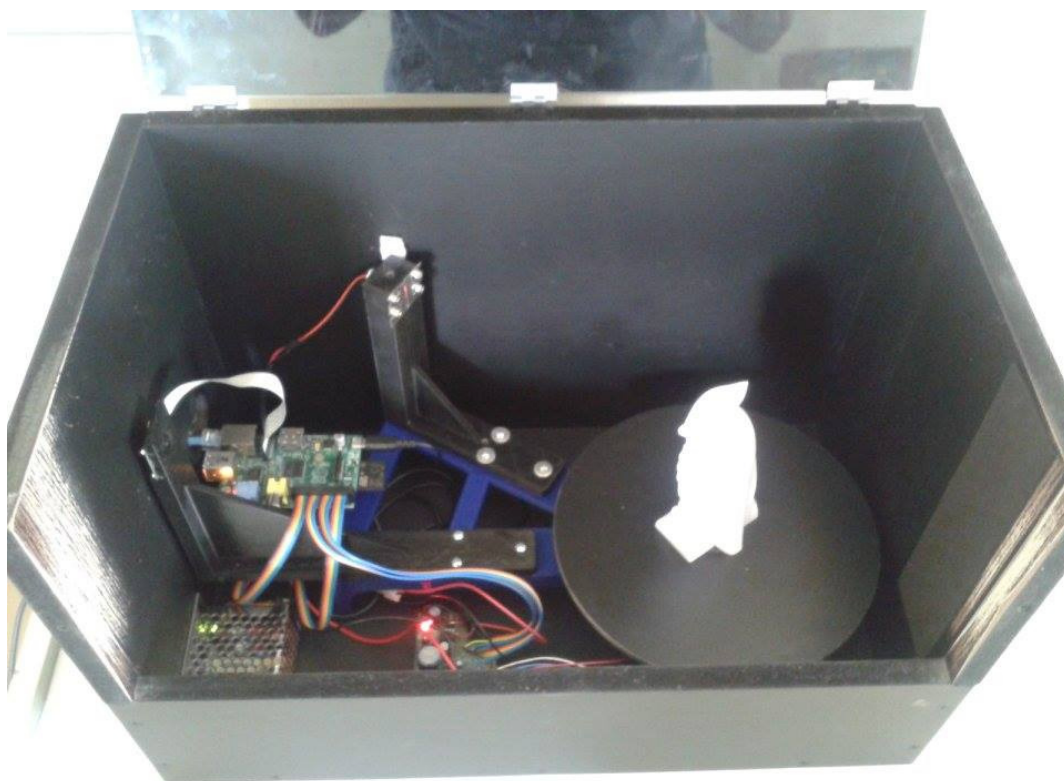


Figura 35. Protótipo final.

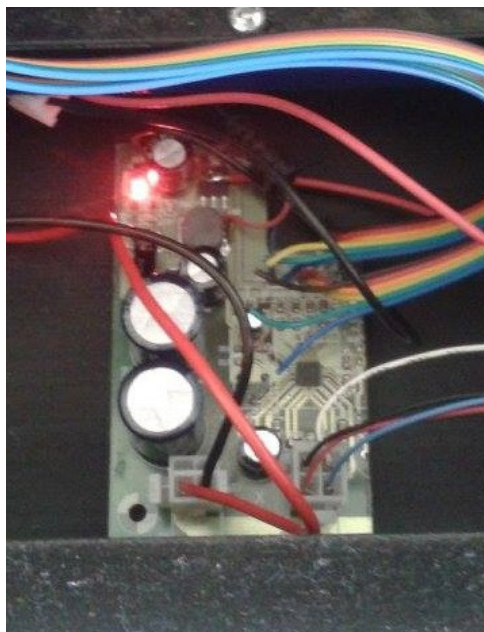


Figura 36. Placa auxiliar.

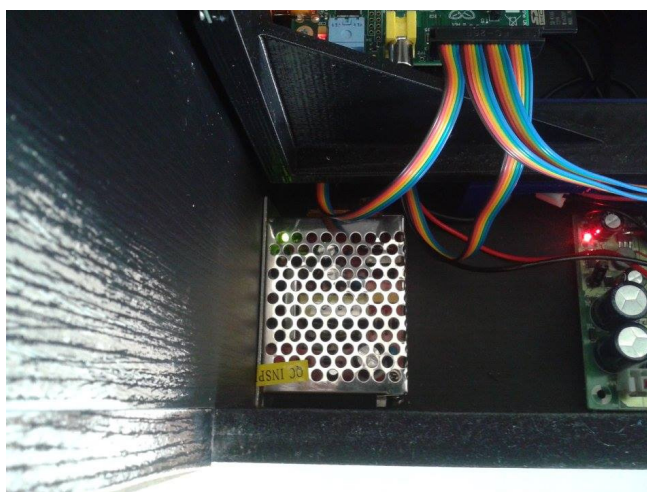


Figura 37. Fonte industrial.

O posicionamento relativo da câmera e do laser foram fielmente executados conforme projetado no tópico 4.1, mantendo o ângulo de 30° entre eles. A exatidão deste ângulo é de extrema importância para a execução correta dos cálculos de posicionamento.

5.2 CALIBRAÇÃO

A calibração foi realizada através da função **SolvePnP**Ransac. A determinação dos pontos de ajuste foi realizada através do objeto padrão, ou calibrador, descrito no item 4.3.6. Após a captura e tratamento da imagem do calibrador, devidamente posicionado sobre a mesa e de frente para a câmera, utilizou-se a função **findChessboardCorners** para obter as coordenadas das extremidades dos quadrados contidos na peça. Esta função foi utilizada duas vezes, uma para cada lateral do calibrador (Figura 38).

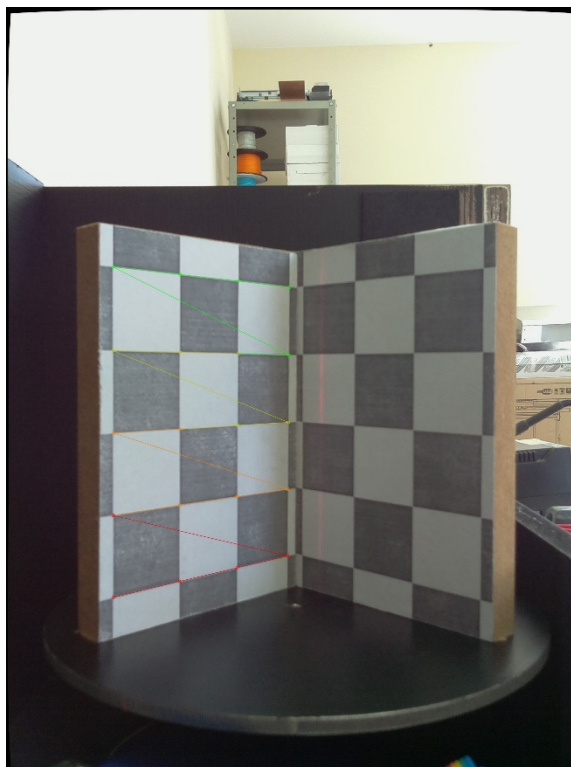


Figura 38. Exemplo da obtenção das coordenadas pela função findChessboardCorners.

Com a posição do calibrador conhecida, as coordenadas reais foram obtidas através de uma reprodução exata deste modelo no software AutoCad® (Figura 39). Isto se fez necessário devido a um erro de décimos de milímetro durante a

aplicação da imagem quadriculada sobre o calibrador, gerando uma pequena rotação nos pontos.

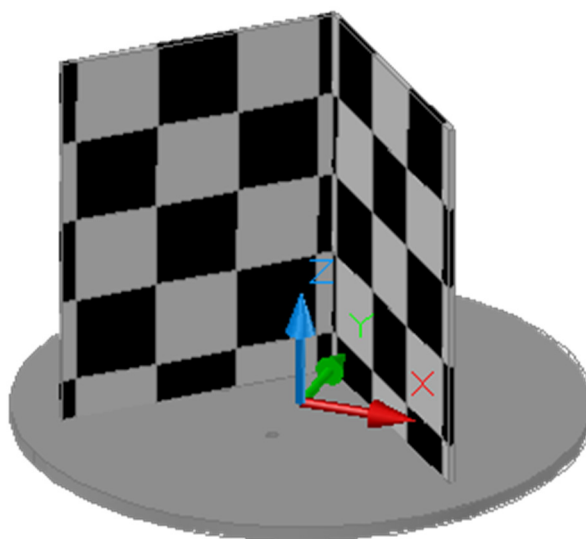


Figura 39. Representação do modelo no software AutoCad®

Com os valores das coordenadas reais e coordenadas em pixels das imagens obtidas, formou-se a Tabela 1 que indica a relação entre estes pontos. Estes valores da tabela foram fornecidos para a função **SolvePnpRansac**, que retorna os valores de **R** e **t**.

Tabela 1. Relação entre as coordenadas reais e as coordenadas da imagem.

Coordenadas (pixel)	Coordenadas (Reais)
	Lado 1
(361.66174, 1997.6805)	(-68.2412, -15.8138, 0.00)
(586.15729, 1943.2151)	(-46.2753, -17.9911, 14.4762)
(780.50568, 1896.9291)	(-25.0621, 3.2221, 14.4413)
(951.05212, 1857.3473)	(-4.5962, 47.6569, 0.00)
(362.12973, 1719.9097)	(-67.4638, -39.1796, 44.5111)
(586.85114, 1685.0166)	(-46.2506, -17.9664, 44.4762)
(781.66565, 1655.8124)	(-25.0375, 3.2468, 44.4412)
(952.49872, 1631.025)	(-3.8243, 24.4600, 44.4063)
(361.98273, 1441.4432)	(-67.4391, -39.1549, 74.5110)
(587.5199, 1427.5433)	(-46.2260, -17.9417, 74.4761)
(782.97083, 1415.0903)	(-25.0128, 3.2715, 74.4412)
(954.18469, 1404.9281)	(-3.7996, 24.4847, 74.4063)

(361.24481, 1162.4148)	(-67.4145, -39.1302, 104.5110)
(587.58154, 1168.1305)	(-46.2013, -17.9170, 104.4761)
(784.22253, 1173.2308)	(-24.9881, 3.2962, 104.4412)
(955.49481, 1177.5558)	(-3.7749, 24.5094, 104.4063)
(361.2757, 878.54547)	(-67.3898, -39.1055, 134.5110)
(587.72015, 904.64795)	(-46.1766, -17.8923, 134.4761)
(785.95612, 927.07947)	(-24.9634, 3.3209, 134.4412)
(957.54761, 947.71234)	(-3.7502, 24.5341, 134.4063)
Lado 2	
(1000.6444, 1854.6389)	(2.9813, 25.3030, 14.8330)
(1174.6252, 1897.4529)	(24.1940, 4.0903, 14.6388)
(1174.6252, 1897.4529)	(45.4068, -17.1225, 14.4445)
(1602.8801, 2005.6504)	(66.6195, -38.3353, 14.2503)
(1001.4041, 1629.162)	(3.1186, 25.1657, 44.8324)
(1174.8167, 1656.4465)	(24.3314, 3.9529, 44.6381)
(1374.1989, 1689.9197)	(45.5441, -17.2598, 44.4439)
(1605.7681, 1726.7551)	(66.7569, -38.4726, 44.2497)
(1002.0419, 1403.3536)	(3.2559, 25.0283, 74.8318)
(1176.2426, 1416.7174)	(24.4687, 3.8156, 74.6375)
(1376.4983, 1431.3632)	(45.6815, -17.3972, 74.4433)
(1609.1625, 1448.921)	(66.8942, -38.6100, 74.2490)
(1002.5334, 1175.9663)	(3.3933, 24.8910, 104.8311)
(1177.8308, 1173.4873)	(24.6061, 3.6782, 104.6369)
(1378.3877, 1171.1881)	(45.8188, -17.5345, 104.4426)
(1611.9457, 1167.7747)	(67.0316, -38.7473, 104.2484)
(1004.3041, 945.39044)	(3.5306, 24.7536, 134.8305)
(1180.2227, 927.86047)	(24.7434, 3.5409, 134.6363)
(1380.6415, 906.50201)	(45.9562, -17.6719, 134.4420)
(1614.5494, 883.47034)	(67.1689, -38.8846, 134.2478)

5.3 FORMAÇÃO DAS NUVENS DE PONTOS E MALHAS

Com o intuito de mostrar a aplicação do escâner nos mais variados objetos, realizou-se uma seleção de sete peças com diferentes formas, cores, texturas e materiais. Após a obtenção dos arquivos das nuvens de pontos, todos os objetos foram submetidos ao software Geomagic Studio® para a geração da malha (*mesh*), que é necessária para a reprodução 3D, e auxilia na visualização final do objeto.

O resultado do escaneamento do Objeto 1 (Figura 40.a) foi satisfatório, resultando em uma malha completa sem a necessidade de alterações (Figura 40.c).

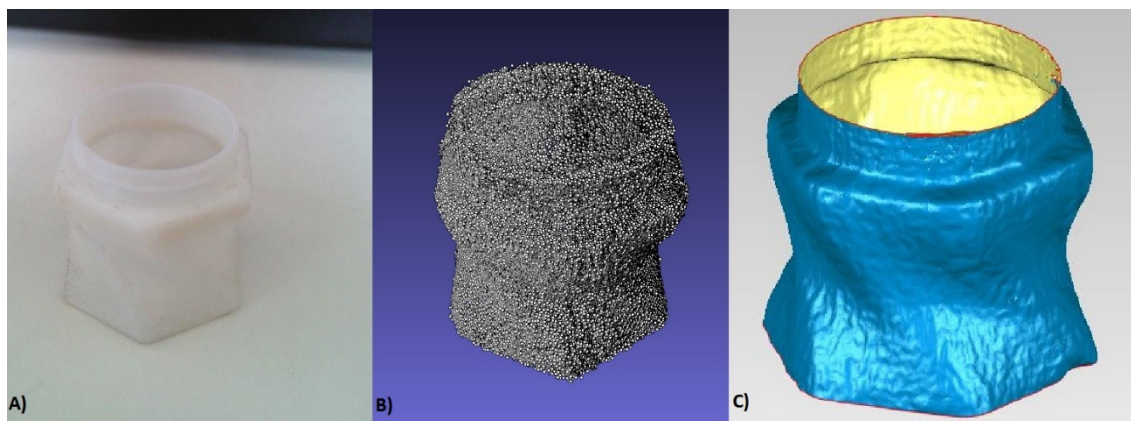


Figura 40 a) Objeto 1 (pote em espiral). b) Nuvem de pontos do objeto 1. c) Malha do Objeto 1

Com o Objeto 2 (Figura 41.a), apesar de sua forma e superfície complexa, foi obtida uma nuvem de pontos (Figura 41.b) bastante satisfatória. Isto possibilitou a geração da malha (Figura 41.c) sem muitas alterações, apenas o preenchimento de pequenos orifícios através do software Geomagic Studio®.

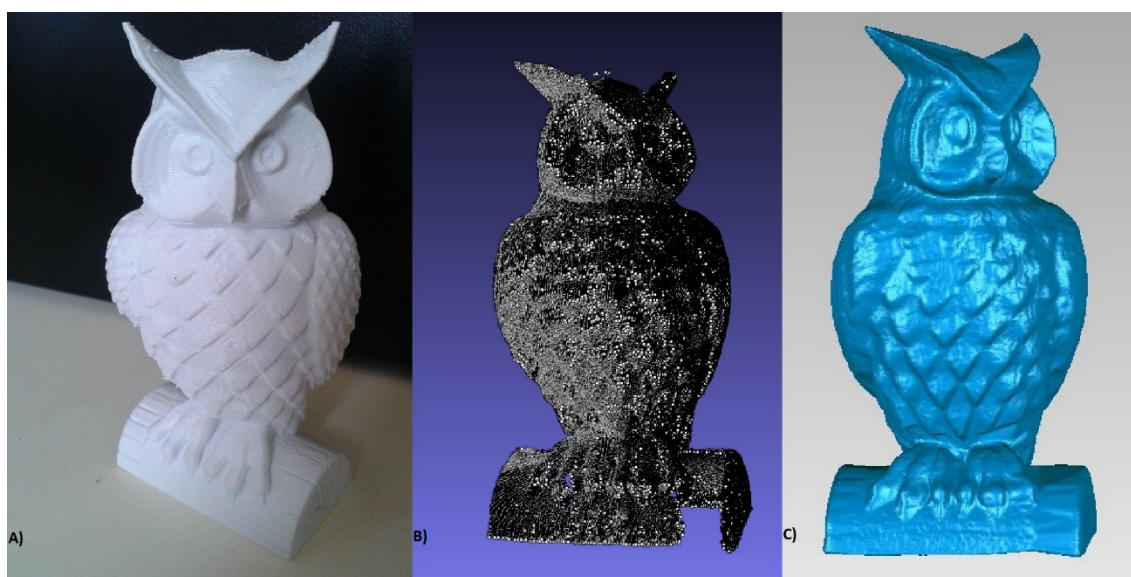


Figura 41. a) Objeto 2, Coruja. b) Nuvem de pontos do objeto 2. c) Malha do objeto 2.

Outra peça que possibilitou a geração de uma nuvem de pontos satisfatória foi o Objeto 3 (Figura 42.a), com o qual foi possível capturar inclusive pequenos relevos na sua região frontal. Por outro lado, a nuvem de pontos apresentou pequenas falhas nas laterais, onde não conseguiu-se a completa geração da malha (Figura 42.c).

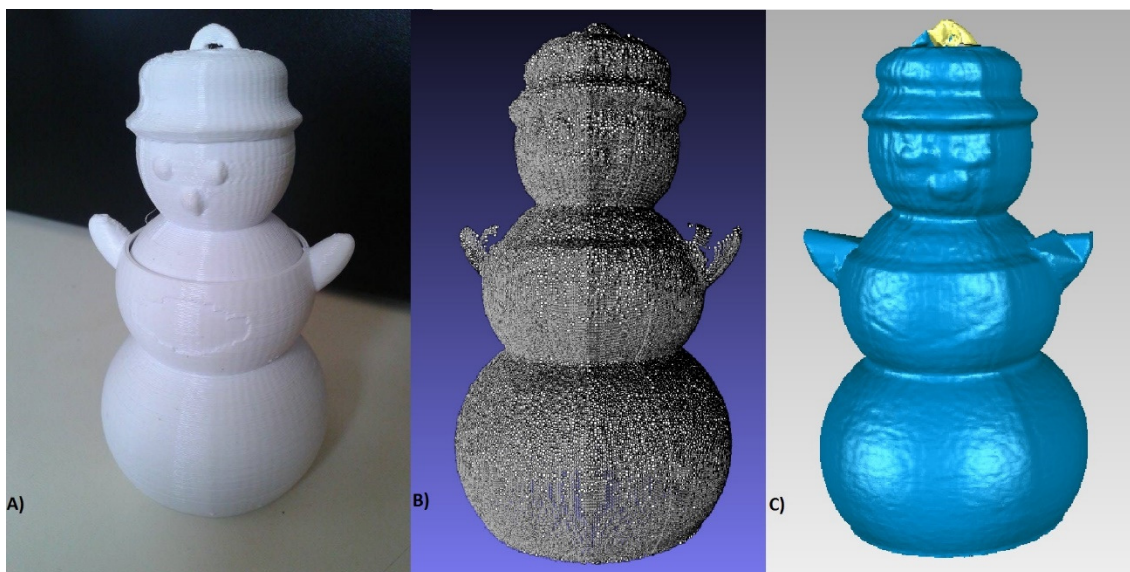


Figura 42. a) Objeto 3, Boneco de neve. b) Nuvem de pontos do objeto 6. c) Malha do objeto 6

5.3.1 Posicionamento do Laser

Algumas das peças escaneadas apresentaram nuvens de pontos com partes sem preenchimento. Isso porque, quanto maior o ângulo utilizado entre o laser e a câmera, mais precisa se torna a detecção e triangulação dos pontos, por detectar pequenas saliências. Por outro lado, isto também aumenta a probabilidade de omissão do laser com relação à câmera pelo próprio objeto.

A Figura 43 demonstra um caso em que o feixe do laser é omitido pelo objeto, ou seja, o mesmo não pode ser detectado pela câmera.

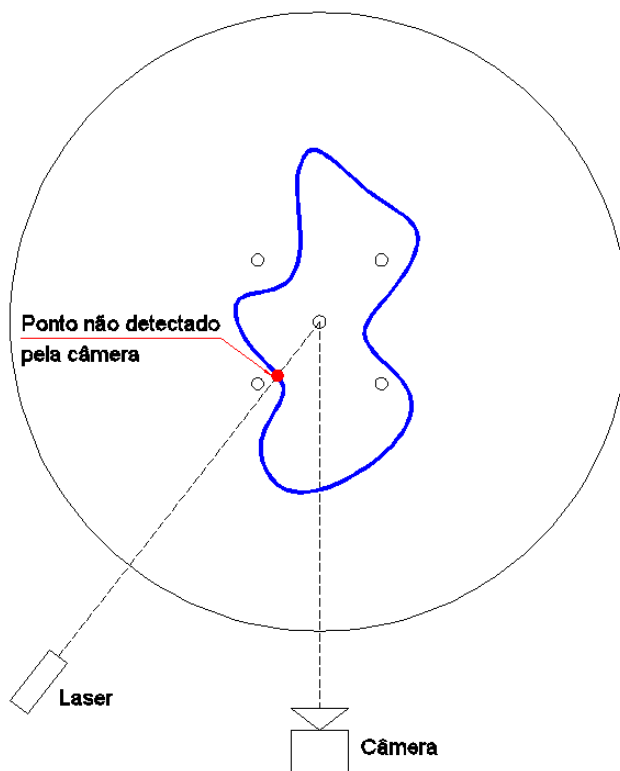


Figura 43. Obstrução do laser pelo próprio objeto

O objeto 4 (Figura 44), mesmo com complexidade maior, apresentou resultados satisfatórios considerando as limitações do laser. Os quatro apoios em relevo e os quatro orifícios (Figura 44.b) foram capturados, sendo que os apoios medem menos de 1,5 mm. Entretanto, devido ao posicionamento do laser e à forma do objeto, em algumas áreas o feixe não ficou visível à câmera, impossibilitando a formação completa da nuvem de pontos (Figura 45).

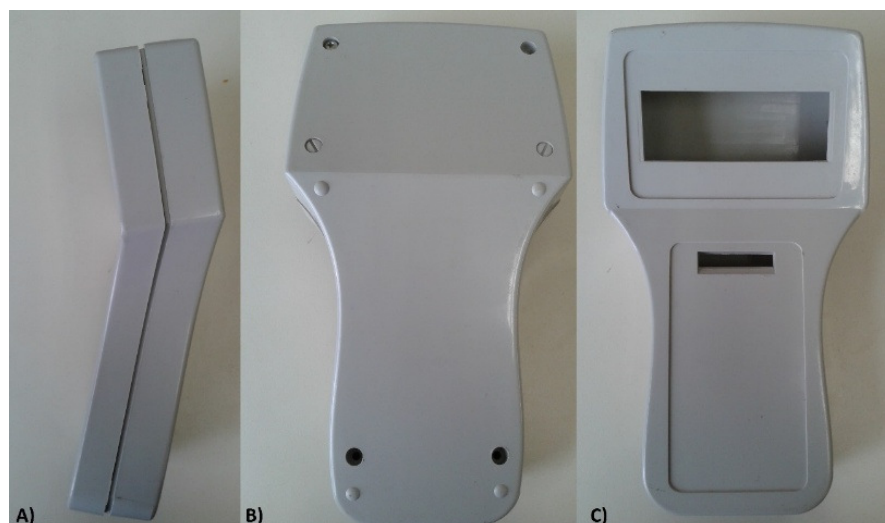


Figura 304. Objeto 4. a) Vista lateral. b) Vista posterior. c) Vista Frontal.

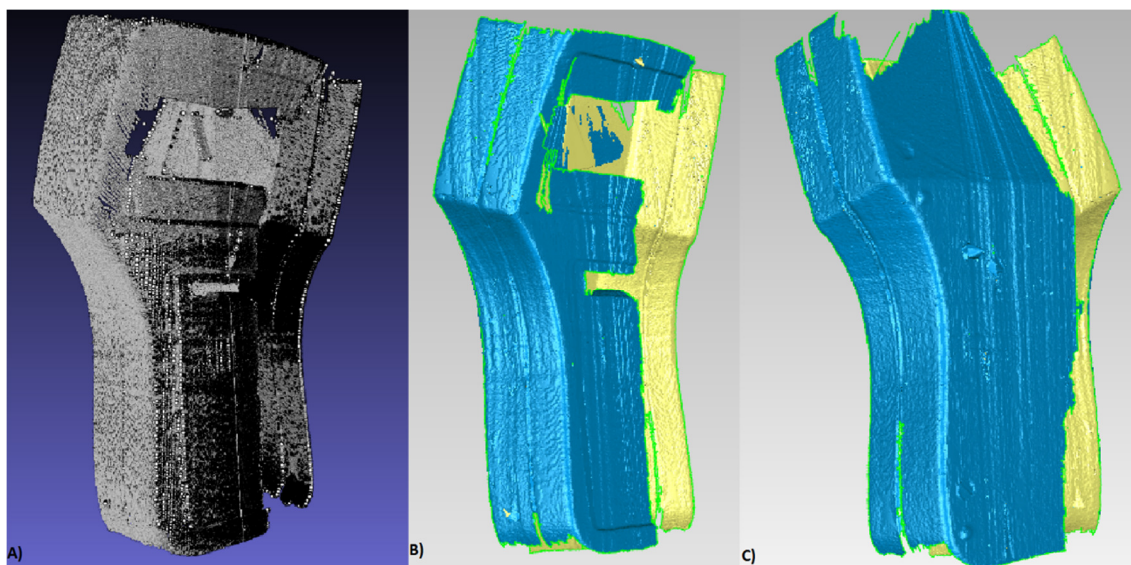


Figura 315. a) Nuvem de pontos do objeto 4. b) Vista lateral da malha 4. c) Vista posterior da malha 4.

O mesmo problema foi verificado no orifício presente na face frontal do Objeto 5 (Figura 46.a) que, devido à obstrução do laser, não permitiu a completa formação da nuvem de pontos.

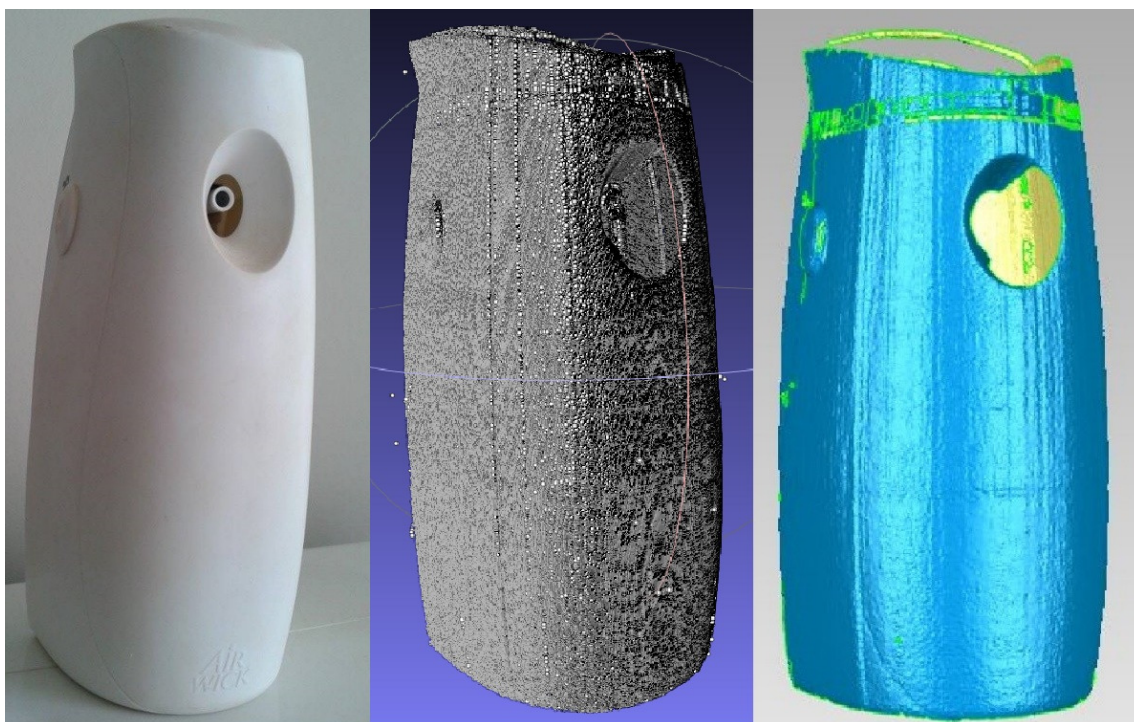


Figura 46. a) Objeto 5 (aromatizador automático de ar). b) Nuvem de pontos do objeto 5. c) Malha do objeto 5.

Este erro foi um dos grandes limitadores da utilização do escâner, podendo, porém, ser facilmente resolvido com a utilização de um laser adicional. Este segundo laser, também posicionado com um ângulo de 30° com relação à câmera, teria acesso ao conjunto de pontos que não ficam visíveis para o primeiro, conforme visto na Figura 47.

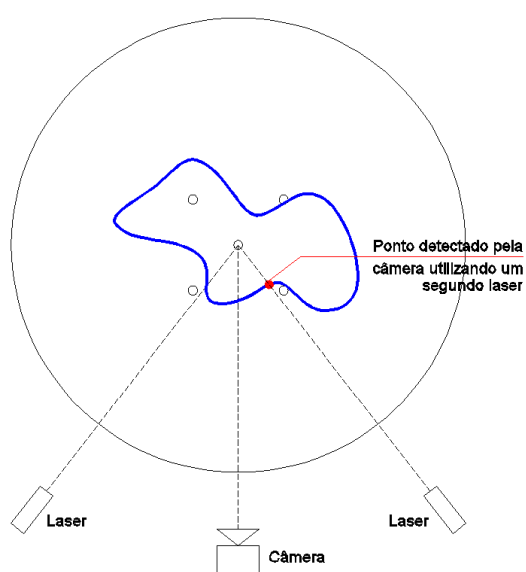


Figura 32. Possível adição de um laser.

5.3.2 FAIXA DE DETECÇÃO DO FILTRO

O filtro de detecção do laser foi desenvolvido para uma faixa de cores bastante específica, o que possibilita o escaneamento de peças com tonalidades próximas à do laser. Por outro lado, o feixe de laser utilizado deve possuir intensidade constante, pois, com pequenas variações, o mesmo não pode ser detectado.

A partir da Figura 48, pode ser comprovado que o filtro detecta uma pequena faixa de cores apenas, pois, se a cor do objeto estivesse contida nesta faixa, além do feixe do laser, a câmera perceberia o objeto todo e o resultado seria uma nuvem de pontos cheia de falhas e ruídos.

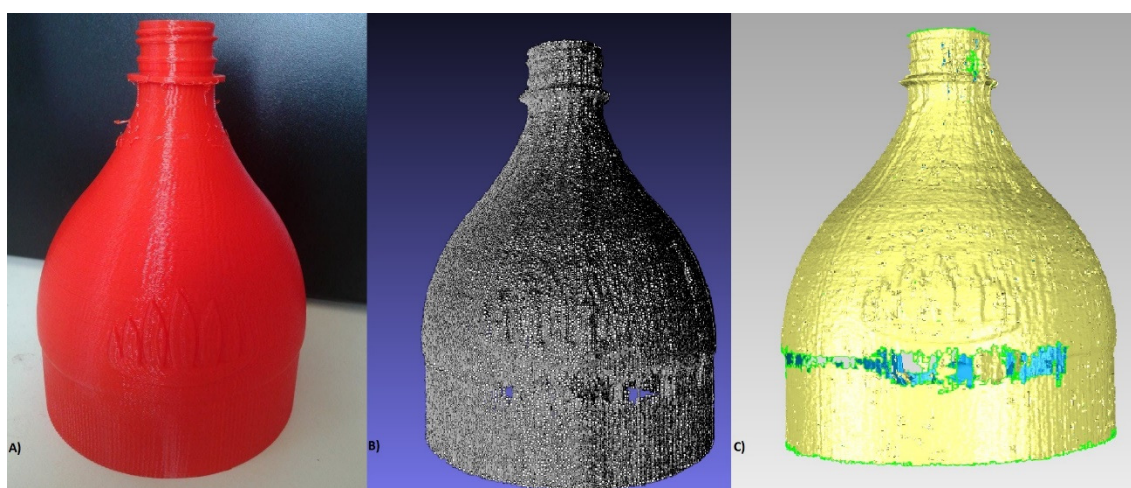


Figura 48. A) Objeto 1 (topo de uma garrafa), com tonalidade próxima à do laser. B) Nuvem de pontos do objeto 1. C) Malha do objeto 1.

Na Figura 49, ao se analisar a alça do objeto, pode ser vista a variação na intensidade do laser, que se deve ao fato de esta possuir uma área pequena com grande curvatura. Isto gera um espalhamento do feixe, impossibilitando sua detecção.

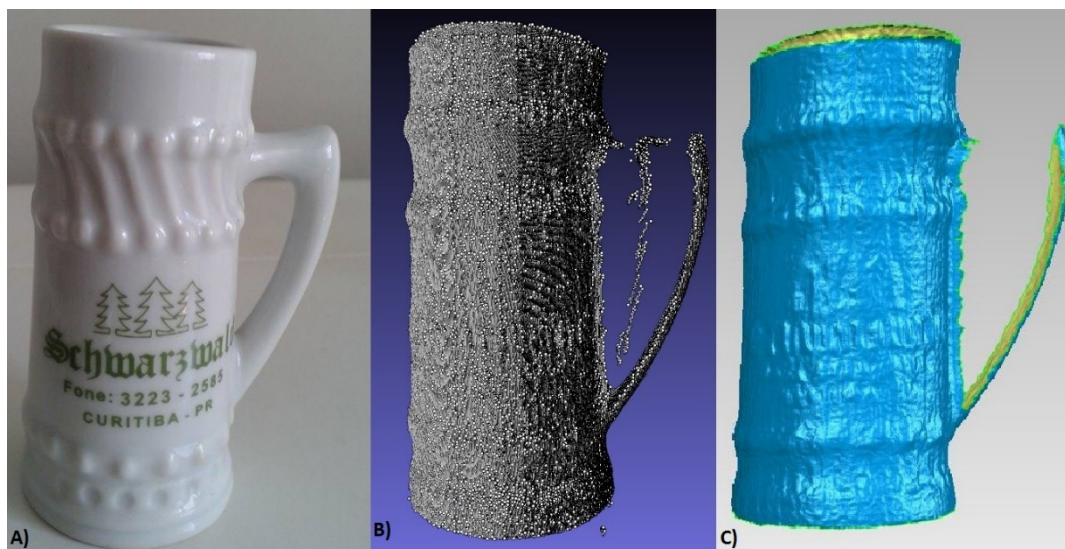


Figura 49. A) Objeto 7, caneca em miniatura. B) Nuvem de pontos do objeto 7. C) Malha do objeto 7.

Além da morfologia da peça, a variação na intensidade do laser também foi verificada pela baixa qualidade do mesmo. Para contornar este problema, é necessário a aquisição de um laser de melhor qualidade, que proporcione um feixe mais definido e homogêneo.

Mesmo com os bons resultados apresentados pelo filtro (alta especificidade), este ainda pode ser melhorado através de um técnica mais robusta com precisão em nível de sub-pixel. Esta técnica consiste na regressão das intensidades do laser para uma determinada linha, a fim de obter coeficientes para uma curva. Isto possibilita a detecção do pico desta curva com uma resolução superior à da imagem utilizada como base.

5.4 ANÁLISES DE ERROS

5.4.1 Plano do Laser

Para efetuar os cálculos da triangulação é considerado que o plano do laser é ortogonal ao plano da mesa giratória. Caso contrário, se os mesmos não apresentarem ortogonalidade, existe a possibilidade da criação de dois tipos de

deformações, o estreitamento no topo e a expansão no topo. A Figura 50 ilustra as três possibilidades de posicionamento do laser com relação à mesa, sendo que o Caso 1 é o ideal.

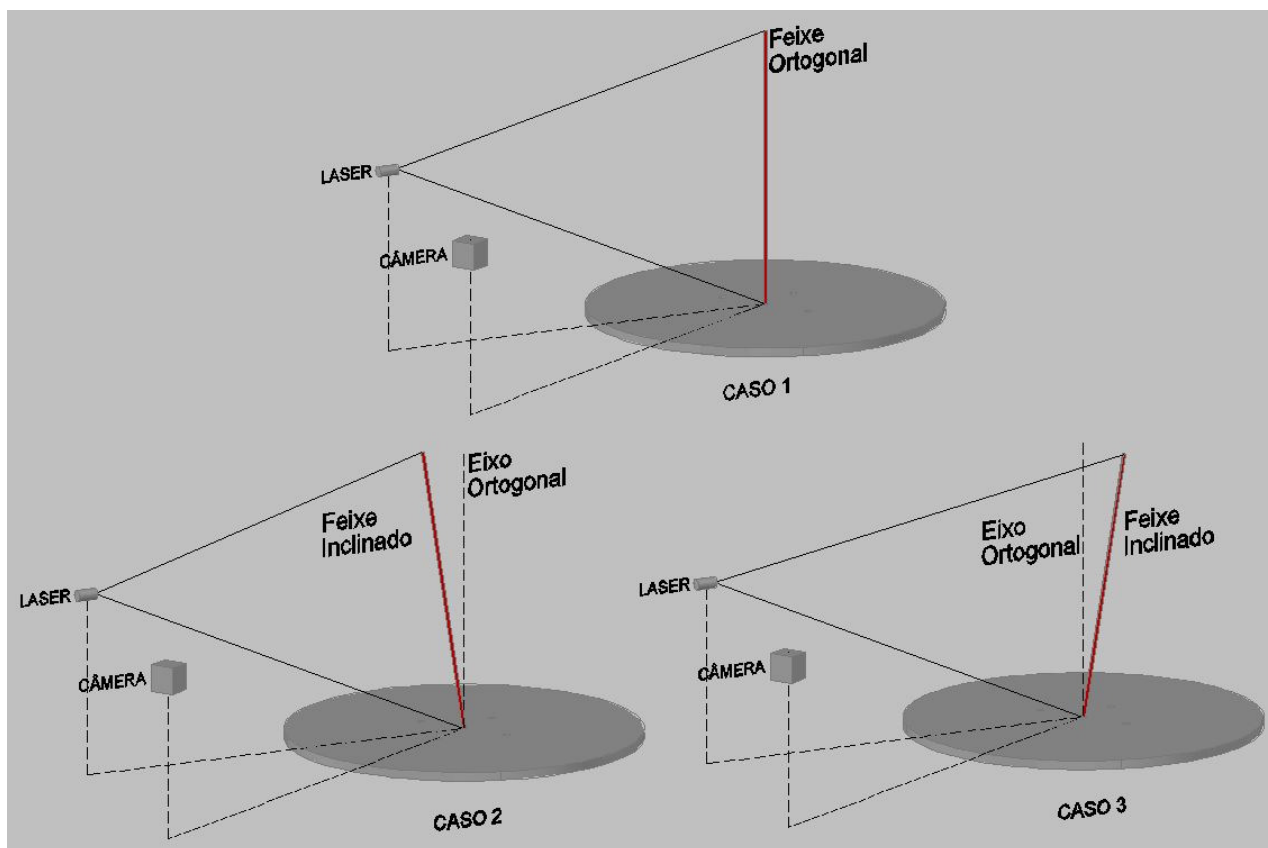


Figura 50. Problemas relacionados com a posição do plano do laser.

Para os Casos 2 e 3, demonstrados na Figura 50, quanto mais próximo do topo do objeto, maior a diferença entre os pontos na linha ortogonal e os pontos atingidos pelo laser.

Considerando o Caso 2, quando ocorre a rotação dos pontos, devido à movimentação da mesa, os mesmos são triangulados com maior afastamento do eixo Z, gerando uma expansão naquela área (Figura 51). Já para o Caso 3, os pontos são triangulados com menor afastamento do eixo Z, gerando um estreitamento na área (Figura 52), e um vórtex no topo (Figura 53).

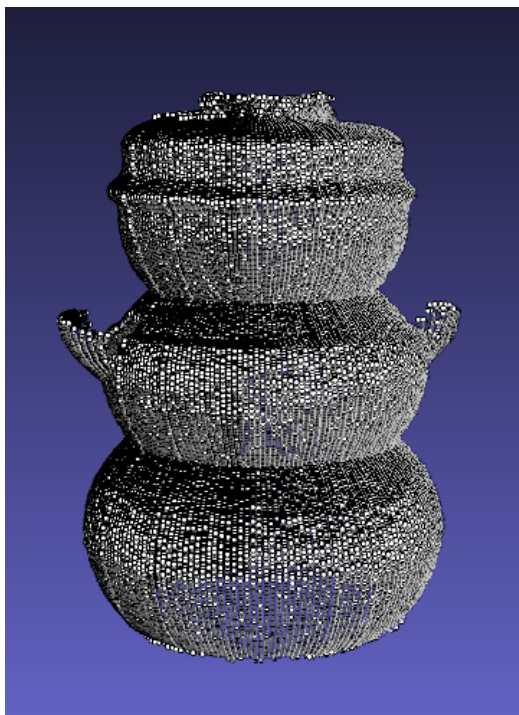


Figura 331. Laser não ortogonal ao plano da mesa.

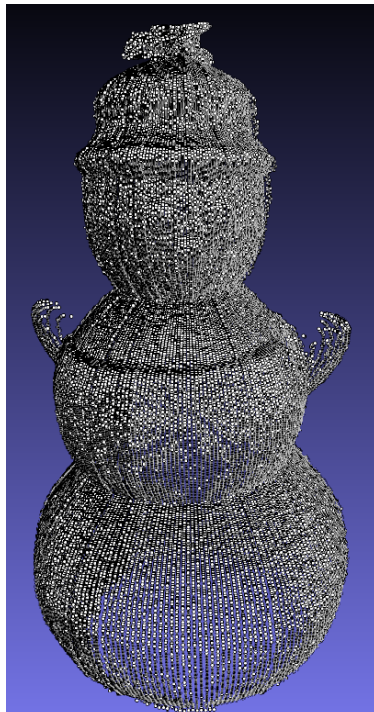


Figura 342. Exemplo de estreitamento no topo.

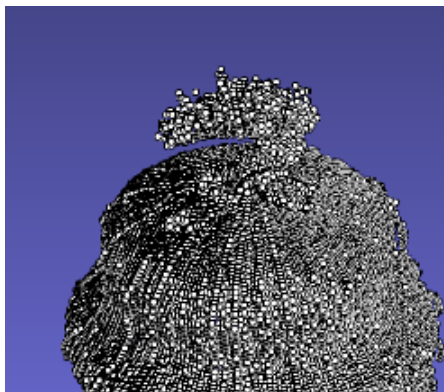


Figura 353. Vortex da Figura 52.

5.4.2 DIMENSIONAL

Após a escolha da melhor combinação dos pontos de calibração (Tabela 1), foram realizadas as análises dimensionais, apresentando a porcentagem de erro e a diferença em milímetros entre as medidas reais do objeto e as medidas obtidas pelo escaneamento (Tabela 2).

As Figuras 54 e 55 exemplificam como foram realizadas as medições referentes aos objetos após o escaneamento, na Tabela 2. Para uma maior confiabilidade das mesmas, foram escolhidos pontos que facilitam a medida, como por exemplo a base do objeto 4. As medidas de importância nas imagens são as distâncias entre os pontos 1 e 2 das imagens, os outros valores são utilizados pelo software, não possuindo relevância para este estudo.

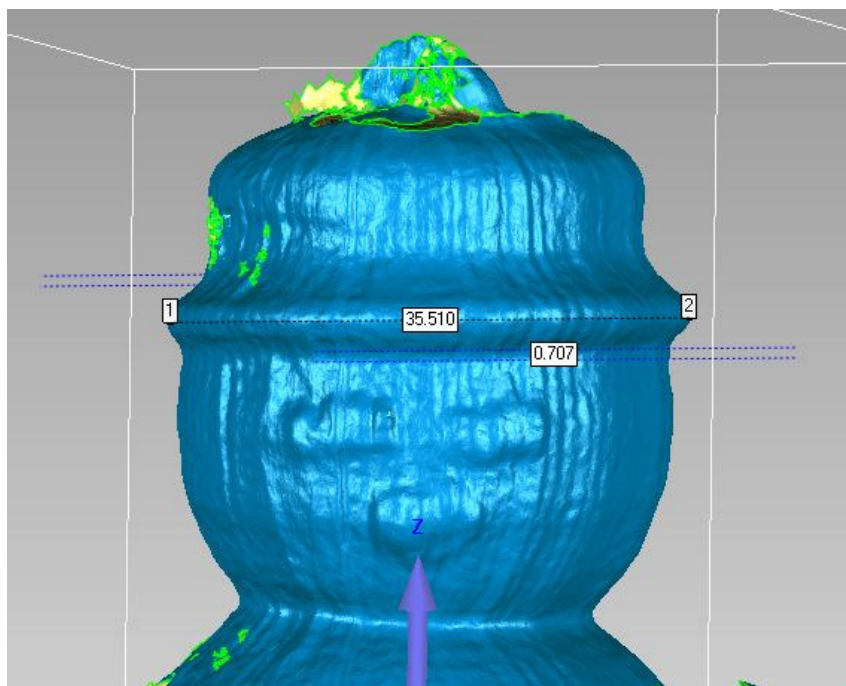


Figura 364. Medida do diâmetro do Objeto 3

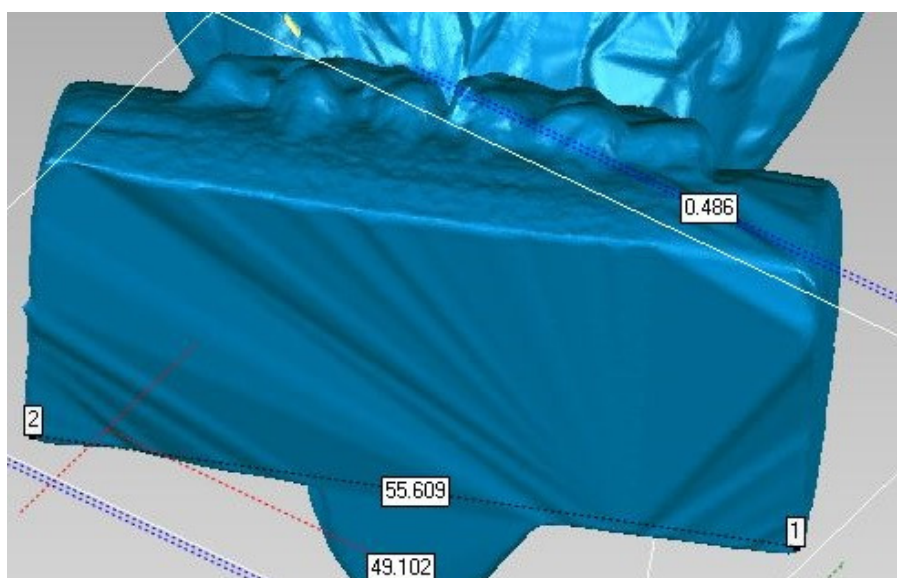


Figura 375. Medida da base do Objeto 2

O mesmo procedimento foi realizado para a medição de todos os objetos estudados, e os resultados estão apresentados na Tabela 2.

Tabela 2. Comparação entre as medidas reais dos objetos e as medidas após o escaneamento.

Objeto	Medida Real (mm)	Medida escaner (mm)	Erro (%)	Diferença
Objeto 1	34,920	33,547	-3,932	1,373
Objeto 1	31,600	31,950	1,108	0,35
Objeto 2	205,000	206,253	0,611	1,253
Objeto 2	23,360	22,864	-2,123	0,496
Objeto 2	29,900	30,916	3,398	1,016
Objeto 2	39,720	39,691	-0,073	0,029
Objeto 2	14,760	14,778	0,122	0,018
Objeto 3	47,200	47,539	0,718	0,339
Objeto 3	36,480	37,382	2,473	0,902
Objeto 3	37,000	37,239	0,646	0,239
Objeto 3	23,000	24,423	6,187	1,423
Objeto 3	23,000	23,232	1,009	0,232
Objeto 3	35,500	35,510	0,028	0,01
Objeto 3	108,580	110,144	1,440	1,564
Objeto 4	38,000	36,608	-3,663	1,392
Objeto 4	198,000	196,206	-0,906	1,794
Objeto 4	6,300	6,940	10,159	0,64
Objeto 4	38,000	39,212	3,189	1,212
Objeto 5	44,400	43,604	-1,793	0,796
Objeto 5	56,300	55,609	-1,227	0,691
Objeto 5	32,940	31,769	-3,555	1,171
Objeto 6	101,500	102,320	0,808	0,82
Objeto 6	24,520	24,760	0,979	0,24
Objeto 6	142,500	143,770	0,891	1,27
Objeto 7	33,500	32,752	-2,233	0,748
Objeto 7	33,500	32,861	-1,907	0,639
Objeto 7	69,300	68,363	-1,352	0,937

A Tabela 2 mostra que não existe relação entre as medidas da peça e as medidas do erro. Em uma mesma peça, os valores do erro não se apresentaram constantes, variando tanto para mais quanto para menos com relação à medida original.

A maior porcentagem de erro foi de 10,16%, e a menor porcentagem foi de 0,07%, apresentando uma média de 2,09%. Além disso, 59% das medidas realizadas apresentaram uma porcentagem de erro menor que 2%.

Apesar de os erros encontrados serem relativamente baixo, esses valores ainda podem ser melhorados utilizando-se um objeto de calibração ainda mais preciso, e que permita uma calibração automática para sempre manter o escâner com parâmetros atualizados no decorrer de sua utilização.

6. CONCLUSÃO

Para competir com modelos de escâneres 3D presentes no mercado ainda são necessários aperfeiçoamentos para obtenção de modelos mais precisos. Entretanto, o custo do protótipo foi relativamente baixo, podendo ser considerado um escâner de baixo custo.

Os modelos obtidos no formato nuvens de pontos possuem a mesma forma dos objetos originais, porém, devido à falta de alguns pontos, ocorreram pequenas alterações na geração do modelo de malha quando comparados aos objetos originais.

O software desenvolvido possui uma interface clara e amigável, facilitando seu uso para qualquer pessoa, até mesmo leigos na área da informática.

Os resultados obtidos foram satisfatórios devido ao tempo de desenvolvimento, pois viabilizaram o entendimento do processo e conceitos envolvidos no projeto possibilitando análises de futuras alterações para que este se torne um produto competitivo dentre os existentes no mercado.

Ademais, os softwares também precisam de melhoria em questões de robustez e desempenho, permitindo um sistema mais seguro e mais eficiente no escaneamento.

7. REFERÊNCIAS

BERNARDINI, Fausto; RUSHMEIER, Holly. **The 3D Model Acquisition Pipeline**, The Eurographics Association and Blackwell Publishers Ltd, v. 21, p. 149-172, 2002.

POLLEFEYS, M.; KOCH, R.; VERGAUWEN, M.; VAN GOOL, L. 3DIM'99 Proceedings of the 2nd international conference on 3-D digital imaging and modeling. **Hand-Held acquisition of 3D models with a video camera**, IEEE Computer Society Washington, DC, USA, 1999, p.14-23. 1999.

CHEN, Yang; MEDIONI, Gérard. Robotics and Automation, 1991. Proceedings IEEE International Conference on. **Object Modeling by Registration of Multiple Range Images**, Sacramento CA, p.2724-2729 v.3, abril. 1991.

ZHENG, Jiang Y. Pattern Analysis and Machine Intelligence, IEEE Transactions on, **Acquiring 3-D Models From Sequences of Contours**. IEEE Computer Society, v.16, p. 163-178. Feb. 1994.

LANMAN, Douglas; TAUBIN, Gabriel. SIGGRAPH '09 ACM SIGGRAPH 2009 Courses, **Build Your Own 3D Scanner: 3D Photography for Beginners**. ACM New York, NY, USA, n.8, 2009.

BLAIS, François. Journal of Electronic Imaging, **Review of 20 years of range sensor development**. Institute for Information Technology, Canada, v.13, p.231-240, jan. 2004

ZHENG, Yang Y. Systems, Man, and Cybernetics, IEEE International Conference, **Constructing a 3D Object Model From Multiple Visual Features**. Beijing, v.4, p. 2457-2462, out. 1996.

HARTLEY, Richard I. Computer Vision and Image Understanding, **Triangulation**. Montbonnot, France. V. 68, n.2, p. 146-157, nov. 1997.

SANSONI, Giovanna; TREBESCHI, Marco; DOCCHIO, **State-of-The-Art and Applications of 3D Imaging Sensors in Industry, Cultural Heritage, Medicine, and Criminal Investigation**, MDPI, Italy, 20 jan. 2009. Sensors.

LOBÃO, Fábio S. **Calibração com Mapeamento das Distorções Geométricas Aplicada a um Sistema de Visão Estéreo**. Brasília, 2004. MTARH.DM - 017 A/99.

ALBUQUERQUE, Landecir A. **Alinhamento de Imagens de Profundidade na Reconstrução 3D de Objetos de Forma Livre**. Brasília, 2006. ENDM.DM – 09 A/06.

MILANO, Danilo de; HONORATO, Luciano B. **Visão Computacional**. Limeira, São Paulo, 2008.

EGGERT, David, W.; FITZGIBBON, Andrew W.; FISHER, Robert B. **Simultaneous Registration of Multiple Range Views for Use in Reverse Engineering of CAD Models**. Computer Vision and Image Understanding, Vol 69, pp. 253-272, 1998.

SHREINER, Dave et al. **OpenGL Programming Guide: The official Guide to Learning OpenGL®, Version 4.3**. 8 ed. Pearson Education, 2013

BRADSKI, Gary; KAEHLER, Adrian. **Learning OpenCV**. 1 ed. O'REILLY, 2008

BAGGIO, Daniel L. **Mastering OpenCV with Practical Computer Vision Projects: Step-by-step tutorials to solve common real-world computer vision problems for desktop or mobile, from augmented reality and number plate recognition to face recognition and 3D head tracking**. 1 ed. Packt Publishing, 2012.

HARTLEY, Richard; ZISSERMAN, Andrew. **Multiple View Geometry in computer vision**. 2 ed. Cambridge, The Press Syndicate of The University of Cambridge, 2003.

LANMAN, Douglas; TAUBIAN, Gabriel. **Build Your Own 3D Scanner 3D Photography for Beginners**. Siggraph, 2009

WIKIPÉDIA, **OpenGL**, Wikipédia. < <http://pt.wikipedia.org/wiki/OpenGL>> acesso em: 20/01/2014

RASPBERRY PI, FAQs, <<http://www.raspberrypi.org/faqs>> acesso em: 14/01/2014

SUNDAY, Dan. **Intersections of Lines and Planes**. <http://geomalgorithms.com/a05-_intersect-1.html> acesso em: 15/01/2014.

BROWN, Duane C. **Decentering Distortion of Lenses: The prisms effect encountered in metric cameras can be overcome through analytic calibration.** pg 444 – 462. Photogrammetric Engineering.

MORVAN, Yannick. **Acquisition, Compression and Rendering of Depth and Texture for Multi-View Video.** University of Eindhoven, 2009

8. ANEXO I

