

UNIVERSIDADE ESTADUAL DE MARINGÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FRANCISCO ALEXANDRE RIBEIRO DE ALENCAR

Concepção e implementação de um *scanner* 3D para ensino e aprendizagem

Maringá
2010

FRANCISCO ALEXANDRE RIBEIRO DE ALENCAR

Concepção e implementação de um *scanner* 3D para ensino e aprendizagem

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Departamento de Informática, Centro de Tecnologia da Universidade Estadual de Maringá, como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Dante Alves Medeiros Filho

Maringá
2010

"Dados Internacionais de Catalogação-na-Publicação (CIP)"
(Biblioteca Setorial - UEM. Nupélia, Maringá, PR, Brasil)

A368c

Alencar, Francisco Alexandre Ribeiro de, 1978-
Concepção e implementação de um *scanner* 3D para ensino e aprendizagem /
Francisco Alexandre Ribeiro de Alencar. -- Maringá, 2010.
113 f. : il. color.

Dissertação (mestrado em Ciência da Computação)--Universidade Estadual de
Maringá, Dep. de Informática, 2010.

Orientador: Prof. Dr. Dante Alves Medeiros Filho.

1. Digitalização tridimensional - *Scanner* 3D - Ensino e aprendizagem. 2.
Sistemas embarcados - Controle digital de motores. 3. Computação gráfica. I.
Universidade Estadual de Maringá. Departamento de Informática. Programa de Pós-
Graduação em "Ciência da Computação".

CDD 22. ed. -006.693
NBR/CIP - 12899 AACR/2

FOLHA DE APROVAÇÃO

FRANCISCO ALEXANDRE RIBEIRO DE ALENCAR

Concepção e implementação de um *scanner* 3D para ensino e aprendizagem

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Departamento de Informática, Centro de Tecnologia da Universidade Estadual de Maringá, como requisito parcial para obtenção do título de Mestre em Ciência da Computação pela Comissão Julgadora composta pelos membros:

COMISSÃO JULGADORA

Prof. Dr. Dante Alves Medeiros Filho
Universidade Estadual de Maringá – DIN/UEM

Profa. Dr. Airton Marco Polidório
Universidade Estadual de Maringá – DIN/UEM

Prof. Dr. Pedro Luís Faggion
Universidade Federal do Paraná – DGEOM/UFPR

Aprovada em: 26 de fevereiro de 2010.

Local da defesa: Sala 110, Bloco C-56, *campus* da Universidade Estadual de Maringá

DEDICATÓRIA

*Dedico esse trabalho a memória
de Percival Otoni de Alencar,
meu Pai, meu Ídolo e meu Herói.*

AGRADECIMENTOS

Agradeço primeiramente a Deus que me guiou e deu forças durante essa dura jornada.

Á minha Esposa, pelo amor incondicional e por sempre ter acreditado no meu trabalho, me dando apoio e carinho, além de ter ficado ao meu lado nos momentos mais difíceis da minha vida.

A meu Pai (*in memorian*), pelo exemplo de honestidade e caráter que eu tive durante toda a vida.

Á minha Mãe, que sempre me incentivou e me ajudou cuidando e educando meu filho enquanto eu não podia.

Á minha irmã Gizeli, pelo carinho, apoio e pelos dias em que me ajudou entender um pouco do que significam as palavras “Ensino” e “Aprendizagem”.

Ao meu grande amigo e sócio André Luiz Scarate, pelas pizzas, pelos churras, pelo apoio e pela amizade.

Aos meus amigos do curso Everton, Maurílio, Gécen, Renata e Nelson, pelas conversas, piadas, paçocas e horas de estudo que passamos juntos.

Ao professor Airton pelo apoio e ajuda nos algoritmos de processamento de imagem.

A Maria Inês Davanço Laccort, a secretária do programa, que sempre esteve presente auxiliando a todos os estudantes, sempre com muita paciência, atenção e carinho.

Aos técnicos da PPG, Michelle e Antonio, que sempre foram muito prestativos e atenciosos.

Agradeço também ao professor e amigo Paulo César Gonçalves, pelo suporte prestado para realizar a gravação do microcontrolador.

Finalmente agradeço a meu orientador, Prof. Dante que acreditou na minha capacidade e foi muito paciente nos momentos mais difíceis, e sem ele esse trabalho não seria possível.

Concepção e implementação de um *scanner* 3D para ensino e aprendizagem

RESUMO

A digitalização tridimensional é um dos grandes desafios em Computação Gráfica (CG). Essa temática abrange conhecimentos das áreas de visão computacional, processamento de imagens, modelagem geométrica, visualização e desenvolvimento de sensores. A reconstrução tridimensional de objetos tem inúmeras aplicações em diversas áreas como Engenharias; Arqueologia; Patrimônio Histórico; Arquitetura; Educação; Arte; Medicina; Desenho Industrial entre outros. Para viabilizar a adequada digitalização de uma superfície, possibilitando sua reconstrução tridimensional, são necessários basicamente três passos: aquisição de dados de elevação dessa superfície, processamento desses dados e a reconstrução da superfície com a precisão especificada. Esta técnica, embora interessante, tem se mostrado de difícil apropriação no processo de ensino e aprendizagem de computação gráfica. Tendo em vista que o contato direto com representações concretas, segundo a literatura, facilita a aprendizagem, o desenvolvimento deste trabalho, foi o ponto de partida para mostrar um método simplificado para a produção de um digitalizador tridimensional, baseado em software livre com o objetivo de motivar e facilitar a aprendizagem de conceitos na área de computação gráfica.

Palavras-chave: *Scanner* 3D. Digitalização tridimensional. Ensino e Aprendizagem. Computação Gráfica.

Design and implementation of a 3d scanner for teaching and learning

ABSTRACT

The three-dimensional scanning is one of the major challenges in computer graphics (CG). This subject covers knowledge of the areas of computer vision, image processing, geometric modeling, visualization and sensor development. Three-dimensional reconstruction has many applications in different areas like Engineering, Archeology, Heritage, Architecture, Education, Arts, Medicine and Industrial Design among others. To be able to perform properly scanning a surface allowing the three-dimensional reconstruction is needed basically three steps: acquisition of elevation data of surface processing such data and surface reconstruction on the accuracy specified. This process, although interesting, has proved difficult to ownership in the process of teaching and learning of computer graphics. Considering that direct contact with concrete representations, according to the literature, facilitates learning the development of this work was the starting point to show a simplified technique for producing a three-dimensional scanner based on free software in order to motivate and facilitate the learning of concepts in the field of computer graphics.

Keywords: 3D Scanner. Three-dimensional scanning. Teaching and Learning. Computer Graphics.

LISTA DE FIGURAS

Figura 1: Cone de experiências.....	28
Figura 2: Imagem gerada a partir de primitivos sólidos (PERTENCE, SANTOS E JARDIM, 2001, pág 224).....	33
Figura 3: Objeto num plano intermediário.....	34
Figura 4: Exemplo de oclusão.....	34
Figura 5: Gazebo.....	35
Figura 6: exemplo de sombra.....	36
Figura 7: reflexão da luz.....	36
Figura 8: Visões da modelagem de uma nave espacial.....	37
Figura 9: modelo de camera pinhole.....	38
Figura 10: Pato em diferentes níveis de discretização.....	40
Figura 11 – Digitalização por silhueta.....	41
Figura 12 – Taxonomia de tecnologias de digitalização 3D.....	42
Figura 13 – Digitalizador 3D por contato.....	42
Figura 14: Padrões luminosos utilizados em digitalizadores 3D.....	43
Figura 15 - Dados do Laser Range Scanner Riegl LMS Z-360I com informação de textura associada.....	43
Figura 16: Maquete eletrônica do scanner 3D.....	46
Figura 17: Primeiro protótipo do scanner 3D.....	46
Figura 18: Diagrama de blocos do software principal.....	47
Figura 19: Tela de opções do ambiente de desenvolvimento Delphi.....	50
Figura 20: Tela de adição de pastas no caminho de busca das bibliotecas.....	50
Figura 21: Árvore de pastas da tela de adição de pastas ao caminho de busca de bibliotecas.....	51
Figura 22: Compilação e instalação do GLScene.....	51
Figura 23: Mensagem de sucesso da instalação do GLScene.....	52
Figura 24: Barra de ferramentas do GLScene adicionada ao Delphi.....	52
Figura 25: GLScene Editor – editor específico de um objeto TGLScene.....	53
Figura 26: Ordem de plotagem dos vértices em relação ao vetor normal.....	54
Figura 27: Pacote DirectX9 a ser apenas compilado.....	55
Figura 28: Mensagem de sucesso da instalação do DSPack.....	56
Figura 29: Barra de ferramentas do DSPack adicionada ao Delphi.....	56
Figura 30: Diagrama de classes do DSPack.....	57
Figura 31: Esquema de funcionamento de um motor.....	58
Figura 32: Motor de passo unipolar de 5 fios.....	59
Figura 33: esquema de seqüência de 4 passos de um motor de passo.....	60
Figura 34: Microcontrolador genérico.....	61
Figura 35: Diagrama de blocos do PIC 16F628A.....	62
Figura 36: Exemplo de palavras de 14 bits do PIC16F628A.....	63
Figura 37: Ciclo de instruções e pipeline do PIC 16F628A.....	64
Figura 38: Diagrama de pinagem do PIC16F628A.....	65
Figura 39: Diagrama de blocos do programa gravado no microcontrolador.....	66
Figura 40: Diagrama de funcionamento das portas do CI ULN2003.....	68
Figura 41: Corrente dos níveis lógicos do PIC 16F628A e da comunicação serial pela norma EIA-232.....	69
Figura 42: Diagrama de pinagem do CI MAX232.....	69
Figura 43: Representação de um pacote de dados transmitido em uma comunicação serial pela norma EIA-232.....	71
Figura 44: Esquema do circuito da placa controladora de motor de passos.....	73
Figura 45: Diagrama do circuito da placa controladora de motor de passos.....	73
Figura 46: Posicionamento inicial do objeto na plataforma.....	74

Figura 47: Primeira seqüência de pontos.....	75
Figura 48: Processo de captura de imagem por um sensor digital	76
Figura 49: Amostragem de um sinal analógico	77
Figura 50: Aliasing.....	78
Figura 51: Anti-aliasing.....	79
Figura 52: Imagem original capturada	80
Figura 53: Imagem após o realce da silhueta	80
Figura 54: Silhueta destaca após o segundo processamento	81
Figura 55: Silhueta fina destacada e resultado final do processamento apresentando apenas a silhueta fina	81
Figura 56: Impressão do circuito em uma folha de revista.....	85
Figura 57: Processo de transferência térmica	86
Figura 58: Processo de esfoliação da placa de circuito	86
Figura 59: Placa de circuito com correções nos desenhos das trilhas	87
Figura 60: Placa mergulhada em percloreto de ferro	87
Figura 61: Placa de circuito com alguns componentes soldados.....	88
Figura 62: Diagrama de montagem da plataforma giratória	88
Figura 63: Furação da caixa de MDF e encaixe do motor de passo	89
Figura 64: Fixação da plataforma giratória e finalização da plataforma.....	89
Figura 65: Tela inicial – seleção da porta serial onde o scanner está conectado.....	90
Figura 66: Menu de seleção de dispositivo de captura de imagens.....	90
Figura 67: Menu de configuração do dispositivo de captura de imagens	91
Figura 68: Centralização do objeto em relação a linha guia do sistema.....	91
Figura 69: resultado da captura do objeto	92
Figura 70: resultado da captura do objeto em outro ângulo	92
Figura 71: resultado da captura do objeto: nuvem de pontos adquirida pelo sistema	92

LISTA DE TABELAS

Tabela 1: Pinagem do PIC16F628A.....	65
Tabela 2: seqüência de energização dos fios do motor de passo.....	68
Tabela 3: Tabela de caracteres ASCII.....	72
Tabela 4: Tabela de caracteres ASCII completa	108

LISTA DE ABREVIATURAS E SIGLAS

AC	<i>Corrente Alternada</i>
AD	<i>Analógico para Digital</i>
ANSI	<i>American National Standards Institute</i>
ASCII	<i>American Standard Code for Information Interchange</i>
bps	<i>bits por segundo</i>
B-REP	<i>Boundary Representation</i>
CAD	<i>Computer Aided Design</i>
CG	<i>Computação Gráfica</i>
CI	<i>Circuito Integrado</i>
DA	<i>Digital para Analógico</i>
DC	<i>Corrente Direta</i>
E/S	<i>Entrada e Saída</i>
EEPROM	<i>Electrically-Erasable Programmable Read-Only Memory</i>
EIA	<i>Electronic Industries Association</i>
IDE	<i>Integrated Development Environment</i>
MDF	<i>Médium Density Fiberboard</i>
MIT	<i>Massachusetts Institute of Technology</i>
PIXEL	<i>Picture Elements</i>
PWM	<i>Pulse-Width Modulation</i>
SAGE	<i>Semi-Automatic Ground Environment</i>

SUMÁRIO

Introdução	19
O processo de ensino e aprendizagem	22
2.1 Considerações Iniciais	22
2.1.1 As contribuições de Skinner.....	24
2.1.2 As contribuições de Piaget	25
2.1.3 As contribuições de Vygotsky	26
2.2 Recursos de ensino	27
Digitalização 3D.....	31
3.1 Considerações Iniciais	31
3.2 Scanner 3D: Conceitos básicos	40
Da teoria à prática.....	45
4.1 Considerações Iniciais	45
4.2 Materiais e métodos	48
4.2.1 GLScene.....	48
4.2.2 DSPack.....	54
4.3 A aquisição da imagem.....	57
4.3.1 Motor de passo	58
4.3.2 Circuito	60
4.3.3 Utilização da plataforma	74
4.3.4 Captura da Imagem.....	75
4.4 O processamento da imagem	79
4.5 O processamento do eixo central da silhueta	81
4.7 A modelagem	82
Desenvolvimento e resultados.....	84
5.1 Placa de circuito impresso.....	84
5.2 Plataforma giratória	88
5.3 Uso do software gerenciador do sistema	90
Conclusão	94
6.1 Limitações da pesquisa.....	94
6.2 Dificuldades encontradas	95
6.3 Trabalhos futuros.....	95
6.4 Considerações finais.....	95
Referências.....	98

Apêndice A: Código fonte do programa embarcado	101
Anexo A: Linha temporal da área de computação gráfica.....	105
Anexo B: Linha temporal da área de computação gráfica.....	108

Introdução

Atualmente a sociedade vive uma época em que se desenvolve uma profusão de aparatos tecnológicos, entre os quais se inclui a digitalização tridimensional (3D) de objetos reais que constitui uma área de pesquisa promissora da Computação Gráfica (CG), pois unifica técnicas de visão computacional, processamento digital de imagens, modelagem geométrica, visualização e desenvolvimento de sensores habilitados para esse fim. Existem aplicações em campos como Arqueologia, Arquitetura, Educação, Artes, Medicina, Desenho Industrial e Engenharia, possibilitando a recriação digital de itens a serem estudados.

A digitalização 3D consiste basicamente em adquirir dados sobre a posição espacial de diversos pontos dispostos em uma determinada superfície tridimensional de forma organizada para possibilitar a ação de um processamento que seja capaz de realizar a reconstrução digital tridimensional dessa superfície e por fim gerar um modelo digital correspondente. Esses modelos permitem, por exemplo, realizar estudos não destrutivos e investigativos acerca de objetos reais correspondentes.

A aquisição dos dados de posição espacial para pontos em uma superfície é a primeira etapa do processo de digitalização. Essa aquisição pode ser realizada a partir de sensores ópticos, sensores de contato e sensores ultra sônicos. Após isso, é necessário processar esses dados adquiridos e escolher um modelo apropriado para a reconstruir e representar o objeto.

Para se fazer estudos adequados a respeito de técnicas de digitalização tridimensional é necessário especificar e calibrar os sensores utilizados; definir os tipos e a natureza das superfícies que podem ser reconstruídas com o uso desses sensores; definir as relações geométricas entre sensor(es) e superfície; definir a natureza do ambiente de aquisição dos dados (condições controladas ou não); definir a taxa de amostragem para aquisição dos dados e; definir os pré-processamentos necessários para tratar esses dados.

Contudo os pesquisadores dessa área tendem a descrever a eficácia e os modelos matemáticos utilizados no desenvolvimento desses projetos, por meio de uma linguagem que se caracteriza por sua estrutura técnica e complexa, os quais dificultam o entendimento e o aprendizado desses assuntos por parte dos alunos.

Pesquisas têm mostrado que o processo de ensino e aprendizagem na área tecnológica é carente de aplicação de metodologias e técnicas facilitadoras. Mas, de outra sorte, é lugar comum afirmar que atividades que trabalham com objetos concretos motivam e facilitam a aprendizagem (BAZZO, 1998).

Diante desta situação é que surgiu a motivação para o desenvolver o presente trabalho para mostrar a produção de um digitalizador tridimensional simplificado, o qual pode ser utilizado como recurso mediador em sala de aula, capaz de envolver na prática os alunos em todas as etapas do processo de reconstrução 3D de uma superfície.

Neste contexto o desafio do trabalho é propor uma forma contemporânea e contextualizada de mediação de um conteúdo específico da disciplina de computação gráfica que possa ser utilizada como recurso prático no ensino e aprendizagem de técnicas relativas a Computação Gráfica. Para tanto construímos um scanner 3D baseado em análise de silhuetas de projeção de feixe de luz.

Os objetivos específicos deste trabalho consistiram em projetar e construir uma plataforma giratória, capaz de girar um objeto durante o processo de digitalização; projetar e manufaturar uma placa de circuito impresso para instalação dos componentes necessários ao controle digital do motor de passo; implementar um sistema computacional para controlar a plataforma giratória, receber e processar os dados oriundos da webcam; implementar um sistema que ilustre na prática as fases do processo de reconstrução 3D.

O texto que compõe esse trabalho está organizado da seguinte forma: introdução; o

segundo capítulo versa sobre o processo de ensino e aprendizagem abordando os principais autores da área e tematiza sobre os recursos de ensino; o terceiro capítulo abrange os conceitos básico sobre digitalização 3D; no quarto capítulo encontra-se o processo de produção e implementação do hardware e do software que compõem o scanner 3D; no quinto capítulo são discutidos os resultados do trabalho; por fim, no sexto capítulo, apresenta-se as considerações finais.

O processo de ensino e aprendizagem

2.1. Considerações Iniciais

O processo de ensino e aprendizagem possui uma característica que deve ser destacada, a “intencionalidade”. Para executar essa característica é necessário que haja planejamento, organização e fundamentação. A mediação escolar é algo que não pode ocorrer ao acaso, é preciso fundamentação para a sua execução. Nesse sentido é impreterível apontar algumas características do processo de ensino e aprendizagem que devem ser conhecidas, para corroborar com essa fundamentação.

A natureza do processo de ensino e aprendizagem é complexa por envolver aspectos intrínsecos a cada ser humano, o que permite derivar várias interpretações e não pode ser considerada como uma realidade acabada. A este respeito tem-se o seguinte comentário:

“Há muitas formas de se conceber o fenômeno educativo. Por sua própria natureza, não é uma realidade acabada que se dá a conhecer de forma única e precisa em seus múltiplos aspectos. É um fenômeno humano, histórico e multidimensional. Nele estão presentes, tanto a dimensão humana quanto a técnica, a cognitiva, a emocional, a sócio-política e cultural. Não se trata de mera justaposição das referidas dimensões, mas sim, da aceitação de suas múltiplas implicações e relações” (MIZUKAMI, 1986, p.1).

Embora o processo educativo não seja uma realidade acabada que se dá a conhecer de forma única e precisa, muitas interpretações deste fenômeno foram realizadas, sendo atualmente

consideradas aproximações históricas que permitem explicá-lo, se não em sua totalidade, pelo menos em alguns de seus aspectos.

Essas aproximações são interpretações do processo educativo que foram disseminadas em forma de propostas, abordagens ou teorias, com inegáveis contribuições para o processo de ensino-aprendizagem, sendo ainda analisadas, contextualizadas e discutidas criticamente.

Cada proposta, abordagem ou teoria privilegia um ou outro aspecto do fenômeno educacional, o que as levam a vários tipos de reducionismo, mas também são explicativas de determinados aspectos do processo ensino-aprendizagem, não podendo ser desconsideradas.

Neste sentido sabe-se que são tão diferentes os significados de certos conceitos quanto são os estudiosos que tentam defini-los. Assim, tem ocorrido, muitas vezes, com os conceitos sobre “ensino” e “aprendizagem”. Ora se enaltece a particularidade de uma, ora a generalidade da outra sem que haja clareza e precisão em suas abordagens com propósitos científicos. É de se esperar que as definições empregadas sejam orientadoras das práticas educativas não só dos objetivos traçados como das ações para sua realização.

Dentro dos limites do presente trabalho não é possível abordar todos os aspectos referentes a questões concernentes a ensino e aprendizagem. Contudo, alguns apontamentos se fazem necessários haja vista o objetivo do estudo em questão ser a construção de um instrumento que auxilie no processo de mediação escolar.

Para melhor sistematizar as correntes teóricas sobre o processo de ensino e aprendizagem, a partir da qual tenta-se melhor explicitar e viabilizar o processo de apropriação de conhecimento, em relação ao objeto de pesquisa, é conveniente definir o que é ensinar e o que é aprender.

De acordo com Bordenave e Pereira (1982), o “ensinar” recai sobre três fatores: o aluno, o assunto e o professor. De acordo com esses autores a aprendizagem por parte do aluno está relacionada ao desejo de aprender e ao conhecimento que ele já possui sobre o conteúdo de referencia. Para que essa aprendizagem venha a se efetivar, afirmam, faz-se necessário estruturar o conteúdo, ou seja, seus componentes e relações bem como a ordem em que serão apresentados.

O professor, por sua vez, deve propor situações estimuladoras, ter uma boa comunicação verbal para passar as instruções (conteúdo) e informações ao aluno. Para esses autores “o processo

de ensino é um processo pragmático, isto é, um mecanismo pelo qual se pretende alcançar certos objetivos e para isso se mobilizam meios, organizando-se em uma estratégia seqüencial e combinatória” (BORDENAVE e PEREIRA, 1982). Portanto, consistem em planejamento, orientação e controle da aprendizagem do aluno.

Em contrapartida ao que é ensinar temos o que é aprender. Segundo Bordenave e Pereira (1982) a aprendizagem está presente em vários processos:

- Na necessidade de se resolver um problema, portanto, quem aprende tem um objetivo a alcançar;
- A preparação para superar as possíveis barreiras que impedem o alcance do objetivo, tais como estudo, leitura, consultas, perguntas, dentre outras;
- A constatação do sucesso ou fracasso das tentativas;
- Os diversos objetos de aprendizagem, ou seja, os diferentes processos mentais utilizados;
- As experiências prévias;
- As mudanças cognitivas e emotivas;
- Aquisição de novos conhecimentos;
- Processo qualitativo que proporciona transformação estrutural da inteligência, dentre outros.

Embora exista uma variedade de teorias que buscam definir o processo de ensino e aprendizagem, pode-se distinguir pelo menos três abordagens que tem maior disseminação no campo educacional, a saber: os postulados de Skinner, Piaget e Vygostsky.

Esta pesquisa não tem por objetivo aprofundar-se nas referidas teorias, haja vista a complexidade que envolve tal tarefa, mas sim abordar apenas os conceitos fundamentais para fundamentar este trabalho.

2.1.1. As contribuições de Skinner

A teoria de estímulo e resposta (E – R) desenvolvida por Skinner influenciou diretamente os procedimentos e materiais usados em sala de aula na década de 60 e 70 do século passado. Essa abordagem é denominada de teoria conexionista, pois partem do pressuposto de conexão entre estímulo e resposta (MOREIRA, 1999).

Esse enfoque, segundo Moreira (1999), não se preocupa com o processo intermediário ao

estímulo (E) e resposta (R), que ocorrem durante o processo de ensino e aprendizagem. Dentro dessa proposta, as pessoas envolvidas no processo tendem a se comportar de forma satisfatória com intuito de obter recompensas e evitar punições. Para atingir respostas satisfatórias, essa teoria propõe um procedimento definido como condicionamento. Contudo, não podemos confundir condicionamento com aprendizagem,

O termo aprendizagem é, razoavelmente, definido como ‘uma mudança no comportamento produzida pela experiência’. [...]. O significado de condicionamento é muito mais específico e exato do que o de aprendizagem: é o aumento na frequência de uma resposta que foi, recentemente, associada com um reforçador positivo sob condições explícitas. (MOREIRA, 1999, p.52)

O reforçador é subentendido como recompensa. O ensino, dentro dessa perspectiva, se dá apenas quando o professor coloca sob controle de certas contingências – reforços – o que precisa ser ensinado. Dessa forma, a principal função do professor é programar as contingências.

2.1.2. As contribuições de Piaget

A teoria de Piaget, de acordo com Moreira (1999), configura-se como teoria do desenvolvimento mental.

Piaget disserta sobre o processo de assimilação, acomodação e equilíbrio, para explicar o que ele denomina de “aumento do conhecimento”, ou seja, crescimento cognitivo. Nesse processo, ao interagir com o objeto o sujeito constrói esquemas de assimilação para abordar a realidade. Quando o organismo, a mente, assimila determinado conteúdo ocorre a assimilação. Quando essa assimilação não ocorre o organismo (mente) desiste ou se modifica. Essa modificação denomina-se na teoria Piagetiana de acomodação.

Por meio da acomodação ocorrem novas construções de esquemas de assimilação. Em suma, pode-se dizer que o desenvolvimento mental é explicado tomando como referência os esquemas de assimilação que a pessoa utiliza.

Sob essa perspectiva, só há aprendizagem quando há acomodação, pois a mente enquanto estrutura cognitiva funciona em equilíbrio e aumenta o grau de organização e adaptação ao meio. Quando o equilíbrio não se efetiva ela se reestrutura, ou seja, se acomoda novamente para então construir novos esquemas de assimilação para assim atingir um novo equilíbrio, denominado na teoria piagetiana de equilíbrio majorante responsável pela aprendizagem também denominado aumento de conhecimento. Nas sistematizações teóricas de Piaget, conhecer significa organizar, estruturar e explicar o real a partir das experiências vividas. Conhecer é modificar, transformar o

objeto; é compreender o mecanismo de sua transformação e, conseqüentemente, o caminho pelo qual o objeto é construído. O conhecimento é sempre produto da ação do sujeito sobre o objeto. Nesse sentido, a operação é a essência do conhecimento: a ação interiorizada modifica o objeto do conhecimento, impondo-lhe uma ordenação no espaço e no tempo. (PALANGANA, 1998, p. 72).

De acordo com essa concepção para que o sujeito extraia as características intrínsecas e assimile algo é de vital importância agir sobre os objetos para assim combinar e efetuar deduções das características abstraídas por meio de experiência física e lógico-matemática.

Na experiência física o sujeito extrai de uma classe de objetos as características e desenvolve a experiência lógico-matemática, além de agir sobre o objeto, descobre propriedades que “estão, pelo contrário, abstratas das ações mesmas do sujeito, de tal forma que, num certo nível de abstração, a experiência sobre os objetos se torna inútil e a coordenação das ações basta para engendrar uma manipulação operatória simplesmente simbólica e procedendo assim de maneira puramente dedutiva.” (PIAGET, 1974, p. 37 *apud* PALANGANA, 1998, p. 73). Assim sendo, quanto mais à teoria da aprendizagem se distancia das necessidades do sujeito, mais terá que apelar para os fatores motivacionais (externos) para explicar o processo de aprendizagem.

Mediante o exposto, o ensino envolve três aspectos em termos de assimilação:

1. os esquemas de assimilação do aluno,
2. aquilo que se quer ensinar,
3. e o professor.

Piaget afirma que o ensino “deve ser acompanhado de ações e demonstrações e, sempre que possível, deve dar aos alunos a oportunidade de agir (trabalho prático).” (MOREIRA, 1999, p. 104).

2.1.3. As contribuições de Vygotsky

Os pressupostos Vygotskyanos, por sua vez, têm enfoque nas interações sócio-culturais. Para este teórico o homem é um ser biológico e histórico cultural. Assim sendo, o desenvolvimento da consciência (processo psicológico superior, do sujeito e da humanidade) são resultantes de um processo sócio-histórico. O desenvolvimento, de acordo com sua abordagem, ocorre de fora para dentro, ou seja, do intersubjetivo para o intrasubjetivo.

Vygotsky postula quatro entradas para o desenvolvimento: a filogênese, a ontogênese, a sociogênese e a microgênese.

Em síntese: o autor postula que por meio da filogênese (história da espécie animal), define-se os limites e possibilidades de desenvolvimento psicológico e, por conseguinte, do cérebro humano. A ontogênese, por sua vez, que estuda o desenvolvimento de um indivíduo, de uma determinada espécie, está ligada à filogênese.

A sociogênese estuda a história da cultura na qual o sujeito está inserido, bem como as formas de funcionamento cultural que constituem o desenvolvimento humano. Por fim a microgênese, que discute a percepção das singularidades de cada sujeito cognoscente, ou seja, que cada fenômeno psicológico tem sua própria história, o que permite romper com o primado determinista presente nas três entradas acima.

Mediante o exposto, Vygotsky afirma que toda atividade humana tem como característica a mediação pautada por instrumentos técnicos e sistemas de signos, os quais são construídos historicamente. A linguagem ocupa lugar de destaque enquanto signo, pois carrega em si os conceitos generalizados e elaborados pela cultura humana. Portanto, na mediação simbólica, a relação do homem com o mundo é mediada por instrumentos (natureza concreta) e por signos (natureza simbólica).

A interação do indivíduo com o objeto, de acordo com o postulado, aparece como uma atividade mais ampla, dotada de especificidade histórico-cultural, portanto “os experimentos deveriam servir, sobretudo, para iluminar processos. Para isso, a metodologia experimental deveria oferecer o máximo de oportunidades para que o sujeito se engajasse nas mais diversas atividades que pudessem ser observadas, ao invés de rigidamente controladas” (COLE e SCRIBNER, 1988, p. 13 apud MOREIRA, 1999, p.117). Esse pressuposto enfatiza o processo e não o produto.

2.2. Recursos de ensino

Os recursos de ensino são utilizados com intuito de transcender as instruções que geralmente são viabilizadas em sala de aula no quadro negro ou por meios de textos digitados, “são componentes do ambiente de aprendizagem que dão origem à estimulação para o aluno” (GAGNÉ, 1971, p.247). Esses recursos podem ser classificados de várias formas: recursos visuais (projeções,

cartazes e gravuras), auditivos (rádio e gravações) e audiovisuais (cinema e televisão); recursos humanos (professor, alunos, pessoal escolar e comunidade) e materiais (do ambiente: natural e escolar; da comunidade: bibliotecas, indústrias, lojas, etc.) ou ainda como cone de experiências (parte do imediatamente vivenciado e chega ao simbólico abstrato) (PILETTI, 1997), dentre outros. Para tanto, deve-se levar em consideração o grau de abstração dos educandos.

Dentre essas classificações nos ateremos ao cone de experiências (figura 1), desenvolvido por Edgar Dale. (*apud* PILETTI, 1997, p. 153), onde é possível verificar que quanto mais teórico o conteúdo for, mais conceitos prévios são necessários para o seu entendimento. Já, em relação ao conteúdo prático, percebe-se que há uma menor quantidade de conceitos prévios necessários à sua compreensão.

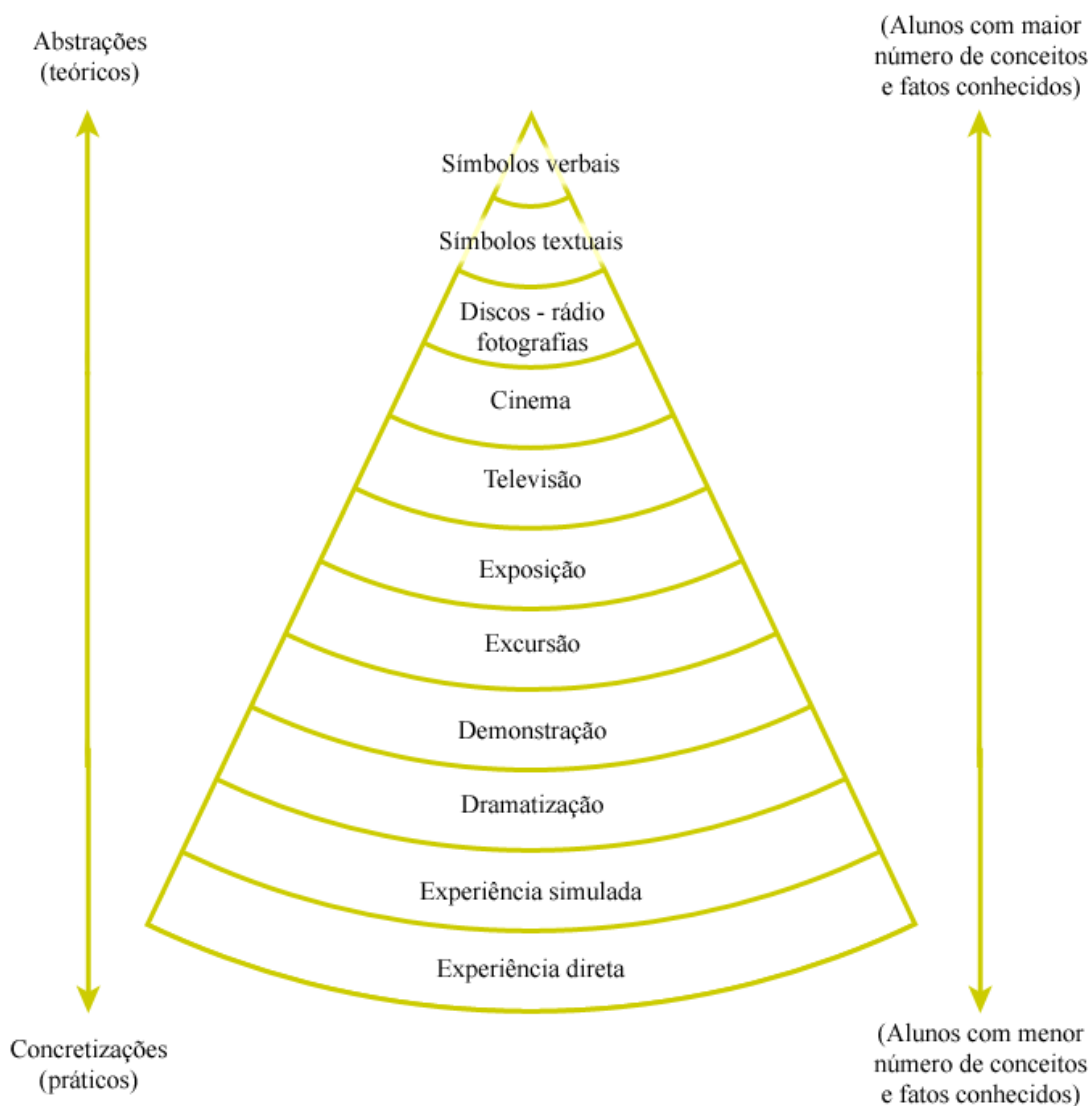


Figura 1: Cone de experiências.

Fonte: Piletti(1997)

De acordo com o autor esse tipo de classificação permite ao professor utilizar o recurso selecionado de forma a proporcionar gradativamente aos alunos experiências diretas evitando assim recorrer apenas ao ensino verbalístico.

Vários fatores motivam a utilização de recursos de ensino, como: aumentar a motivação e despertar o interesse; beneficiar o desenvolvimento e a capacidade de observação; desenvolver a experimentação concreta; ilustrar noções mais abstratas; aproximar o aluno da realidade, permitir acesso a informações e dados reais e auxiliar uma melhor fixação da aprendizagem. (PILETTI, 1997).

Contudo, a utilização de um recurso de ensino deve ser baseada em critérios e princípios. Faz-se necessário ter claramente definidos os objetivos a serem alcançados, o domínio do recurso selecionado, conhecer as possibilidades e limitações de suas características, para assim avaliar a eficácia do mesmo; analisar as condições ambientais onde o recurso de ensino será aplicado, bem como o tempo disponível.

Vale frisar que a eficiência do processo didático está associada, dentre outros fatores, à seleção dos recursos utilizados, devendo estes, estarem voltados às necessidades do aprendiz.

Digitalização 3D

3.1. Considerações Iniciais

A digitalização 3D consiste basicamente em adquirir dados sobre a posição espacial de diversos pontos dispostos em uma determinada superfície tridimensional que possibilitam um processamento que permita reconstruí-la para gerar um modelo digital correspondente. Esses modelos permitem, por exemplo, realizar estudos não destrutivos e investigativos acerca de seus objetos reais. Em um ambiente de digitalização 3D é essencial dispor de uma plataforma genérica e flexível para se obter bons resultados. Essa plataforma deve facilitar a investigação de novos problemas e, ao mesmo tempo, permitir validar as soluções de maneira consistente com as aplicações reais.

Assim, para iniciar os estudos sobre o scanner 3D se faz necessário, num primeiro momento, uma breve descrição sobre as origens da computação gráfica, para em seguida adentrarmos nos conceitos utilizadas para o desenvolvimento do mesmo.

Há um consenso entre os pesquisadores que o primeiro computador com recursos gráficos foi o *Whirlwind I*, desenvolvido pelo MIT (*Massachusetts Institute of Technology, USA*) em 1950. O *Whirlwind I* foi utilizado, em 1955, como plataforma para o desenvolvimento de um sistema de monitoramento de controle de vôos denominado SAGE (*Semi-Automatic Ground Environment*). No final da década de 50, do século passado, segundo Azevedo e Conci(2003), Verner Hudson cria o termo *Computer Graphics*, durante a coordenação de um projeto

desenvolvido para a Boeing de simulação de fatores humanos em aviões.

A década de 60 foi palco de importantes publicações, desenvolvimento e disseminação da computação gráfica, mais especificamente ao relacionar-se à produção do primeiro software CAD (*Computer Aided Design*), pela *General Motors*, utilizado posteriormente nas indústrias automobilísticas e aeroespaciais americanas.

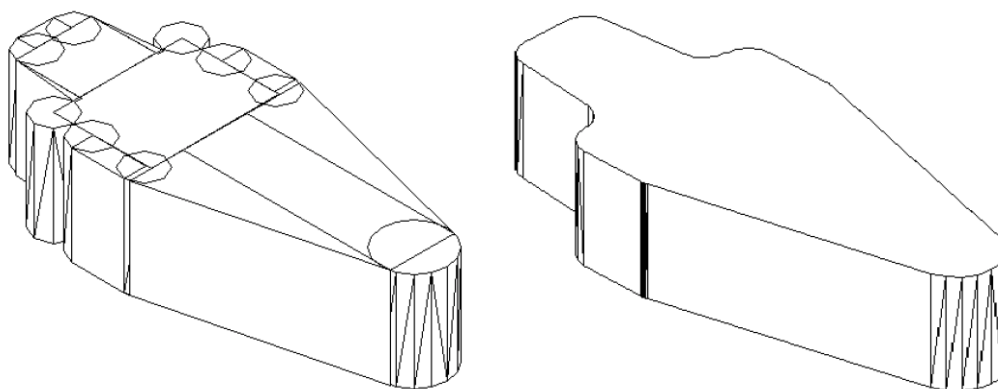
O algoritmo *z-buffer*, utilizado até os dias atuais, foi concebido na década de 1970, concomitante a outras tecnologias tais como: circuitos integrados e o primeiro computador com interface visual. Cumpre frisar que neste período a computação gráfica foi reconhecida como área específica da ciência da computação. Uma década depois fotos de satélites e de explorações interplanetárias receberam tratamento utilizando técnicas da computação gráfica para o processamento das imagens. Na década de 1990, atinge maior maturidade e ganha destaque nas produções cinematográficas de longa metragem, que pode ser constatada nas produções *O Exterminador de Futuro 2* e *Jurrassic Park*, onde o fotorrealismo se tornou algo impressionante para a época. Em 1992 surge a linguagem de programação *OpenGL*. Nessa década também foi lançado *Cassiopeia*, produção 100% brasileira e o primeiro filme realizado inteiramente utilizando computação gráfica. Há uma controvérsia nesse fato, pois a *Disney* afirma que o primeiro filme inteiramente feito por computador foi *Toy Story*, mas seus personagens foram feitos utilizando moldes de argila, e só depois foram digitalizados, ao contrário de *Cassiopeia* que teve todos os personagens concebidos diretamente no computador. No final da década de 90 a *NVidia* lançava suas primeiras placas gráficas. Nesse período vale frisar outro importante marco na história do cinema e campeão do Oscar de efeitos especiais: o filme *Matrix*.

A partir do ano 2000 evidenciou-se um grande avanço no campo das produções gráficas e grandes sucessos no cinema devido a tecnologia, tais como: *Shrek*, *Final Fantasy* e *Matrix Reloaded*¹.

A computação gráfica é complexa e para entendê-la e correlacioná-la com o objeto de estudo (scanner 3D) se faz necessário conhecimento prévio dos conceitos de percepção tridimensional; perspectiva; conhecimento prévio do objeto; oclusão; densidade de texturas; sombra; reflexão da luz; modelagem tridimensional; geometria projetiva; câmeras e modelos de câmeras; calibração; reconstrução no espaço da imagem, reconstrução no espaço da cena e geração de malhas.

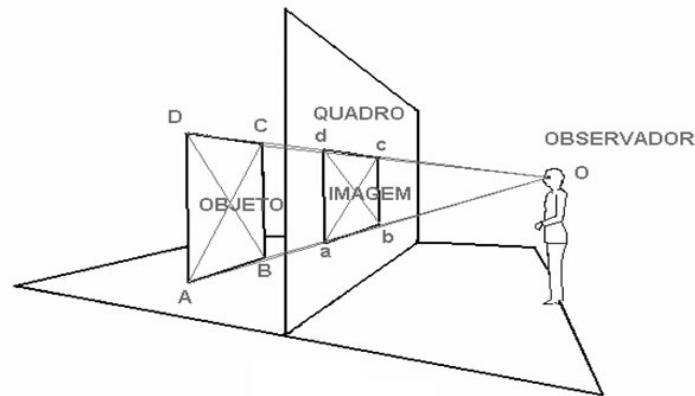
¹ Em anexo segue a linha temporal dos avanços na área de computação gráfica de acordo com Azevedo e Conci(2003).

- **Percepção tridimensional:** A percepção tridimensional se desenvolve à medida que um indivíduo vivência o espaço através principalmente da visão e do tato, captam estímulos, como brilho, sombra, cor, frio ou quente, tipo de contato etc. Esses estímulos vão para o cérebro que os interpreta elaborando conceitos de forma, proporção, posição e orientação. Toda nova informação que chega ao cérebro é interpretada, classificada, comparada e armazenada à luz dos conceitos tridimensionais já formulados. Assim, todo esse processamento leva à formulação de novos conceitos ou ao refinamento dos já existentes. Por outro lado, considerando-se que as peças mecânicas, ou ainda os modelos didáticos, têm formas compostas por sólidos elementares, integrais ou parciais (figura 2). A percepção tridimensional da composição dessas formas pode ser desenvolvida por exercício de observação interpretativa (PERTENCE, SANTOS E JARDIM, 2001).



*Figura 2: Imagem gerada a partir de primitivos sólidos
Fonte: Pertence, Santos e Jardim (2001)*

- **Perspectiva:** de acordo como dicionário Aurélio é a arte de representar os objetos sobre um plano tais como se apresentam à vista. Simplificando, pode-se definir que perspectiva é a arte de desenhar o mundo tridimensional em um plano bidimensional de forma correta, como exemplifica a figura 3.



*Figura 3: Objeto num plano intermediário.
Fonte: Enciclopédia Prática Jackson (1972)*

- **Conhecimento prévio do objeto:** Azevedo e Conci (2003) - O conhecimento prévio do tamanho de um objeto serve tanto para determinar a distância absoluta a partir do observador, quanto às distâncias relativas entre os objetos. Além disso, quando há dois ou mais objetos no mesmo campo de visão, e o observador tem noção de seus tamanhos reais, o tamanho aparente serve para determinar qual deles está mais próximo ou mais distante.
- **Oclusão:** obstrução que impede o observador de visualizar parte ou partes de um ou mais objetos em uma cena, ou seja, a parte oculta da cena para a visualização a partir de um determinado ponto de observação (figura 4).



*Figura 4: Exemplo de oclusão
Fonte: Autor*

- **Densidade de texturas:** conhecido como “gradiente de texturas” este fenômeno visual baseia-se no fato de que muitos objetos possuem em sua aparência, algum tipo de padrão com certa regularidade. Em exemplos como

um gramado ou um piso com ladrilhos, onde os padrões estão quase regularmente espaçados, a densidade destes padrões na imagem gera uma importante informação de profundidade. Nestes casos, à medida que os padrões aparecem mais densos e menos detalhados, tanto mais distantes estão do observador (figura 5). Outra informação importante fornecida pela textura é a respeito do movimento dos objetos. No caso da rotação (sem deslocamento) de uma esfera o movimento é muito mais facilmente percebido se esta tiver sobre si a aplicação de uma textura (PINHO, 2009).



Figura 5: Gazebo
Fonte: www.renderosity.com

- **Sombra:** Sombra é uma região escura de uma cena iluminada (figura 6). Esta região escura é determinada pela ausência de energia luminosa em certas regiões da cena. Se um objeto impede a passagem de luz, seja total ou parcial, uma região de sombra é formada. A presença de sombras é uma constante na maioria das cenas iluminadas, e essa presença é fundamental para aumentar o realismo. As sombras estabelecem diversos níveis de profundidade na cena, sendo eficazes em melhorar a impressão espacial na imagem de um objeto tridimensional, pois evitam que os objetos pareçam estar flutuando no ar, fornecem fortes indicações sobre formas, posições e características das superfícies dos objetos (CASSAL, 2001).



Figura 6: exemplo de sombra
Fonte: Autor

- **Reflexão da luz:** a reflexão da luz é um dos fenômenos mais comuns envolvendo a propagação da luz. A reflexão ocorre quando a luz incide sobre a superfície de separação entre dois meios com propriedades distintas (figura 7). A reflexibilidade é a tendência dos raios de voltarem para o mesmo meio de onde vieram. Quando a luz incide sobre uma superfície separando dois meios, podem ocorrer dois fenômenos distintos: reflexão da luz e refração da luz. Parte da luz volta e se propaga no mesmo meio no qual a luz incide (a reflexão da luz). A outra parte da luz passa de um meio para o outro se propagando nesse segundo. A esse último fenômeno (no qual a luz passa de um meio para o outro) damos o nome de refração da luz (<http://efisica.if.usp.br>, 2009).

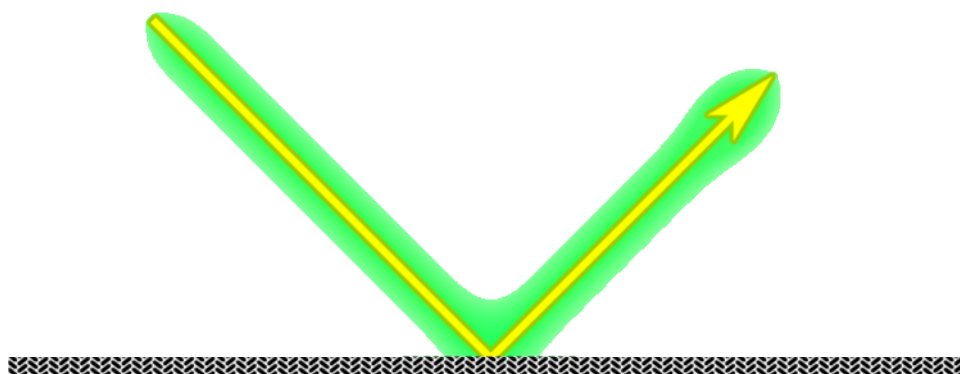


Figura 7: reflexão da luz.

Fonte: Autor

- **Modelagem tridimensional:** Trata da representação computacional de formas geométricas, seu tratamento, recuperação de informações e características (RAPOSO, 2007) (figura 8). Azevedo e Conci (2003) basicamente dividem as técnicas de modelagem em três formas: modelagem manual, automática ou matemática. O método matemático de modelagem usa uma descrição matemática e algoritmos para gerar um objeto. A modelagem manual é sem dúvida o método mais fácil, barato e antigo que utiliza basicamente as medidas de um modelo real e a intuição do modelador. A forma manual foi inicialmente usada pela indústria automobilística e aeronáutica para a concepção e teste de novos modelos. A modelagem automática é sem dúvida a mais sofisticada, porém, a mais rápida e poderosa. Através de equipamentos especiais como scanners 3D, podemos obter o modelo tridimensional de quase tudo.

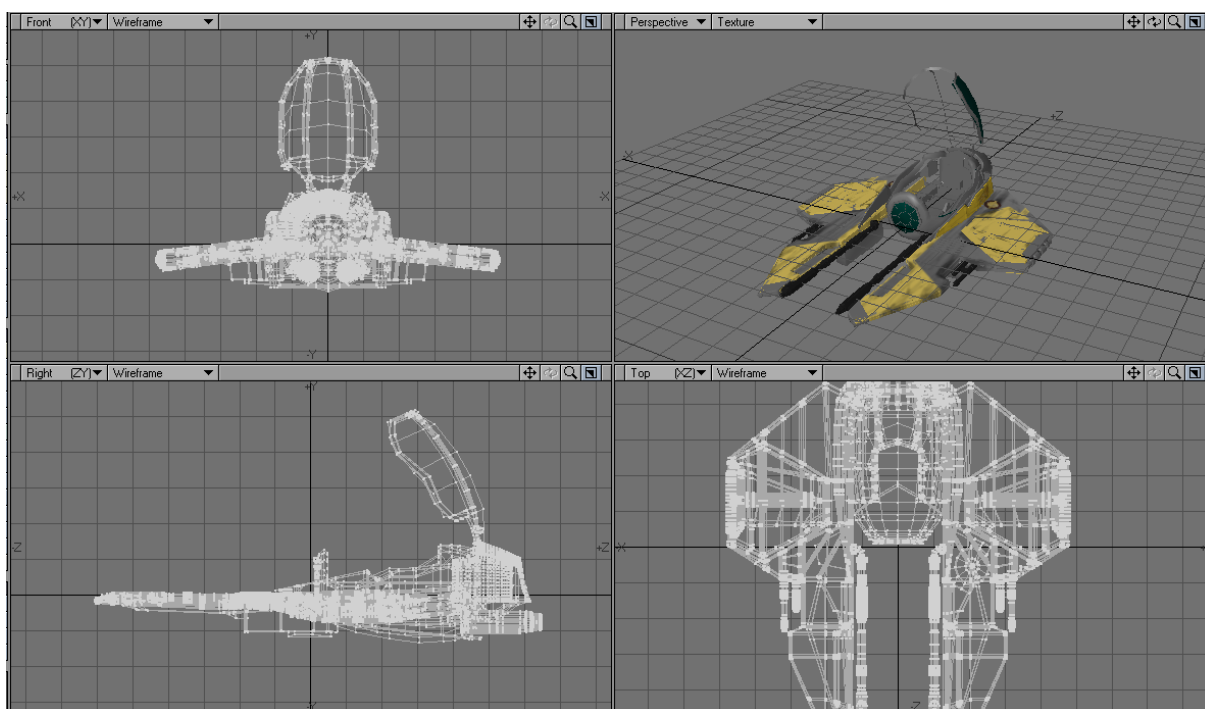


Figura 8: Visões da modelagem de uma nave espacial.

Fonte: Autor

- **Geometria Projetiva²:** "Geometria Projetiva é o estudo das propriedades descritivas das figuras geométricas". A Geometria Projetiva surgiu com a necessidade de formalizar matematicamente as novas tendências artísticas da Renascença, onde a perspectiva começou a ser utilizada em pinturas com o

²Definição de acordo com o site: <<http://www.matematica-na-veia.blogspot.com/2008/01/introduo-historia-da-geometria-projetiva.html>>, acessado em 12 de setembro de 2009

objetivo de retratar as sensações de profundidade nos objetos e cenários. Um exemplo que evidencia as diferenças entre a Geometria Projetiva e a Geometria Euclidiana, e o que ocorre com a interseção de retas. Enquanto na Geometria Euclidiana pode haver retas que não se interceptam, na Geometria Projetiva isto nunca ocorre. Imagine uma estrada de ferro retilínea, os trilhos nunca se cruzam. “No entanto, se fotografarmos tal cenário, ou retratarmos em um quadro, os trilhos parecem interceptar num ponto distante”.

- **Câmeras e modelos de câmeras:** O modelo mais simples de câmera é o chamado modelo pinhole. Tal câmera pode ser realizado por uma caixa fechada, em que se faça um pequeno orifício (figura 13). Os raios luminosos passam por este orifício (o centro óptico) e atingem o plano do fundo da caixa, onde uma imagem se forma. Se este plano for revestido por um material sensível à luz (filme fotográfico ou sensores digitais, por exemplo), é possível registrar a imagem formada.

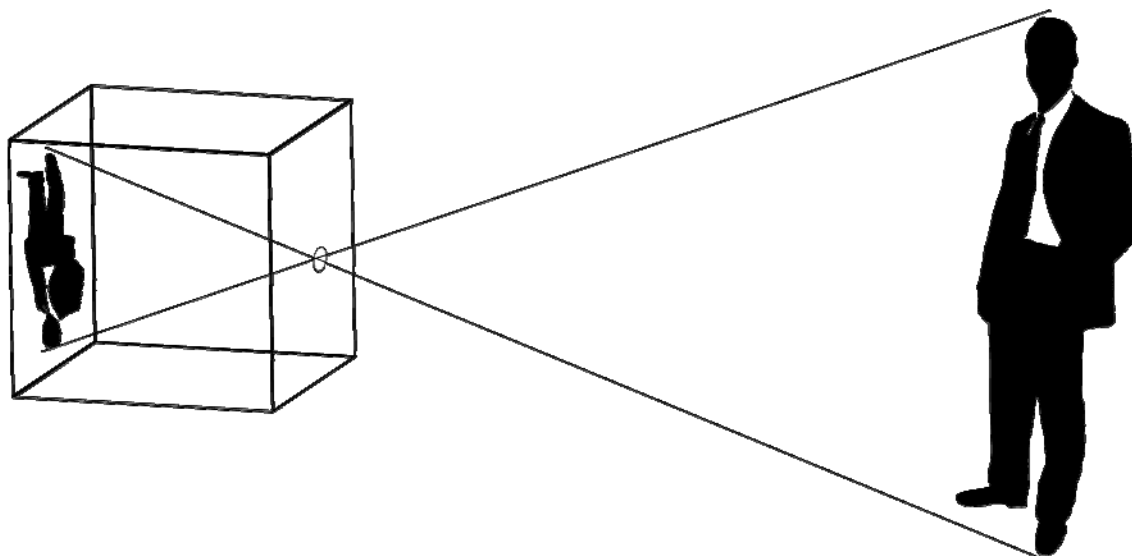


Figura 9: modelo de camera pinhole

Fonte: Autor

- **Calibração:** a determinação dos parâmetros geométricos que participam do modelo matemático que relaciona as dimensões de um objeto no espaço real com a sua imagem fotografada (ANDRADE, 1998 *apud* BAZAN *et al*, 2007). Para o presente trabalho não foi realizada nenhum tipo de calibração ou cálculo relacionado, devido seu objetivo simplista e didático. Para maiores aprofundamentos sobre o assunto pesquise os trabalhos de Tsai (1987) e Zhang (1998), que desenvolveram suas pesquisas com foco na calibração das câmeras, além das referencias acima citadas.

- **Reconstrução no espaço da imagem:** é baseada na reconstrução do objeto no espaço local do dispositivo de captura. a maior parte dos cálculos é feita no sistema de coordenadas local³ do dispositivo de captura. Dessa forma, os dados geométricos e fotométricos são estruturados em partes e a integração global dessas partes é postergada para a última etapa do processo de reconstrução (VELHO *et al*,2005).
- **Reconstrução no espaço da cena:** realiza a reconstrução do objeto diretamente no espaço global da cena. A maioria dos cálculos é feita no sistema de coordenadas global associado ao objeto. Assim, os dados geométricos e fotométricos, logo após serem capturados, são referenciados no sistema de coordenadas global para serem imediatamente integrados na representação do objeto. Um aspecto importante do processo de reconstrução está ligado ao tipo de representação geométrica utilizada no processo. Métodos locais empregam descrições paramétricas que exploram a estrutura dos dispositivos de captura. Métodos globais empregam descrições implícitas baseadas em uma subdivisão do espaço da cena. Muitas vezes, se faz necessária a conversão entre esses dois tipos de representação geométrica. Em geral, tais conversões implicam numa mudança de sistemas de coordenadas – de um referencial local para um referencial global (VELHO *et al*,2005).
- **Geração de malhas:** A representação de objetos através de malha poligonal, chamada de representação por fronteira ou *boundary representation* (ou simplesmente B-REP), é uma forma bastante difundida de se representar superfícies no contexto da síntese de imagens por computador. Esta representação é ideal para a representação de superfícies planas, típicas de elementos facetados como, por exemplo, pirâmides e paralelepípedos. Entretanto, a representação de superfícies curvas como, por exemplo, uma calota esférica, só é possível de forma aproximada. Uma superfície, de acordo com Azevedo e Conci(2003), assim como uma curva, pode ser gerada por famílias de conjuntos de pontos; ter representação analítica; explícita ou implícita; paramétrica ou não-paramétrica e pode-se ainda interpolar, ajustar ou aproximar superfícies a partir de pontos. A figura 10 demonstra a representação de um pato em diferentes níveis de discretização.

³ Coordenadas locais dizem respeito as coordenadas dos dispositivos de captura e coordenadas globais dizem respeito as coordenadas do mundo real onde o objeto se encontra. (VELHO *et al*, 2005)

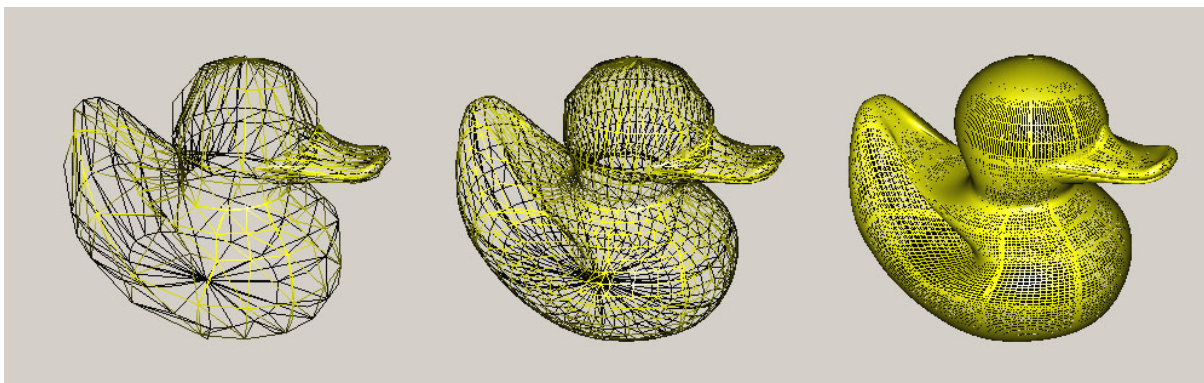


Figura 10: Pato em diferentes níveis de discretização.

Fonte: Autor

Esses conceitos estão relacionados com a subdivisão da computação gráfica, pontuada por Raposo (2007), a saber:

- **Síntese de imagens:** produzir imagens a partir de primitivas geométricas (círculos, linhas, pontos, etc.).
- **Processamento de imagens:** engloba operações realizadas sobre uma imagem para análise, compressão, detecção de padrões não perceptíveis ao olho humano. Essa área está ligada à área de processamento de sinais.
- **Visão computacional:** Estima primitivas geométricas e obtém dados a partir de uma imagem.
- **Modelagem geométrica:** trata da representação computacional de formas geométricas, seu tratamento, recuperação de informações e características.

3.2. Scanner 3D: Conceitos básicos

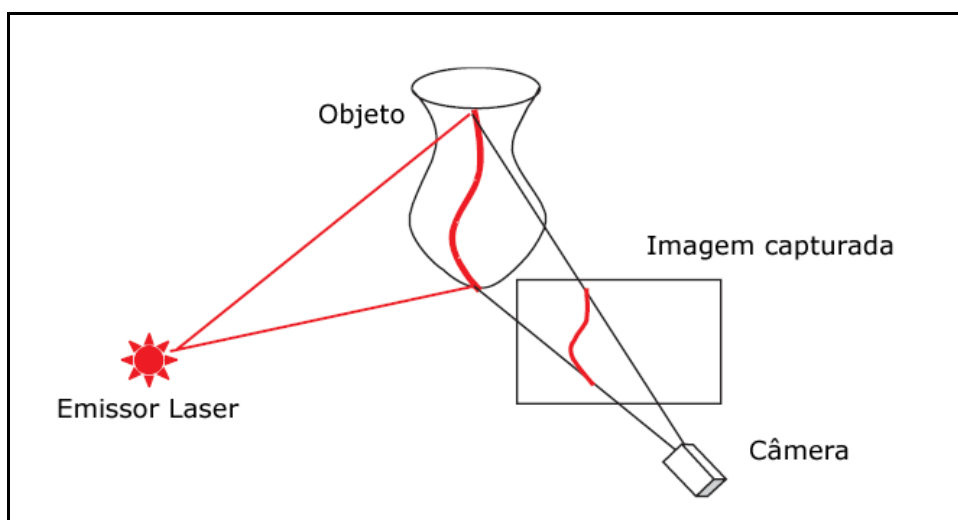
Um scanner 3D tem por finalidade adquirir dados sobre a posição espacial de pontos de uma superfície e transforma-los em uma malha. Essa malha deve, na medida do possível, ser formada por pontos que estejam associados com dados espaciais significativos. Em outras palavras, que representem com determinada precisão as transições de posicionamento no espaço da superfície em estudo (VALENTIN e CONSULARO, 2002).

A densidade de pontos amostrados nessa superfície, a estratégia usada para ordenar um arranjo coerente entre esses pontos, para formar uma malha representativa da superfície e o modelo matemático usado determinam a qualidade do resultado da reconstrução dessa superfície.

O resultado da reconstrução e os dados adquiridos podem ser armazenados em arquivos digitais e usados em programas de visualização 3D, onde podem ser sombreados com a aplicação de cor, textura e iluminação, com a possibilidade de se manipular as formas e dimensões dos elementos que compõe essa superfície digital, além de ser possível criar cenas animadas.

Métodos de aquisição de posições espaciais de superfície seguem duas abordagens principais: o uso de sensores passivos e o uso de sensores ativos. Os sensores passivos não têm fonte de energia própria e não interagem com o objeto, enquanto que os ativos fazem contato com o objeto ou projetam algum tipo de energia nele. Alguns pesquisadores como Morency, Rahime e Darrel (2002) ou Azevedo, Tavares e Vaz (2005), focam seus estudos em dados adquiridos por sensores passivos utilizando duas ou mais imagens radiométricas da superfície, em técnicas conhecidas como visão estereoscópica.

As abordagens utilizadas para realizar a aquisição dos dados para a reconstrução 3D podem utilizar técnicas diversas visando o mesmo objetivo como por exemplo: sombreamento do objeto, imageamento por silhueta (figura 11), triangulação estéreo de pares de imagens e fluxo óptico juntamente com métodos de fatoração de seqüências de vídeo.



*Figura 11 – Digitalização por silhueta
Fonte: Velho et al (2005)*

A taxonomia definida por Curless (1997) e utilizada por Velho *et al* (2005) referente aos métodos utilizados na aquisição de dados de superfície, é:

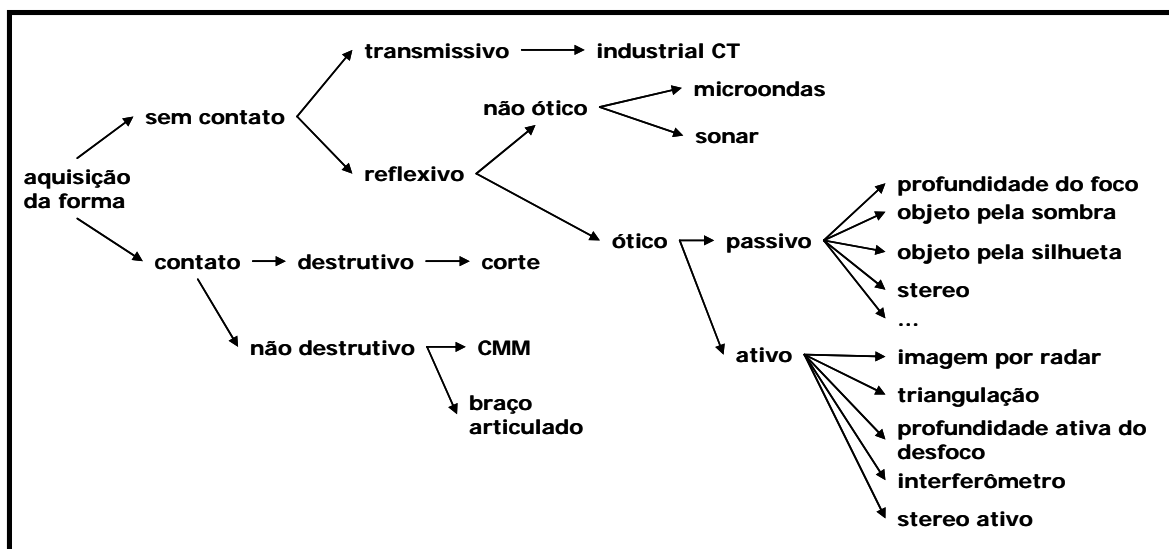


Figura 12 – Taxonomia de tecnologias de digitalização 3D
 Fonte: Velho et al (2005)

Em relação à detecção ativa, podemos identificar duas abordagens diferentes: a) sensores mecânicos e; b) sensores ópticos. Sensores mecânicos são tipicamente sondas que constituem braços com juntas anexadas a um sensor de toque chamado ponteiro. Os ângulos dos braços ou os comprimentos das amarras indicam a localização do ponteiro nos diferentes momentos. Ao tocar o ponteiro na superfície, um evento de contato é sinalizado e a posição do ponteiro é registrada (ALMEIDA *et al*, 2000) (figura 13).



Figura 13 – Digitalizador 3D por contato
 Fonte: Almeida et al(2000)

Sensores ópticos são os que recebem a reflexão de algum tipo de energia luminosa aplicada ao objeto. Tipicamente neste método há aplicação de algum padrão luminoso conhecido sobre o objeto e a coleta desse padrão com algum sensor óptico específico. Essa energia luminosa ou padrão luminoso podem ser distintos, por exemplo: xadrez (preto e branco), linhas verticais ou horizontais, uma única linha de uma cor específica, dentre outras (VELHO *et al*, 2005)(figura 18).



Figura 14: Padrões luminosos utilizados em digitalizadores 3D

Fonte: a) e b) Velho et al (2005), c) Disponível em <www.david-laserscanner.com>

Existem vários sistemas que operam de variadas formas para realizar a aquisição de dados 3D. Sistemas de varredura a laser mensuram a distância do objeto em relação a fonte de emissão, baseando-se no atraso entre a emissão do sinal luminoso e sua recepção (FERNANDES *et al.* 2004 *apud* JOSÉ, M. A. 2008). Esse tipo de scanner é empregado em reconstrução de fachadas e estruturas arquitetônicas. Um exemplo de utilização desse sistema é mostrado na Figura 15.



Figura 15 - Dados do Laser Range Scanner Riegl LMS Z-360I com informação de textura associada.

Fonte: disponível em <www.artescan.net>

Da teoria à prática

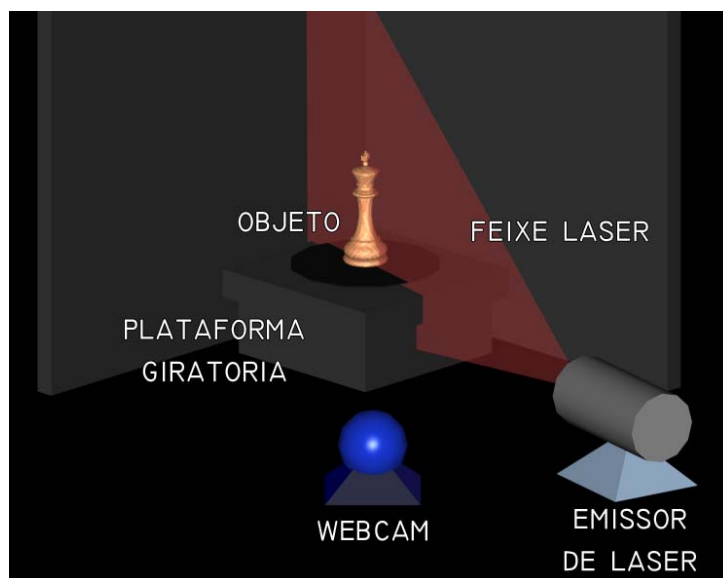
4.1. Considerações Iniciais

O capítulo 2 mostrou alguns fatores envolvidos no processo de ensino e aprendizagem e a importância dos recursos de ensino, que envolvam objetos concretos, motivando o aprendiz e criando condições que facilitam a aprendizagem. Levando-se em consideração que ensinar é criar condições, flexibilizar situações que favoreçam a aprendizagem é que o presente trabalho mostra como é possível construir um scanner simplificado que pode ser utilizado como recurso didático na mediação escolar de conteúdos da disciplina de computação gráfica.

A ferramenta desenvolvida propõe um método de reconstrução digital que contempla de forma bem definida as distintas etapas envolvidas com esse processo. A parte lógica desse sistema apresenta um código aberto e, comentado, o que faz com que o utilizador possa entender o código e possa alterá-lo para atingir objetivos específicos. A parte física envolve uma plataforma composta por um motor de passo, uma fonte de emissão de raio laser, uma câmera fotográfica (webcam) e um microcomputador como recursos básico integrados entre si (figura 16).

A idéia é usar a plataforma giratória para abrigar um objeto que se deseja modelar. A fonte de luz incide sobre o objeto, produzindo uma silhueta sobre o mesmo. Conforme o objeto gira, pela ação da plataforma, são adquiridas imagens. Cada imagem registra a forma da silhueta do feixe de luz projetado. A plataforma giratória completará 360 graus colhendo discretamente a

respectiva imagem a cada intervalo de graus girados produzindo assim faixas de delineamentos do objeto.



*Figura 16: Maquete eletrônica do scanner 3D
Fonte: Autor*

Nesse processo é possível destacar algumas etapas: montagem da plataforma giratória, aquisição de imagens correspondentes a cada giro da plataforma, processamento das imagens, aquisição e processamento das silhuetas, modelagem, visualização intermediária e visualização final. As figuras 16, 17 e o diagrama mostrado na figura 18 ilustram o processo.



Figura 17: Primeiro protótipo do scanner 3D

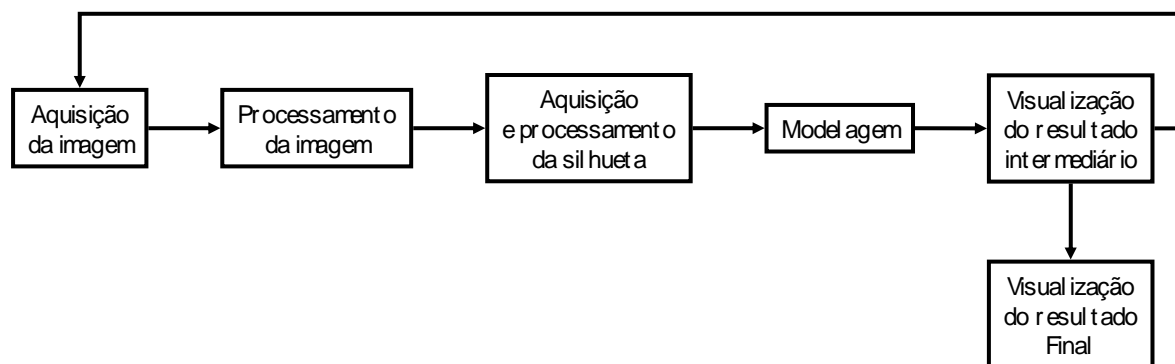


Figura 18: Diagrama de blocos do software principal

Fonte: Autor

O processo de “aquisição de imagens” envolve o acionamento dos componentes de hardware do scanner. É nessa etapa que o sistema envia informações para a plataforma girar o objeto a ser digitalizado, por meio de um cabo serial e para adquirir as imagens pelo uso de uma *webcam*.

A etapa de “processamento de imagem” contempla um algoritmo simples para realçar, segmentar e obter o eixo central da silhueta promovida pela incidência do feixe laser sobre a superfície do objeto.

Em seguida, após o processamento das silhueta tem-se a fase de “modelagem”, que realiza alguns cálculos para estimar a reconstrução do objeto.

Para cada imagem capturada, o sistema realiza a “visualização do resultado intermediário”, mostrando os resultados parciais na tela, proporcionando o acompanhamento da digitalização.

Finalmente após o objeto girar 360 graus o sistema reconstrói a superfície do objeto e permite visualiza-lo. Concluindo a digitalização, é possível salvar o arquivo em um formato OBJ⁴.

⁴ O formato OBJ carrega informações sobre a malha poligonal, sobre o mapeamento de texturas e uso de materiais. No formato mais básico, o arquivo é composto por um conjunto de vértices (linhas que começam com “v”), conjunto de normais (linhas que começam com “vn”), conjunto de mapeamentos de texturas (linhas que começam com “vt”) e conjunto de faces (linhas que começam com “f”). É possível ainda que o arquivo seja formado grupos. Neste caso, cada grupo delimita o seu subconjunto de vértices, normais, mapeamentos e faces. Os dados de um grupo são delimitados por uma linha com instrução: “g <identificador nominal>” até encontrar um novo “g”. Na estrutura de

Um fator importante no desenvolvimento desse projeto é a despreocupação com implementações de algoritmos de calibração e filtros complexos, pois dependem de cálculos que são tratados como desafios de implementação a serem apresentados ao aluno, como parte da proposta pedagógica da ferramenta.

4.2. Materiais e métodos

Por se tratar de um projeto com objetivos didáticos, os componentes que compõem o conjunto de hardware e software foram escolhidos de forma a propiciar uma estrutura clara e bem definida, objetivando a facilidade de interação.

O software foi desenvolvido utilizando o sistema operacional *Microsoft Windows XP*, e a IDE (*Integrated Development Environment*) de desenvolvimento *Borland Delphi* versão 7.0, e alguns componentes com licença de software livre. A escolha da IDE de desenvolvimento foi motivada pela utilização da linguagem de programação *object pascal*, que é uma linguagem de programação simples e agrega utilização de orientação a objeto. Além do fato de existir uma grande comunidade de desenvolvedores de componentes, o que facilita muito o desenvolvimento, conseqüentemente, diminui o tempo de desenvolvimento e aumenta a qualidade do software, pois esses componentes já foram utilizados e testados.

Os principais componentes utilizados para o desenvolvimento do software são o *GLScene* versão 1.0.0.0714, disponível para download em www.glscene.org e; o componente *DSPack* versão 2.3.4, disponível em www.prodigy.com. A seguir uma descrição do processo de instalação e utilização desses componentes.

4.2.1. GLScene

O *GLScene* é uma biblioteca 3D baseada em *OpenGL*. Ela fornece componentes visuais e objetos, permitindo a descrição e processamento de cenas 3D de maneira fácil, simples e poderosa.

GLScene não é apenas uma biblioteca de interface para *OpenGL* ou um utilitário, ele

dados do programa, é importante guardar informações sobre o grupo, inclusive, atribuindo um tipo relacionado ao seu nome, para futura classificação e facilitação de mapeamento (textura certa para cada tipo).

creceu e se tornou um conjunto de classes de base, com um motor 3D genérico para desenvolvimento rápido de aplicações. *GLScene* permite-lhe rapidamente desenvolver e processar as cenas 3D sem ter que se aprofundar em *OpenGL*.

Após o download do *GLScene* e a descompactação do arquivo, a estrutura de pastas deve apresentar uma lista contendo todos os ambientes de desenvolvimento suportados (*Borland C Builder 5*, *Borland C Builder 6*, *Delphi 4*, *Delphi 5*, *Delphi 6*, *Delphi 7*, *Delphi 2005*, *Delphi2006*, *Kylix e Lazarus*), onde dentro de cada pasta estão os arquivos para a instalação do componente em sua respectiva IDE. Existirá também uma pasta com o código fonte do componente (*Source*), uma pasta com utilitários (*Utilities*), uma pasta contendo arquivos de ajuda, uma pasta com informações de instalação (*Readme*) e finalmente uma pasta contendo demonstrações práticas.

Para realizar a instalação desse componente no ambiente Delphi, primeiramente deve-se copiar algumas bibliotecas para uma pasta específica do sistema operacional, que no caso do Windows XP é a pasta “C:\Windows\System32”. Os arquivos necessário são:

- *GLScene\Source\GameAPIs\SDL.dll*
- *GLScene \Source\PhysicsAPIs\ode.dll*
- *GLScene \Source\SoundAPIs\bass.dll*
- *GLScene \Source\SoundAPIs\fmmod.dll*

Em seguida deve-se configurar o caminho para as bibliotecas do componente. Para isso deve-se abrir o Delphi e clicar no menu “*Tools*” e logo em seguida escolher a opção “*Environment Options*”. Após realizada essa ação uma tela, como mostra a figura 23, é mostrada e então é escolhida a aba “*Library*”. Em seguida basta clicar no botão da opção “*Library Path*” para abrir a janela onde será buscada a pasta onde o *GLScene* foi descompactado (figuras 19 e 20) e no caso desse projeto a pasta utilizada foi “D:\Projeto Scanner\GLScene”.

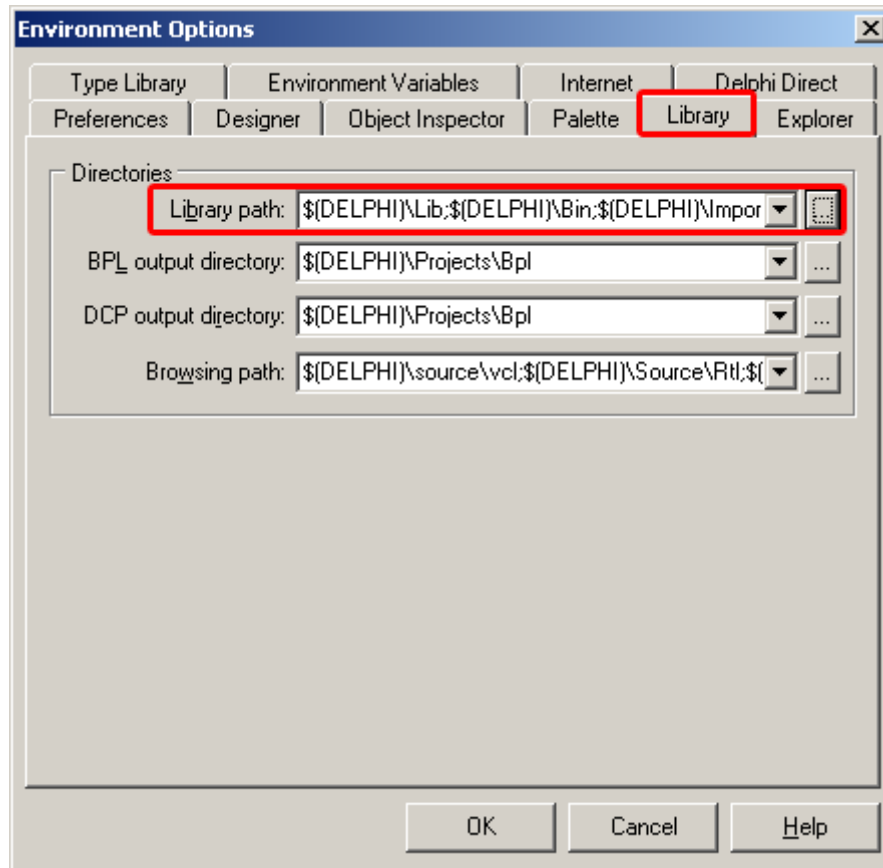


Figura 19: Tela de opções do ambiente de desenvolvimento Delphi

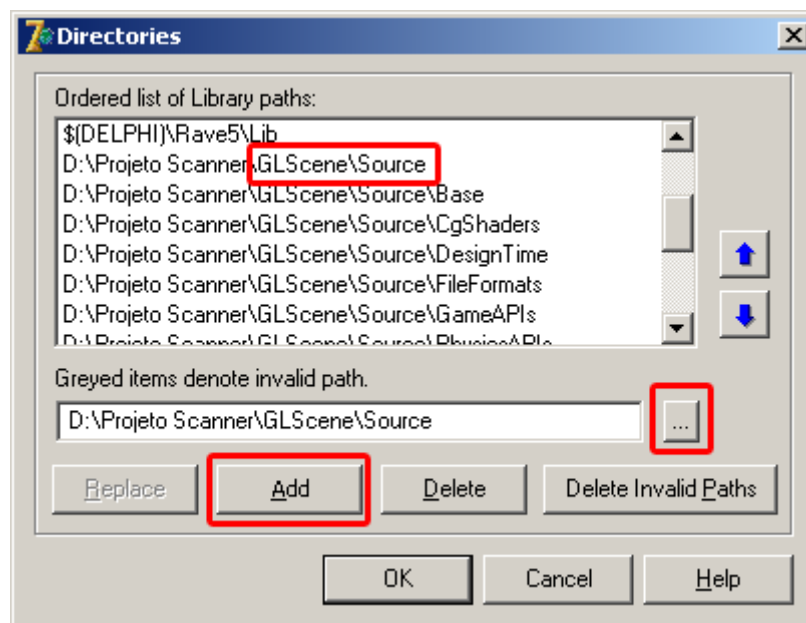


Figura 20: Tela de adição de pastas no caminho de busca das bibliotecas

Após aberta a janela com a árvore de pastas, basta selecionar a pasta desejada e clicar em “OK”. É necessário que todas as pastas que aparecem na figura 21 sejam adicionadas, inclusive a pasta “Source”.

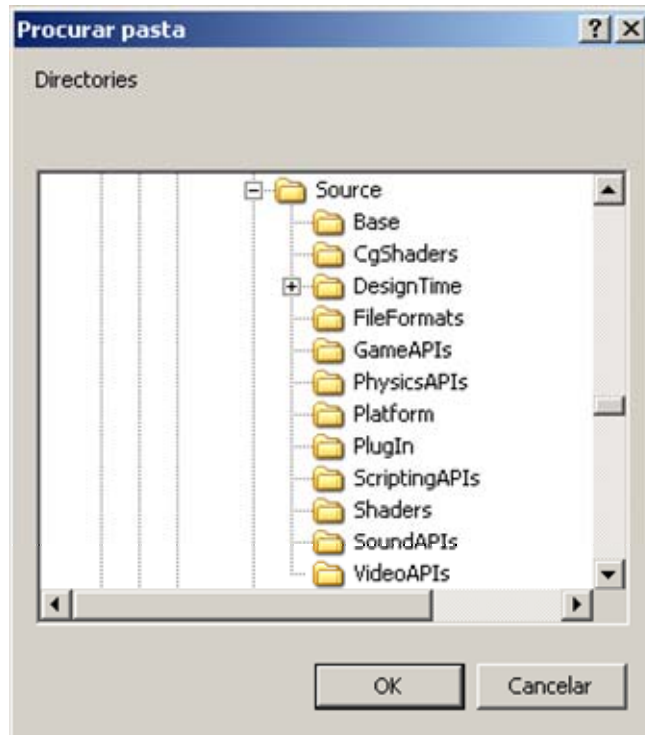


Figura 21: Árvore de pastas da tela de adição de pastas ao caminho de busca de bibliotecas

Depois que os caminhos das bibliotecas estão corretamente configurados no ambiente basta abrir o arquivo “GLScene7.dpk” que se encontra na pasta “GLScene\Delphi7”. Após a abertura desse arquivo é necessário compilá-lo e só depois instalá-lo, como mostra a figura 22.

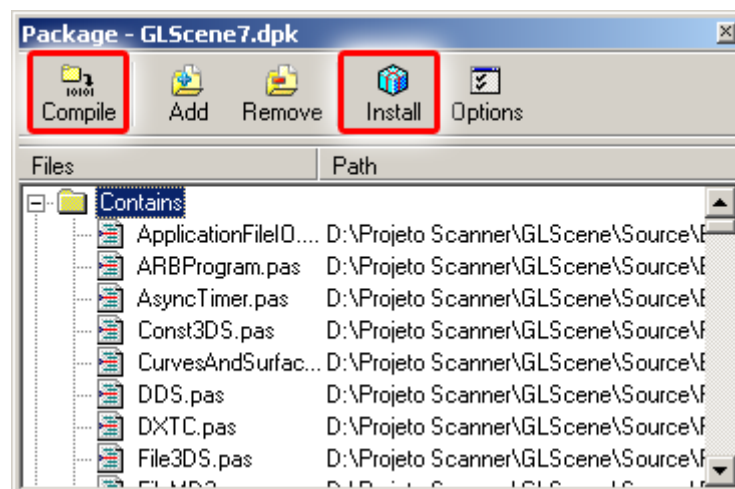


Figura 22: Compilação e instalação do GLScene

Após a instalação, se ocorrer tudo corretamente, o *Delphi* deve retornar uma mensagem de sucesso da instalação e uma lista dos novos componentes disponíveis no sistema como mostram as figuras 27 e 28.



Figura 23: Mensagem de sucesso da instalação do *GLScene*

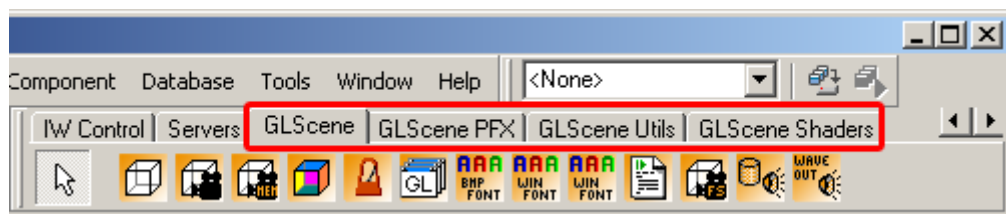


Figura 24: Barra de ferramentas do *GLScene* adicionada ao *Delphi*

Finalizada a instalação é o momento de aprender a empregar alguns componentes visuais que foram utilizadas nesse projeto.

A primeira classe a ser explicada é a *TGLScene*. Ela faz parte do componente base do *GLScene*. Uma instância da classe *TGLScene* contém a descrição da cena (iluminação, geometria, etc.), que é basicamente um gráfico de cena hierárquico feito com o *TGLBaseSceneObject*. Essa classe geralmente contém um ou mais objetos *TGLCamera*, que pode ser consultado por um visualizador de componentes para fins de processamento (*TGLSceneViewer*). Um objeto

TGLCamera define uma câmera no ambiente virtual, por onde se torna possível visualizar os objetos 3D construídos no *GLScene*, por esse motivo ter um objeto *TGLCamera* é indispensável.

Além do objeto *TGLCamera* existem mais 2 itens indispensáveis: o *TGLLightSource*, que define a luz para a visualização do objeto, pois sem esse objeto a imagem fica preta (como no escuro); e um objeto baseado na classe *TGLObjects*, que define os objetos primitivos como círculos, quadrados, triângulos, pontos, planos, cubos e polígonos; ou um objeto baseado na classe *TGLMesh*, que define malhas e objetos baseados em malhas como malhas simples, atores, planos irregulares e formas livres.

Objetos da cena podem ser acessados diretamente a partir do código *Delphi* (como componentes regulares), mas existem aqueles que são editados com um editor específico (duplo-clique no componente *TGLScene* em tempo de design para chamá-lo) como mostra a figura 25. Para adicionar objetos em tempo de execução, use o método *AddNewChild* da classe *TGLBaseSceneObject*.

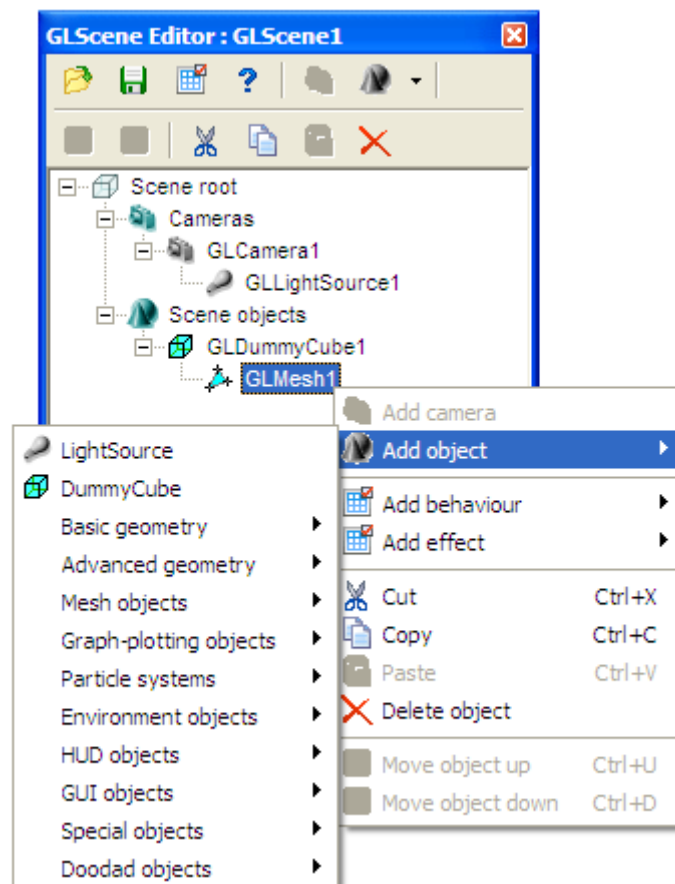


Figura 25: *GLScene Editor* – editor específico de um objeto *TGLScene*

A utilização desse componente nesse projeto é realizada de forma simples. Após configurar algumas opções de cor (cores claras para as arestas, vértices e faces) e modo de junção dos pontos do objeto (definir se a união dos pontos irá ocorrer a cada 3 pontos ou a cada 4 pontos), basta adicionar ao código instruções para adicionar os vértices, sempre lembrando que a inclusão de vértices ocorre sempre em sentido anti-horário em relação a visualização da face desejada, como pode ser visualizado na figura 26.

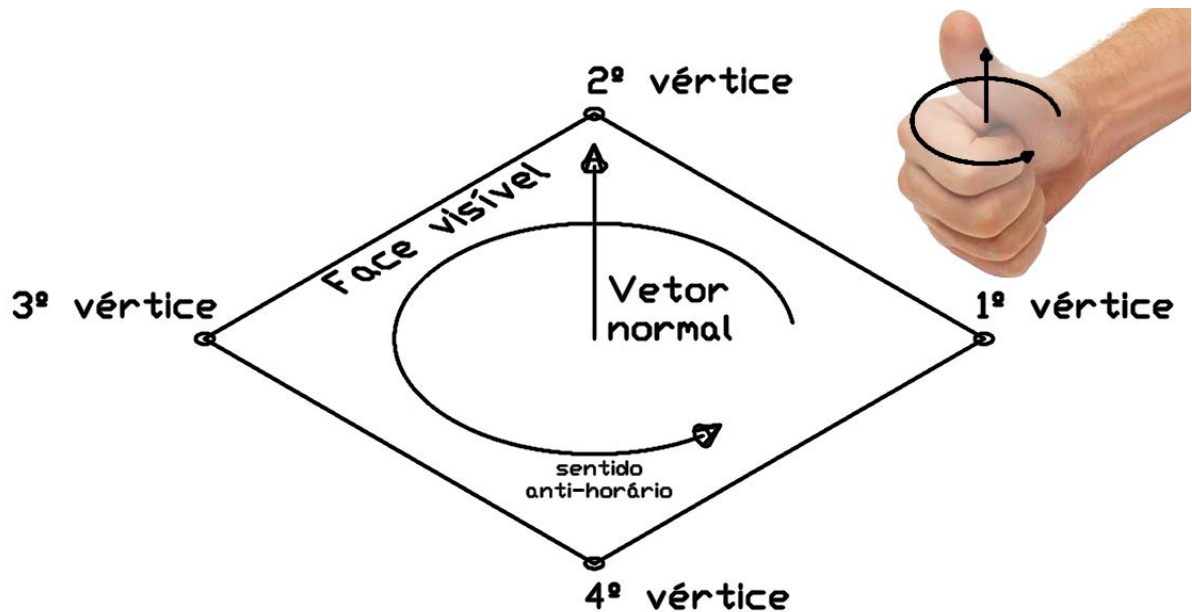


Figura 26: Ordem de plotagem dos vértices em relação ao vetor normal
Fonte: Autor

Outra classe necessária é a *TGLSceneViewer*, na qual são configuradas as variáveis que definem algumas propriedades do *canvas* onde é apresentado o resultado. É uma classe simples onde poucas propriedades são ajustadas antes da utilização, como a cor de fundo (como o objeto foi definido anteriormente com cores claras a cor de fundo deve ser escura), e a câmera virtual que está apontada para a cena.

4.2.2. DSPack

O segundo grupo de componentes utilizado é o *DSPack*, que é um conjunto de componentes e de classes para escrever aplicações multimídia utilizando o *Microsoft Direct Show* e tecnologias *DirectX*. O *DSPack* foi projetado para trabalhar com *DirectX 9* em plataformas Win9X, ME, 2000 e sistemas operacionais *Windows XP/Vista* através das IDEs *Delphi 5, 6, 7* e *Borland C Builder 6*. No projeto do scanner 3D o *DSPack* se encaixa fazendo a interface entre o sensor que captura as imagens com o computador, ou seja, a webcam acoplada ao sistema.

A árvore de arquivos do *DSPack* apresenta a seguinte configuração: uma pasta contendo demonstrações (*Demo*), uma pasta de ajuda (*Help*), uma pasta com arquivos de definições para serem incluídos em projetos (*Include*), uma pasta com bibliotecas (*Lib*), uma pasta contendo alguns atalhos de internet (*Redist*), uma pasta com o código fonte (*Src*) e finalmente uma pasta com os pacotes a serem instalados.

Assim como o *GLScene* é necessário adicionar algumas pastas do *DSPack* ao caminho das bibliotecas do *Delphi*, seguindo os passos mostrados nas figuras 50 e 51, adicionando as pastas “Src\DirectX9” e “Src\DSPack” que se encontram na pasta raiz onde o *DSPack* foi descompactado.

O passo seguinte da instalação do *DSPack* requer um pouco mais de atenção, pois deve-se estar atento a ordem de compilação dos arquivos. O primeiro arquivo a ser aberto no Delphi é o “DirectX9_D7.dpk” que está na pasta “*packages*”. Este arquivo deve apenas ser compilado e não deve ser instalado (figura 27).

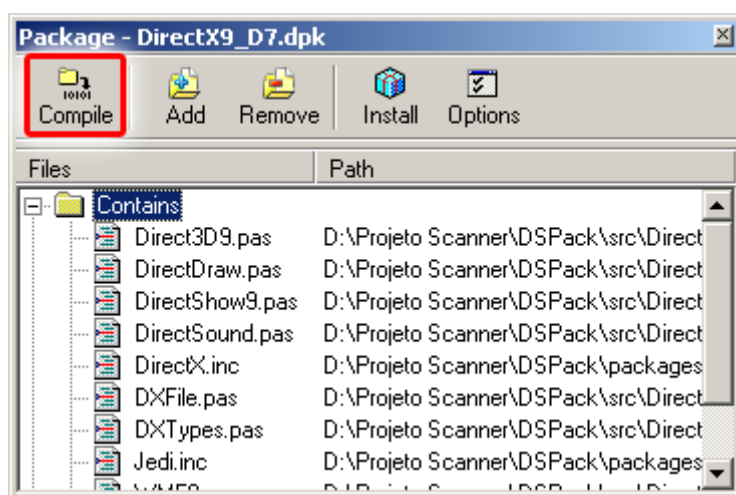


Figura 27: Pacote DirectX9 a ser apenas compilado

O próximo arquivo é o “DSPack_D7.dpk”, que também deve ser apenas compilado como o arquivo anterior. E finalmente o último arquivo é o “DSPackDesign_D7.dpk”, que deve ser compilado e instalado. Se a instalação obteve sucesso, então uma mensagem e uma nova barra de ferramentas devem aparecer como mostra as figuras 28 e 29.

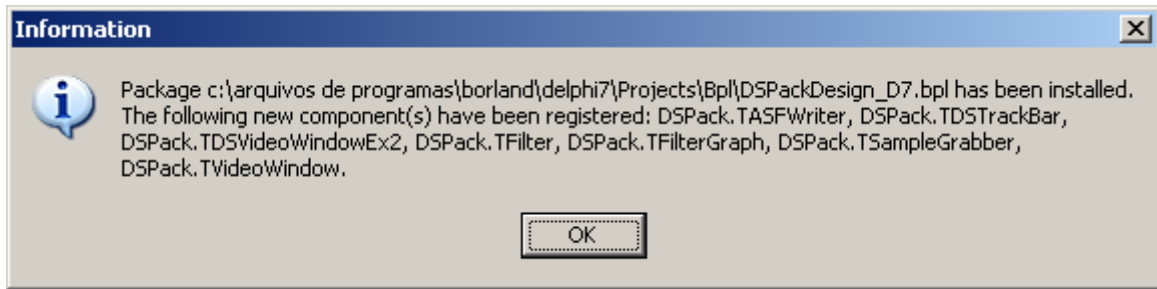


Figura 28: Mensagem de sucesso da instalação do DSPack

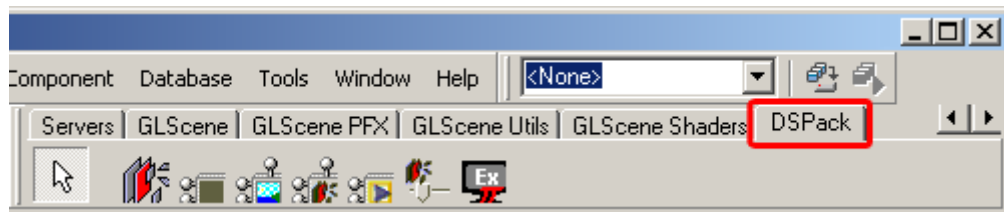


Figura 29: Barra de ferramentas do DSPack adicionada ao Delphi

Finalizada a instalação a tarefa agora é entender o funcionamento das principais funcionalidades. As classes utilizadas no projeto são respectivamente:

- **TFilterGraph:** este é o componente central para utilização do *DirectShow*, o *TFilterGraph* lida com a sincronização, notificação de eventos, e outros aspectos do controle do filtro gráfico e é o componente com o mais simples ajuste para sua aplicação neste contexto, bastando apenas ajusta a propriedade “*Mode*” para a opção “*gmCapture*”, preparando assim para a captura de imagens de um dispositivo externo;
- **TFilter:** este componente é utilizado para adicionar, de maneira fácil, um filtro específico para um filtro gráfico. Em outras palavras, esse componente realiza a interface entre os filtros nativos da *webcam* com o sistema, como por exemplo a definição de tamanho da imagem a ser capturada.
- **TSampleGrabber:** este componente é utilizado para tirar um instantâneo de dados de áudio ou vídeo. Na prática ele é responsável por extrair um quadro de um vídeo ou uma parcela de um fluxo de áudio.
- **TVideoWindow:** gerenciar um processador de vídeo (*Vídeo Renderer*) ou um processador de um misturador de vídeo (*Video Mixer Renderer - VMR*), para mostrar um vídeo na aplicação.

O diagrama de classes do *DSPack*, ilustrado na figura 30, mostra uma visão geral a respeito do componente.

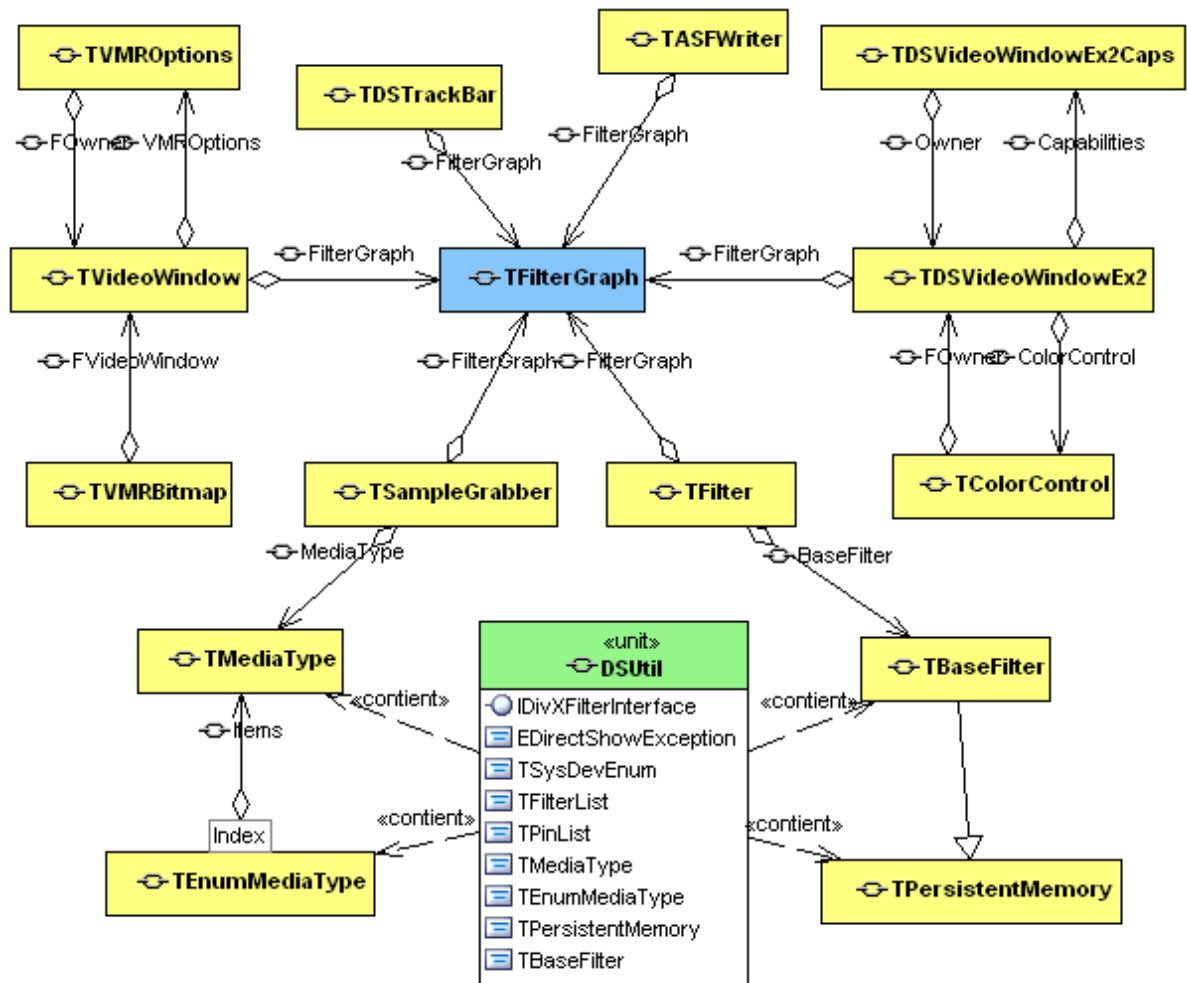


Figura 30: Diagrama de classes do DSPack
Fonte: Arquivo de ajuda do DSPack

Grande parte das configurações do *DSPack* é realizada em tempo de execução, dessa forma o texto não tem intenção de realizar um aprofundamento no assunto, para isso é aconselhável consultar as referências bibliográficas.

4.3. A aquisição da imagem

Aquisição da imagem é a primeira etapa do processo de digitalização. Nessa etapa é realizada a aquisição primária dos dados pelo uso dos sensores da *webcam*. Essa também é a etapa onde um aparato de hardware é necessário, não só a *webcam*, mas também de um sistema capaz de girar uma plataforma sob o objeto a ser digitalizado, por meio do controle digital de um motor.

4.3.1. Motor de passo

Um motor é um dispositivo que transforma energia elétrica em energia mecânica. Quando a corrente elétrica atravessa as bobinas do eixo, as magnetizando, e seus pólos buscam se alinhar com o campo magnético em que estão inseridos, de forma que o eixo realize uma rotação (180° no máximo). Quando o meio giro termina, e o eixo está a ponto de se alinhar, o chaveamento (escovas) do motor invertem o sentido da corrente nas bobinas do eixo, o que faz a rotação prosseguir, completando o giro (180° restantes). O processo se repete continuamente, de forma que a corrente elétrica que passa pelas bobinas sempre mude de sentido, e por tal razão o eixo do motor continua girando, como mostra a figura 31.

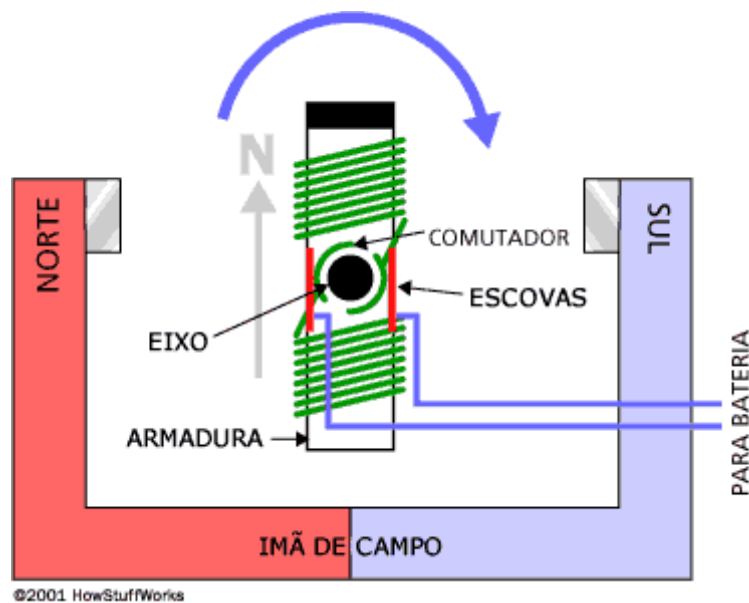


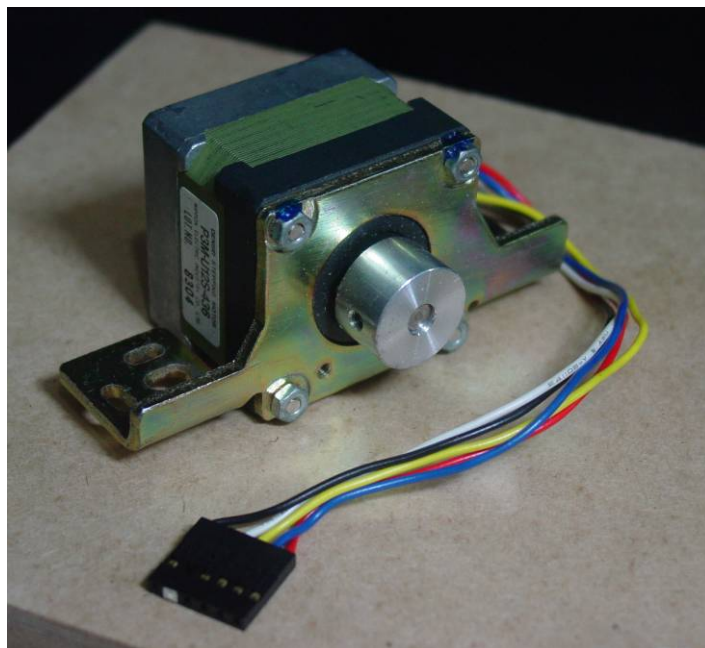
Figura 31: Esquema de funcionamento de um motor
Fonte: Brain (2000)

Os motores em geral são divididos em 2 grandes conjuntos: os motores DC (corrente direta) e motores AC (corrente alternada). Os motores DC são motores onde é possível realizar um controle preciso de velocidade e posicionamento, apesar de ter um custo elevado o seu controle é feito de forma barata, por sistemas transistorizados. Já os motores AC são motores que são utilizados em larga escala onde não é necessário controle de velocidade, apesar de existir um tipo de motor AC conhecido como motor de indução de gaiola, onde o controle de velocidade é possível e está se tornando cada vez mais competitivo em comparação aos motores DC.

Para o objetivo do desenvolvimento do projeto do scanner 3D o modelo escolhido foi um tipo de motor DC: o motor de passo, que de acordo com Leite *et al*(2003), é um transdutor que

converte energia elétrica em movimento controlado através de pulsos, o que possibilita o deslocamento por passo, onde passo é o menor deslocamento angular.

Existem vários tipos de motores de passo disponíveis no mercado, mas o motor de passo escolhido para o trabalho foi um motor de passo unipolar de 5 fios (figura 32). Essa escolha se deve ao fato de ser um motor facilmente encontrado em sucatas de impressoras, além de ser um motor de fácil controle digital.



*Figura 32: Motor de passo unipolar de 5 fios
Fonte: Autor*

O funcionamento desses motores se baseia em uma energização seqüencial das bobinas. A energização somente de uma bobina gera um pequeno deslocamento do eixo do motor, e este deslocamento ocorre pelo fato de o rotor ser magneticamente ativo e a energização das bobinas criarem um campo magnético intenso que atua no sentido de se alinhar com as pás do rotor como mostra a figura 33.

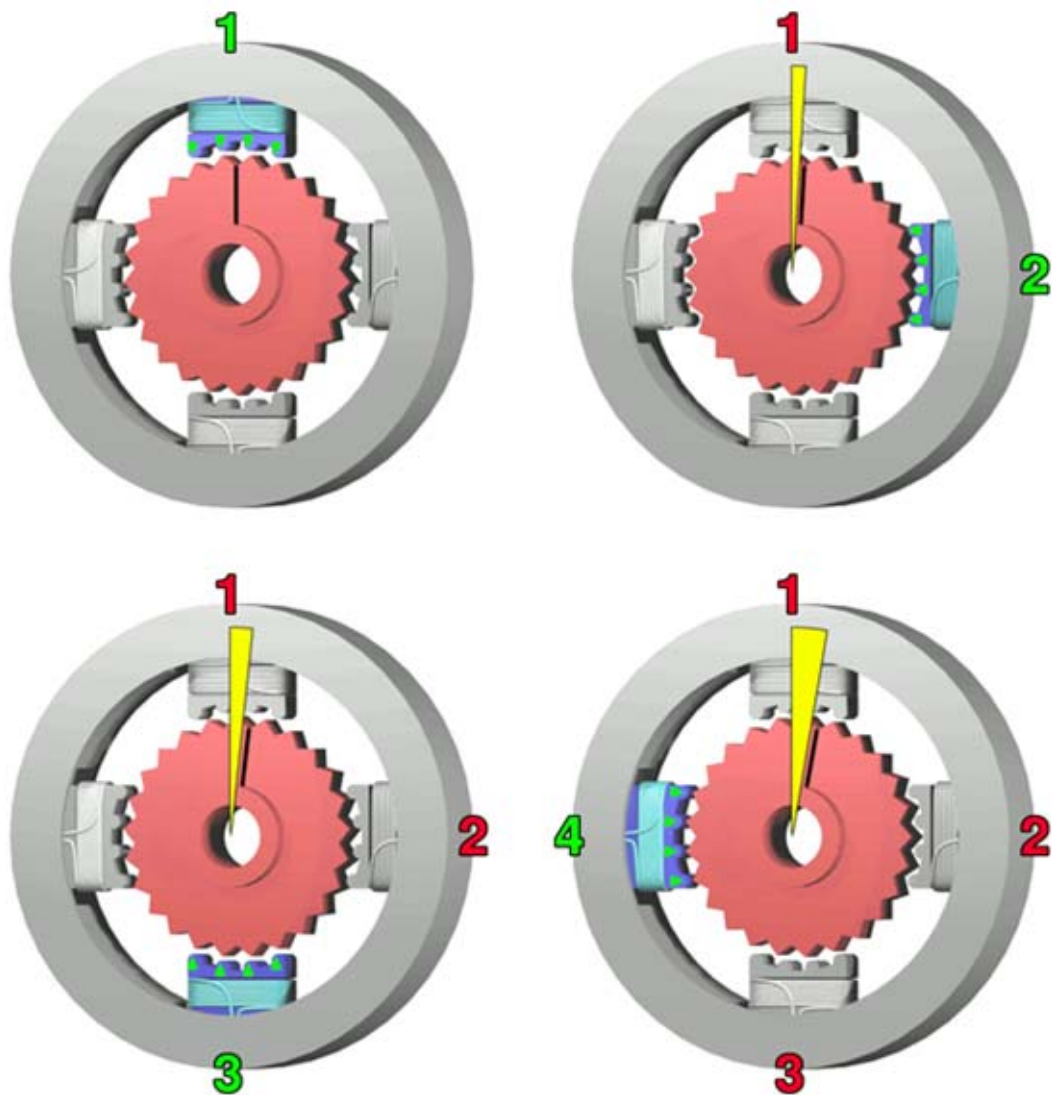


Figura 33: esquema de seqüência de 4 passos de um motor de passo

Seguindo a lógica de funcionamento do motor de passo, é necessário um dispositivo de hardware capaz de energizar os fios de maneira seqüencial e correta, além desse dispositivo poder ser controlado por computador de alguma forma.

4.3.2. Circuito

O hardware utilizado foi produzido com base em um microcontrolador. Esse dispositivo se difere de um microprocessador, pois contém em um único chip, além da unidade central de processamento, memória de programa, memória de dados, conversores AD/DA (sinais digitais para analógicos e vice-versa), temporizadores, etc. (figura 34).

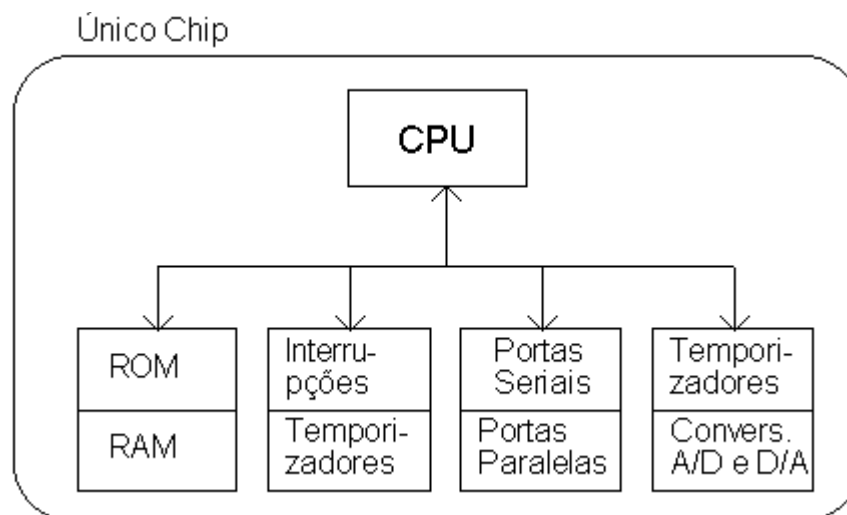


Figura 34: Microcontrolador genérico

Existem diversos tipos de microcontroladores e diversos fabricantes no mercado, mas para este projeto em específico foi utilizado o PIC 16F628A, que é fabricado pela empresa *Microchip*. A escolha derivou-se a partir das necessidades do sistema em questão, tais como:

- **Complexidade de programação:** o microcontrolador deve possuir um conjunto de ferramentas de programação que facilitem a produção do algoritmo a ser embarcado.
- **Controle digital:** o microcontrolador deve possuir portas de E/S (entrada e saída) em quantidade suficiente para a aplicação.
- **Comunicação:** o microcontrolador deve prover algum tipo de comunicação com sistemas de computador.

Os microcontroladores da família PIC de 14 bits⁵, como o utilizado neste trabalho, possuem uma arquitetura do tipo *harvard*, com um barramento para dados e outro para programas, o que possibilita endereçar em apenas uma palavra uma instrução e um operando ao mesmo tempo. O diagrama de blocos do microcontrolador e exemplos de seus registradores são mostrados nas figuras 35 e 36.

⁵ Todas as informações a respeito do microcontrolador PIC foi retirada de seu manual e pode ser encontrada em <www.microchip.com>

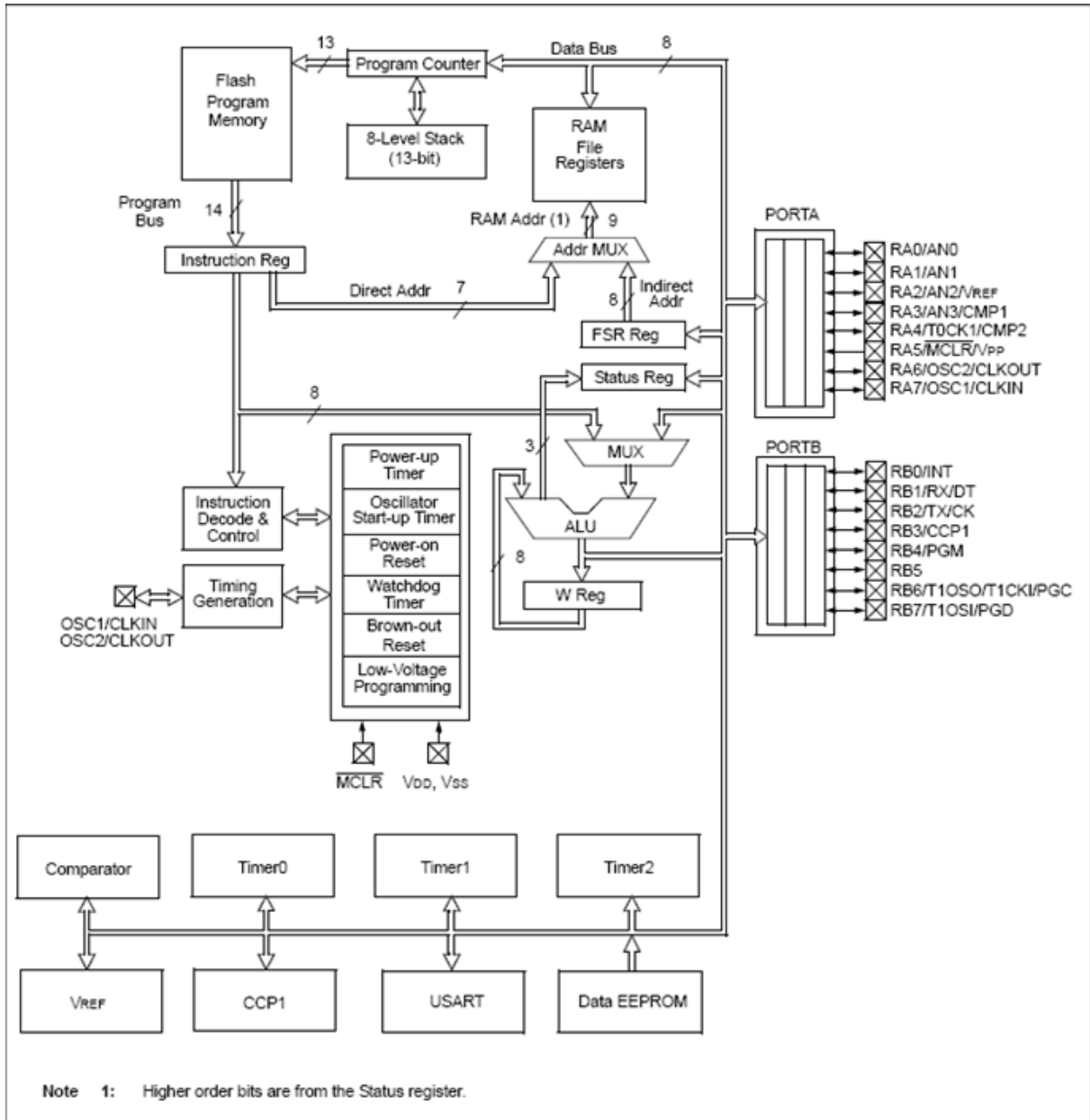


Figura 35: Diagrama de blocos do PIC 16F628A
Fonte:Microchip

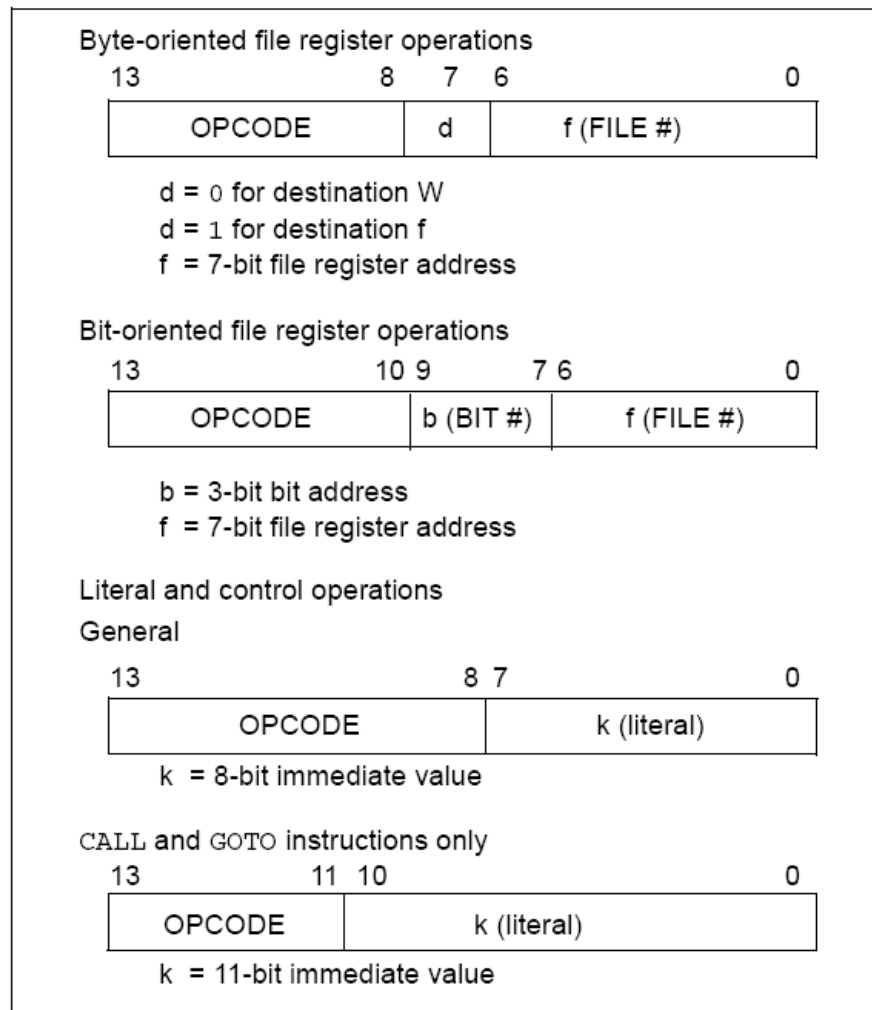


Figura 36: Exemplo de palavras de 14 bits do PIC16F628A
Fonte: Microchip

O ciclo de instruções ocorre a cada 4 períodos do oscilador. No primeiro passo, a instrução é extraída da memória do programa em Q1 e colocada no registrador de instrução em Q4 (figura 37). A decodificação e execução da instrução são realizadas de Q1 a Q4. Os microcontroladores da família PIC de 14 bits dispõem de um *pipeline*⁶ para execução das instruções, enquanto uma instrução é executada, outra instrução é buscada nos registradores.

⁶ É um recurso que possibilita a unidade central de processamento executar mais de uma instrução ao mesmo tempo (TANEMBAUM, 2003)

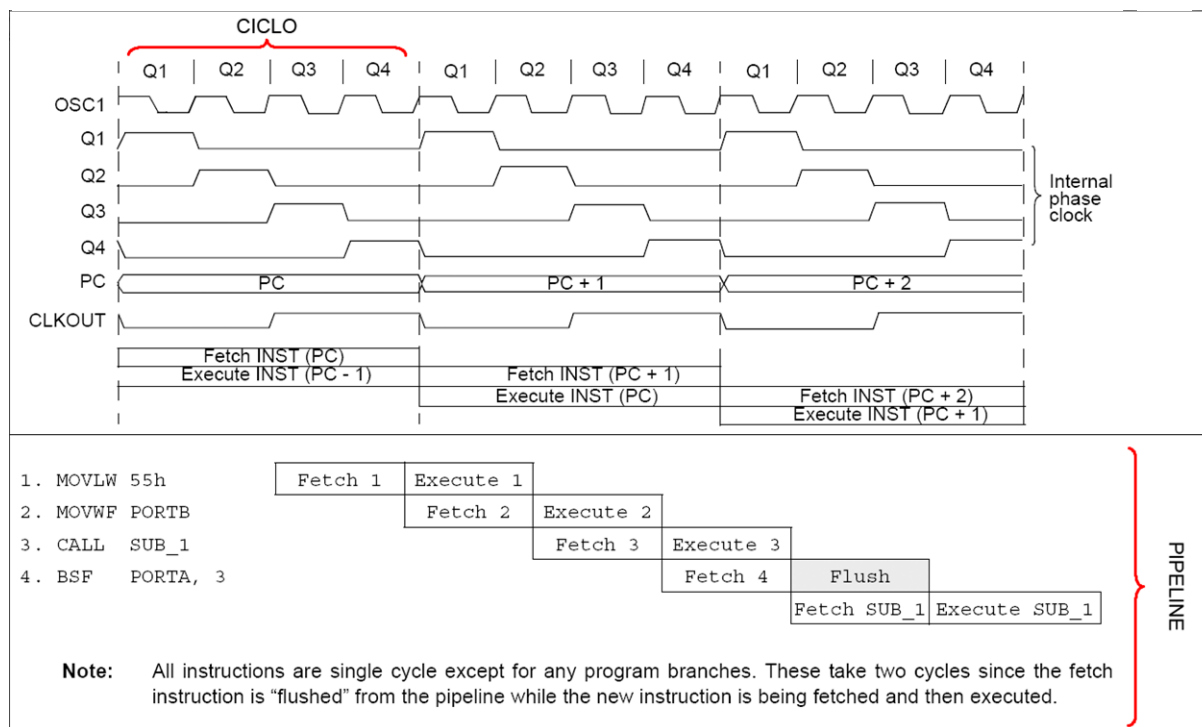


Figura 37: Ciclo de instruções e pipeline do PIC 16F628A

Fonte: Microchip

Existem vários recursos do microcontrolador como: 16 pinos de E/S; velocidade de operação de até 20 MHz (com oscilador externo); oscilador interno de 4 MHz; memória RAM de 224 bytes; memória de dados de 128 bytes (*EEPROM – Electrically-Erasable Programmable Read-Only Memory*); memória de programa de 3584 bytes (2048 palavras de 14 bits); pilha de sistema com 8 níveis; módulo comparador analógico contendo 2 comparadores analógicos e 1 comparador de referência de tensão; 3 temporizadores programáveis (sendo 2 de 8 bits e 1 de 16 bits); módulo PWM (*Pulse-Width Modulation*) no pino RB3 e uma célula de memória de alta durabilidade podendo ser utilizada para até 100.000 gravações para a memória de programa; 1.000.000 gravações para a EEPROM; até 100 anos de retenção de dados na EEPROM e em especial a comunicação serial implementada em hardware, que por sua vez possibilita que a comunicação ocorra em paralelo à execução do programa principal, economizando assim tempo de processamento.

A configuração da pinagem desse microcontrolador (figura 38) é outro fator interessante. Existem 2 conjuntos de portas de E/S nesse microcontrolador: RA e RB. O conjunto RA está disposto em oposição às portas do conjunto RB, separadas pelos pinos Vss e Vdd, o que facilita o projeto do circuito.

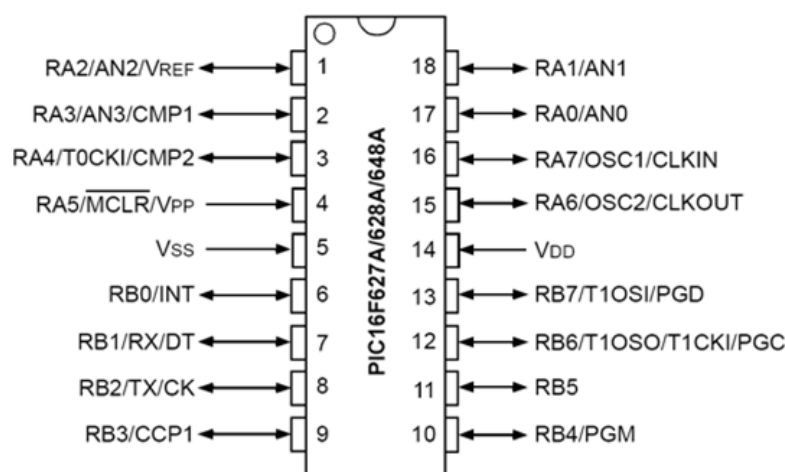


Figura 38: Diagrama de pinagem do PIC16F628A

Fonte: Microchip

Tabela 1: Pinagem do PIC16F628A

Nome	Nº	Descrição
RA0/AN0	17	Porta bidirecional e entrada de comparador analógico
RA1/AN1	18	Porta bidirecional e entrada de comparador analógico
RA2/AN2/V _{REF}	1	Porta bidirecional, entrada de comparador analógico e saída V _{REF}
RA3/AN3/CMP1	2	Porta bidirecional, entrada de comparador analógico e saída do comparador 1
RA4/T0CKI/CMP2	3	Porta bidirecional, entrada de temporizador externo e saída do comparador 2
RA5/MCLR/V _{pp}	4	Porta de entrada, <i>master clear</i> (quando configurada como MCLR/V _{pp} , este pino reinicia o microcontrolador quando colocado em nível lógico 1) e entrada de tensão de programação.
RA6/OSC2/CLKOUT	15	Porta bidirecional, saída de oscilador externo e também saída de <i>clock</i> baseada no <i>clock</i> interno (esta última opção é utilizada para a sincronização com outros componentes)
RA7/OSC1/CLKIN	16	Porta bidirecional, entrada de oscilador externo e também entrada de <i>clock</i> baseada no <i>clock</i> interno (esta última opção é utilizada para a sincronização com outros componentes)
RB0/INT	6	Porta bidirecional e interrupção externa
RB1/RX/DT	7	Porta bidirecional, recepção de comunicação serial e sincronismo de recepção de dados.
RB2/TX/CK	8	Porta bidirecional, transmissão de comunicação serial e sincronismo de transmissão de dados
RB3/CCP1	9	Porta bidirecional e <i>PWM</i>
RB4/PGM	10	Porta bidirecional e porta de programação em baixa tensão
RB5	11	Porta bidirecional
RB6/T1OSO/T1CKI/PGC	12	Porta bidirecional, saída do oscilador do <i>Timer1</i> , entrada de <i>clock</i> externo para o <i>Timer1</i> e <i>clock</i> para programação
RB7/T1OSI/PGD	13	Porta bidirecional, entrada do oscilador do <i>Timer1</i> e E/S de dados para programação

Vss	5	GND
Vdd	14	+5 Vcc

Fonte: Microchip

Uma vez selecionado esses recursos o passo seguinte foi prepara o programa a ser gravado nesse microcontrolador.

O software embarcado no microcontrolador foi produzido utilizando a ferramenta de desenvolvimento *PICC* da *Custom Computer Services* (CCS) que trabalha com linguagem C. Essa ferramenta fornece suporte às ferramentas da *Microchip*, produzindo ainda arquivos de debug compatíveis, provendo total compatibilidade entre as mesmas.

O diagrama de blocos do software desenvolvido para o PIC pode ser visualizado na figura 39.

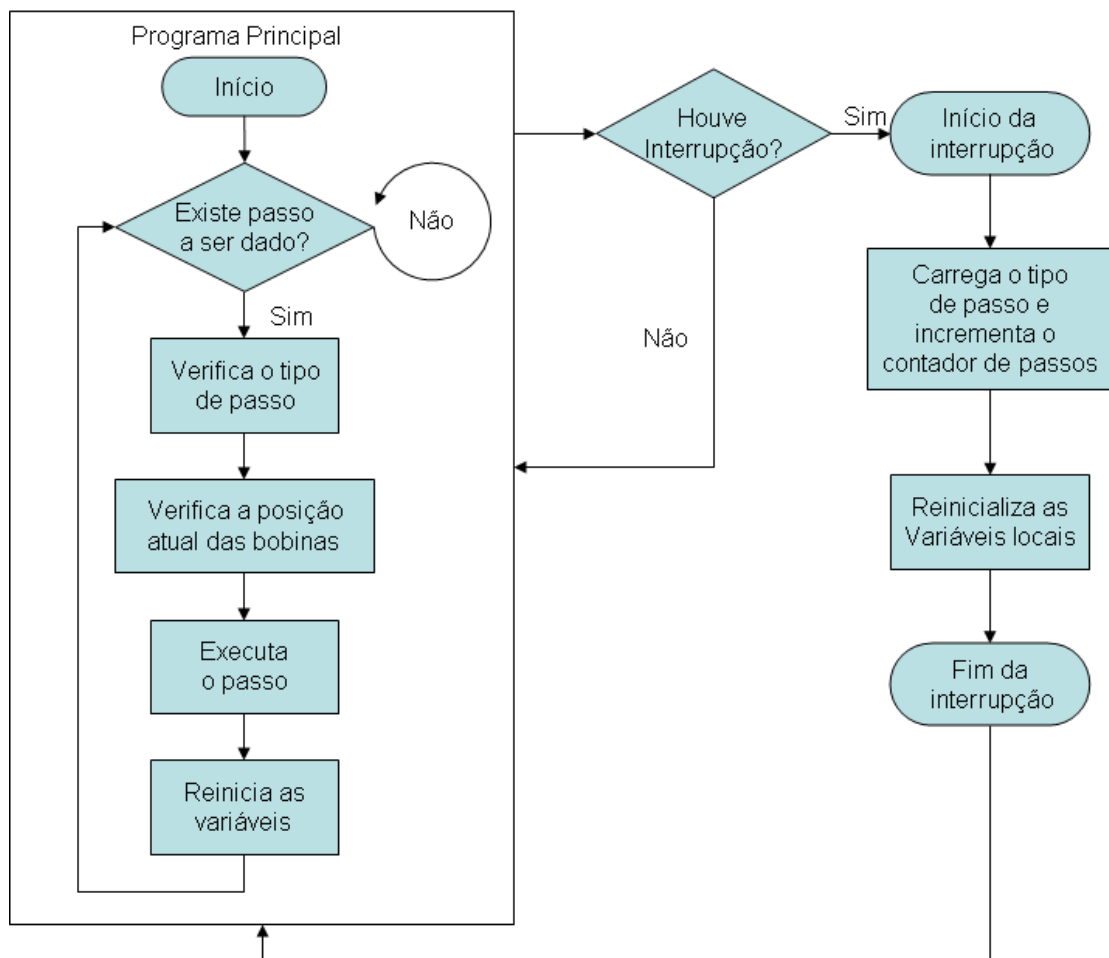


Figura 39: Diagrama de blocos do programa gravado no microcontrolador
Fonte: Autor

O código do programa embarcado é muito simples e enxuto. O funcionamento desse programa foi baseado em um laço infinito onde o programa altera a seqüência de energização das portas assim que chega uma informação válida através da comunicação serial, caso contrário ele não faz nada. A lógica para a energização das bobinas do motor de passo sugere uma seqüência que, se não for respeitada, causa um mau funcionamento do sistema inviabilizando o movimento correto do motor. A seqüência de energização das dos fios do motor utilizado, para a rotação no sentido horário, está descrito a seguir.

Tabela 2: seqüência de energização dos fios do motor de passo

	Fio 1	Fio 2	Fio 3	Fio 4	Fio 5
Passo 1	0	0	1	0	GND
Passo 2	1	0	0	0	GND
Passo 3	0	0	0	1	GND
Passo 4	0	1	0	0	GND

A corrente das portas do microcontrolador é de 5V, quando a porta está em estado lógico “1” (um) e de 0V (zero), não apresentando passagem de corrente, quando a porta está em estado lógico zero. Para o funcionamento do motor de passos utilizado foi necessário que a energização dos fios ocorresse com uma corrente de 12V. Utilizou-se então um circuito integrado (CI) denominado *driver* de corrente, para transformar os sinais de 5V em sinais de 12V. O driver de corrente utilizado nesse projeto foi o ULN2003 (figura 40).

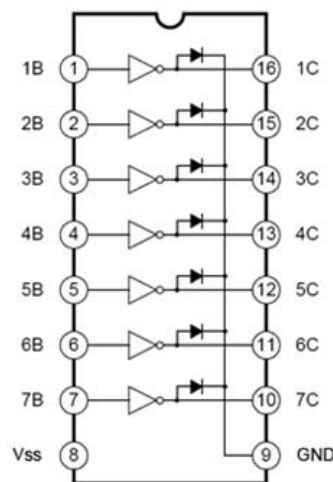


Figura 40: Diagrama de funcionamento das portas do CI ULN2003
Fonte: Texas Instruments

Além do driver de corrente, o sistema também necessitou do auxílio de um outro CI para a comunicação serial, a qual é realizada através de sinais de +12V para valor lógico “1”(um) e -12V para valor lógico “0”(zero) (figura 41).

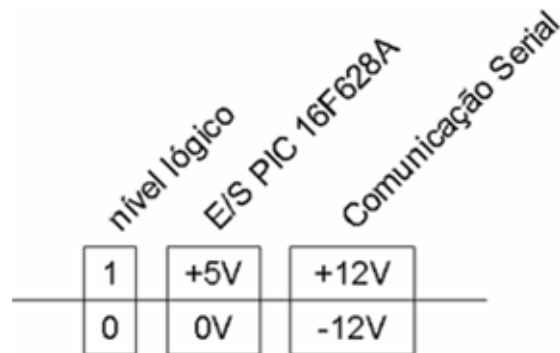


Figura 41: Corrente dos níveis lógicos do PIC 16F628A e da comunicação serial pela norma EIA-232

O CI utilizado foi o MAX232, que trabalha utilizando realizando um “charge pump”⁷ capaz de gerar tensões de +12 volts e -12 volts a partir de uma fonte de alimentação simples de +5 volts, bastando para isso alguns capacitores externos, conforme se pode observar no diagrama apresentado na figura 42 a seguir.

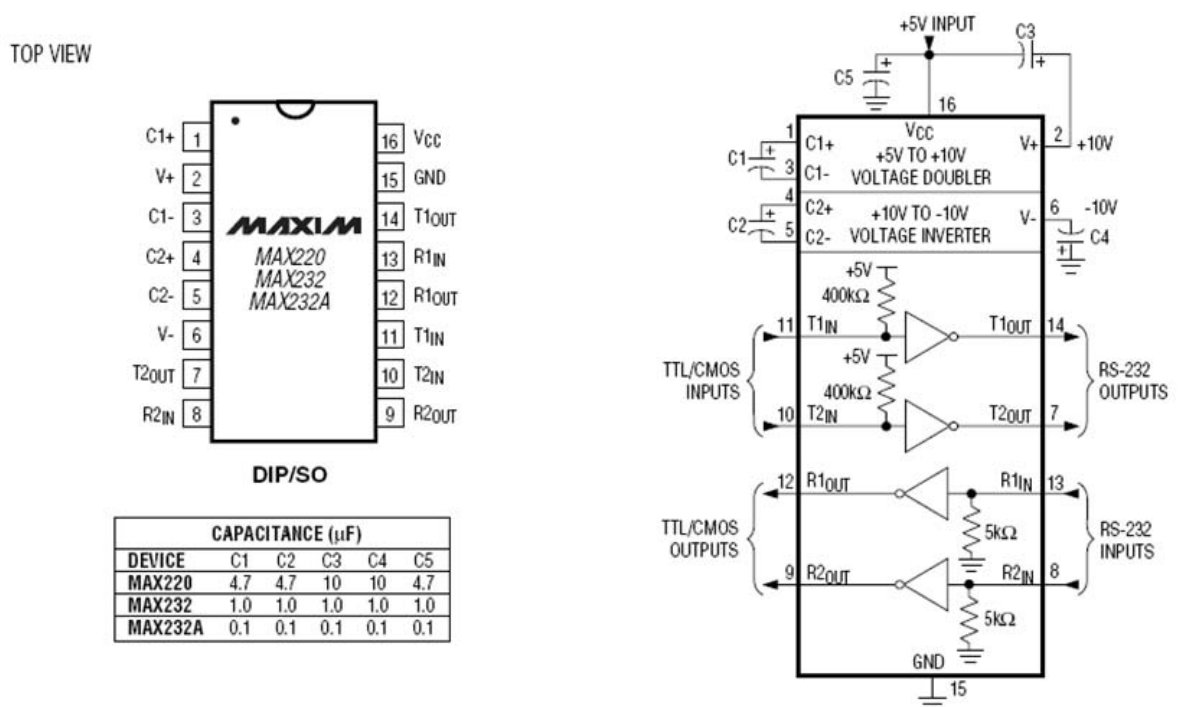


Figura 42: Diagrama de pinagem do CI MAX232
Fonte: Maxim IC

⁷ Um conversor DC/DC que usa capacitores como elementos de armazenamento de energia, para criar uma fonte de energia com maior ou menor tensão.

A comunicação serial, é um tipo de comunicação onde somente 1 único bit por vez é transmitido em um determinado canal de comunicação. Para que isso ocorra de forma satisfatória o sistema emissor da informação quebra o dado em vários bits e os transmite 1 a 1, ficando a cargo do sistema que recebe a informação rearranjar os bits e montar a informação novamente antes de interpretá-la. O padrão de comunicação serial utilizado no sistema é o *EIA232* definido pela “*Electronic Industries Association*” (*EIA*), onde são determinados padrões de tensão, temporizações, funções de sinais, conexões mecânicas e um protocolo de troca de dados.

Se a norma *EIA232* fosse implementada por completo, seria necessário um cabo com 22 fios para o tráfego de dados. Para o sistema proposto é utilizado um tipo de cabo chamado de “*Null-modem*” onde apenas 3 fios são necessários para a comunicação (TxD, RxD e GND).

A norma determina ainda as velocidades padrões para o tráfego das informações. A velocidade é medida em bits por segundo (bps). No padrão *EIA232* a velocidade pode ser de 300, 1200, 2400, 4800, 9600 e 19200 bps. O pacote de dados em uma comunicação serial pode variar de 10 a 11 bits, sendo 8 da mensagem, 1 bit de início de transmissão e 1 para paridade (opcional) e 1 para o final do pacote. Quando o canal de comunicação permanece em repouso, o nível lógico do sinal é igual a “1”(um). Para o início da transmissão o primeiro bit enviado, o chamado “*start bit*”, é um sinal com nível lógico “0”. Seguido do *start bit*, 8 bits de dados são transmitidos na velocidade especificada, logo em seguida vem o bit de paridade e finalmente o bit de parada (*stop bit*) chega, indicando o final do pacote (figura 43).

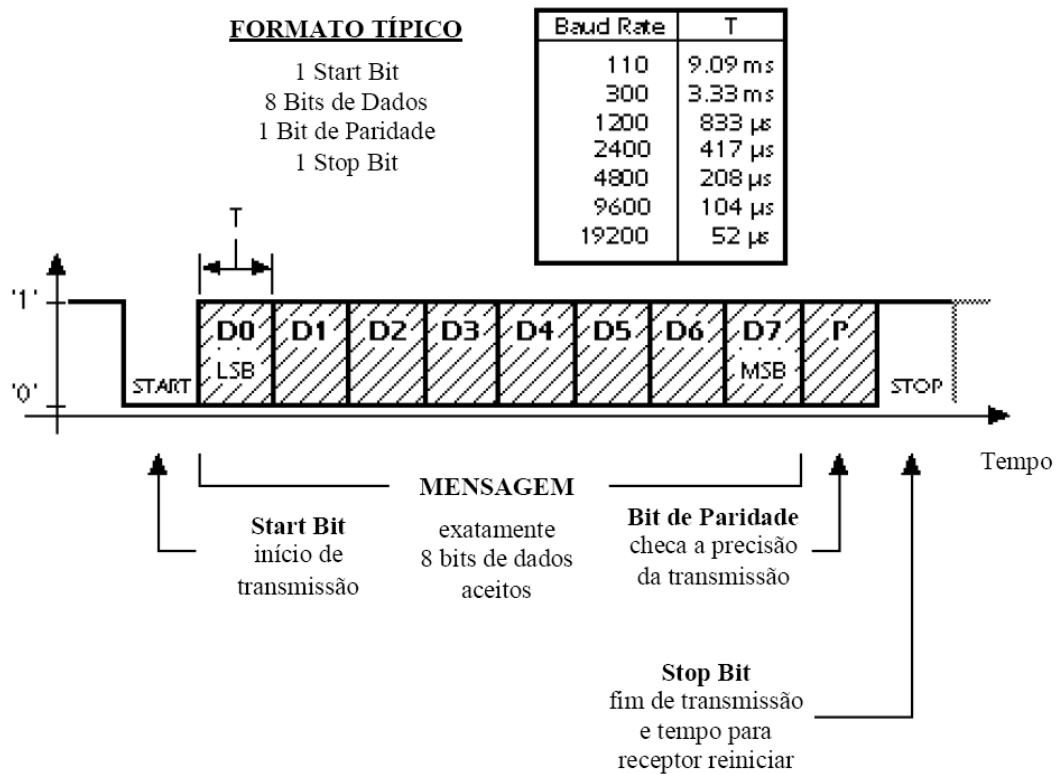


Figura 43: Representação de um pacote de dados transmitido em uma comunicação serial pela norma EIA-232

Fonte: Norma EIA-232

O conjunto de caracteres padrão utilizados na comunicação serial é o *ASCII* (*American Standard Code for Information Interchange*) que contém 255 caracteres, sendo 32 primeiros conhecidos como caractere não imprimível, como consta na tabela abaixo:

Tabela 3: Tabela de caracteres ASCII⁸

Decimal	Binário	Hex	Referência
0	00000000	00	Null - NUL
1	00000001	01	Start of Heading - SOH
2	00000010	02	Start of Text - STX
3	00000011	03	End of Text - ETX
4	00000100	04	End of Transmission - EOT
5	00000101	05	Enquiry - ENQ
6	00000110	06	Acknowledge - ACK
7	00000111	07	Bell, rings terminal bell - BEL
8	00001000	08	BackSpace - BS
9	00001001	09	Horizontal Tab - HT
10	00001010	0A	Line Feed - LF
11	00001011	0B	Vertical Tab - VT
12	00001100	0C	Form Feed - FF
13	00001101	0D	Enter - CR
14	00001110	0E	Shift-Out - SO
15	00001111	0F	Shift-In - SI
16	00010000	10	Data Link Escape - DLE
17	00010001	11	Device Control 1 - D1
18	00010010	12	Device Control 2 - D2
19	00010011	13	Device Control 3 - D3
20	00010100	14	Device Control 4 - D4
21	00010101	15	Negative Acknowledge - NAK
22	00010110	16	Synchronous idle - SYN
23	00010111	17	End Transmission Block - ETB
24	00011000	18	Cancel line - CAN
25	00011001	19	End of Medium - EM
26	00011010	1A	Substitute - SUB
27	00011011	1B	Escape - ESC
28	00011100	1C	File Separator - FS
29	00011101	1D	Group Separator - GS
30	00011110	1E	Record Separator - RS
31	00011111	1F	Unit Separator - US

Fonte: ANSI - American National Standards Institute

O diagrama esquemático do circuito e o desenho da placa de circuito utilizados nesse estudos estão representados nas figuras 44 e 45 a seguir.

⁸ A tabela completa consta no anexo B.

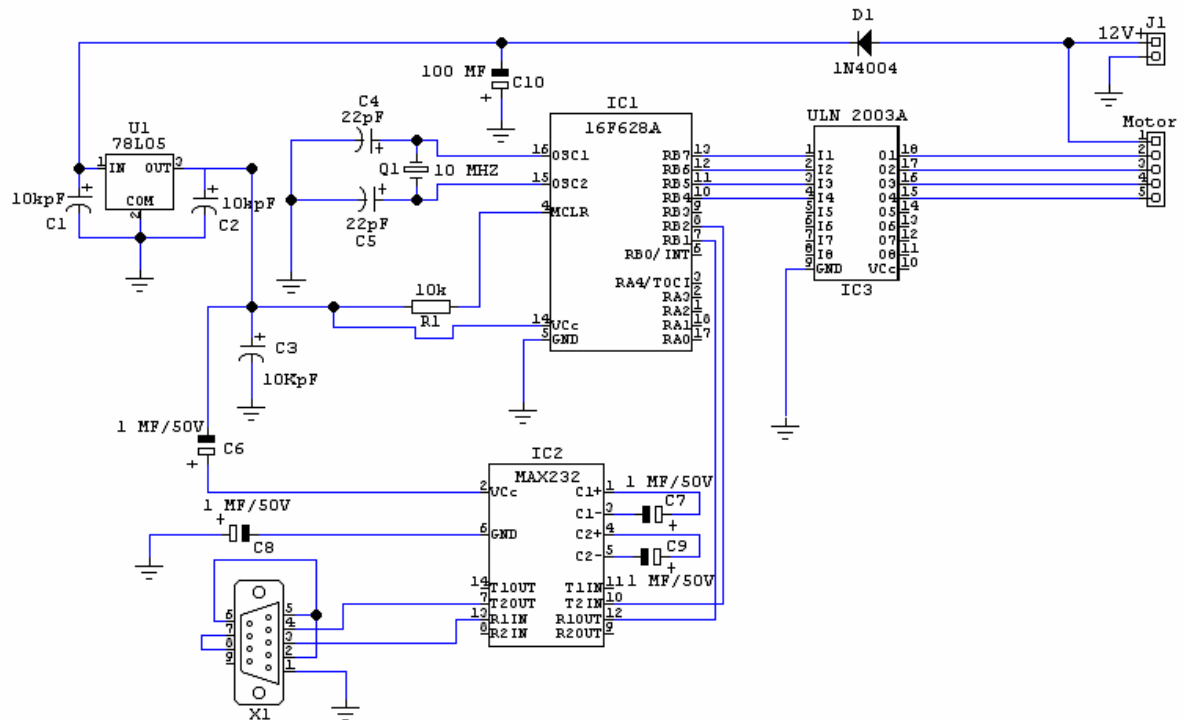


Figura 44: Esquema do circuito da placa controladora de motor de passos

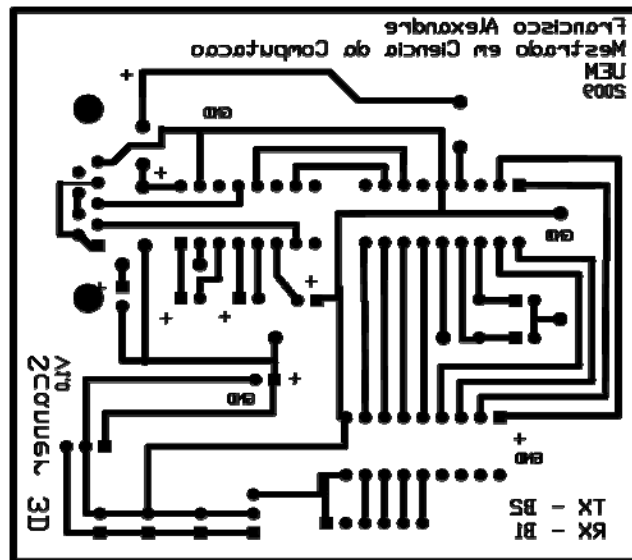


Figura 45: Diagrama do circuito da placa controladora de motor de passos

O algoritmo embarcado na placa controladora realiza um passo do motor sempre que recebe pela porta de comunicação o caractere “W”. Esse caractere é enviado pelo software controlador quando há a necessidade de girar o objeto.

Ao iniciar o processo de digitalização, o software captura uma imagem através da *webcam*, processa-o, modela sua geometria e mostra o resultado parcial, para então enviar o caractere “W” para a placa controladora reiniciando o processo de captura e processamento de imagens novamente.

4.3.3. Utilização da plataforma

Antes de iniciar o processo de aquisição das imagens, é necessário o correto posicionamento do objeto sobre a plataforma giratória, como segue:

- No início do processo de aquisição das imagens, é necessário o correto posicionamento do objeto a partir de uma simples linha de posicionamento como mostra a figura 46.

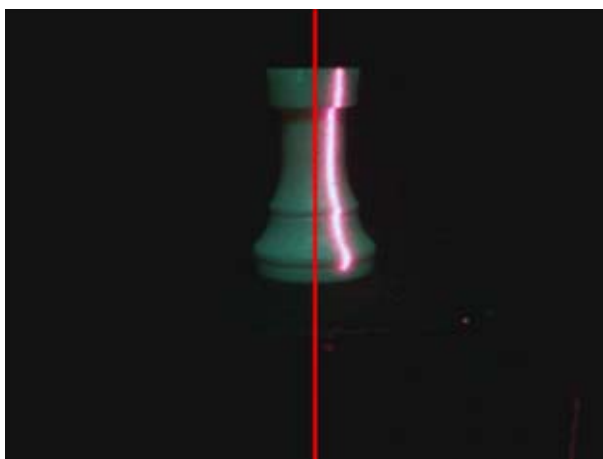


Figura 46: Posicionamento inicial do objeto na plataforma

- Em seguida ao posicionamento, é assumido que o laser está a 45° do centro da imagem.
- Após a obtenção da silhueta o cálculo realizado baseou-se no posicionamento dos pixels brancos em relação ao centro da imagem, definindo o posicionamento de cada pixel da silhueta no modelo a ser gerado (figura 47).

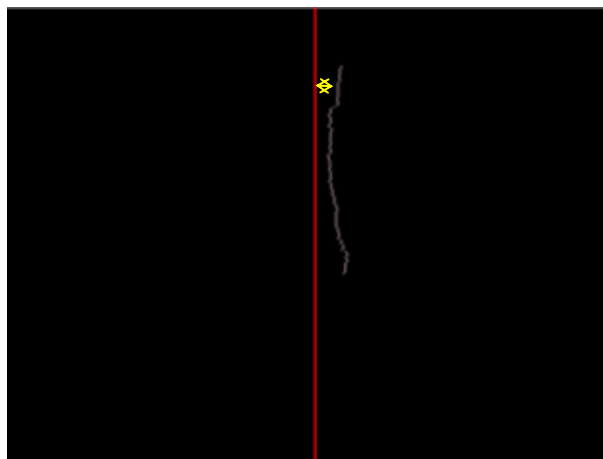


Figura 47: Primeira seqüência de pontos

4.3.4. Captura da Imagem

A cada imagem capturada é realizado um processamento de realce para escurecer e deixar mais evidente o feixe do laser projetado sobre o objeto. Isso é possível pois em um sistema digital uma imagem é representada por uma matriz de pontos chamados pixels (*Picture Elements*). De acordo com PRATT (2007) matematicamente essa representação é realizada pela função $C(x,y,t,\lambda)$. Esta função descreve a distribuição espacial de energia da radiância de uma imagem nas coordenadas espaciais (x,y) , em um tempo t e um comprimento de onda λ , ou seja, a representação digital de uma imagem é uma função que representa a combinação de uma “iluminação” produzida por uma fonte de luz e a energia refletida ou absorvida pelos objetos de uma cena em um tempo determinado de exposição.

Seguindo os passos definidos por PRATT (2007) é possível ilustrar de forma simples como uma imagem digital é formada (figura 48).

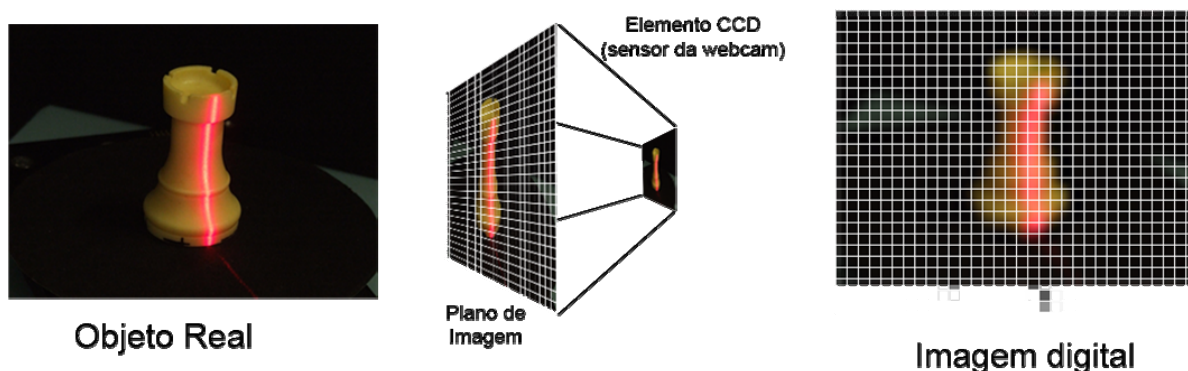


Figura 48: Processo de captura de imagem por um sensor digital

Fonte: Autor

No processo de conversão de uma imagem analógica em uma imagem digital existem 2 etapas importantes segundo PRATT (2007):

- **Quantização:** é a representação do valor discreto correspondente a intensidade da energia captada da cena pelos sensores para cada pixel da imagem.
- **Amostragem:** quando a imagem é amostrada em pontos discretos.

Para que uma função contínua de $C(x,y,t,\lambda)$, como descrita anteriormente, seja capaz de representar uma imagem qualquer se faz necessária a transformação do sinais analógicos em sinais discretos.

De acordo com o Teorema de Nyquist, a quantidade de amostras por unidade de tempo de um sinal, chamada taxa ou frequência de amostragem, deve ser maior que o dobro da maior frequência contida no sinal a ser amostrado, para que possa ser reproduzido integralmente sem erro de aliasing. A metade da frequência de amostragem é chamada frequência de Nyquist e corresponde ao limite máximo de frequência do sinal que pode ser reproduzido. A figura 49 mostra um exemplo de aplicação do teorema de Nyquist, onde é possível constatar que a reconstrução do sinal analógico de forma satisfatória.

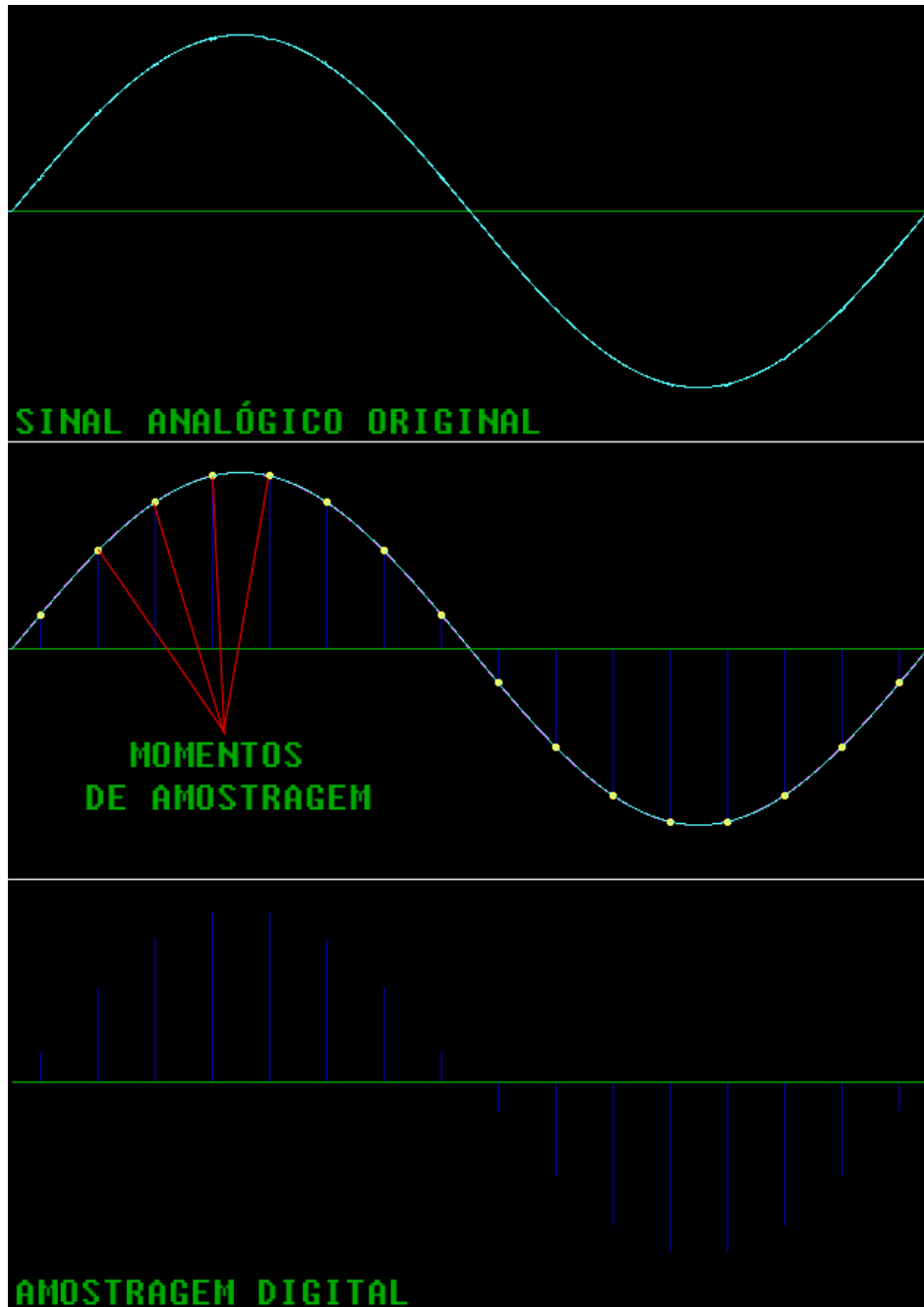


Figura 49: Amostragem de um sinal analógico
Fonte: Autor

Quando o teorema de Nyquist não é obedecido ocorre um fenômeno conhecido como aliasing, como é ilustrado na figura 50.

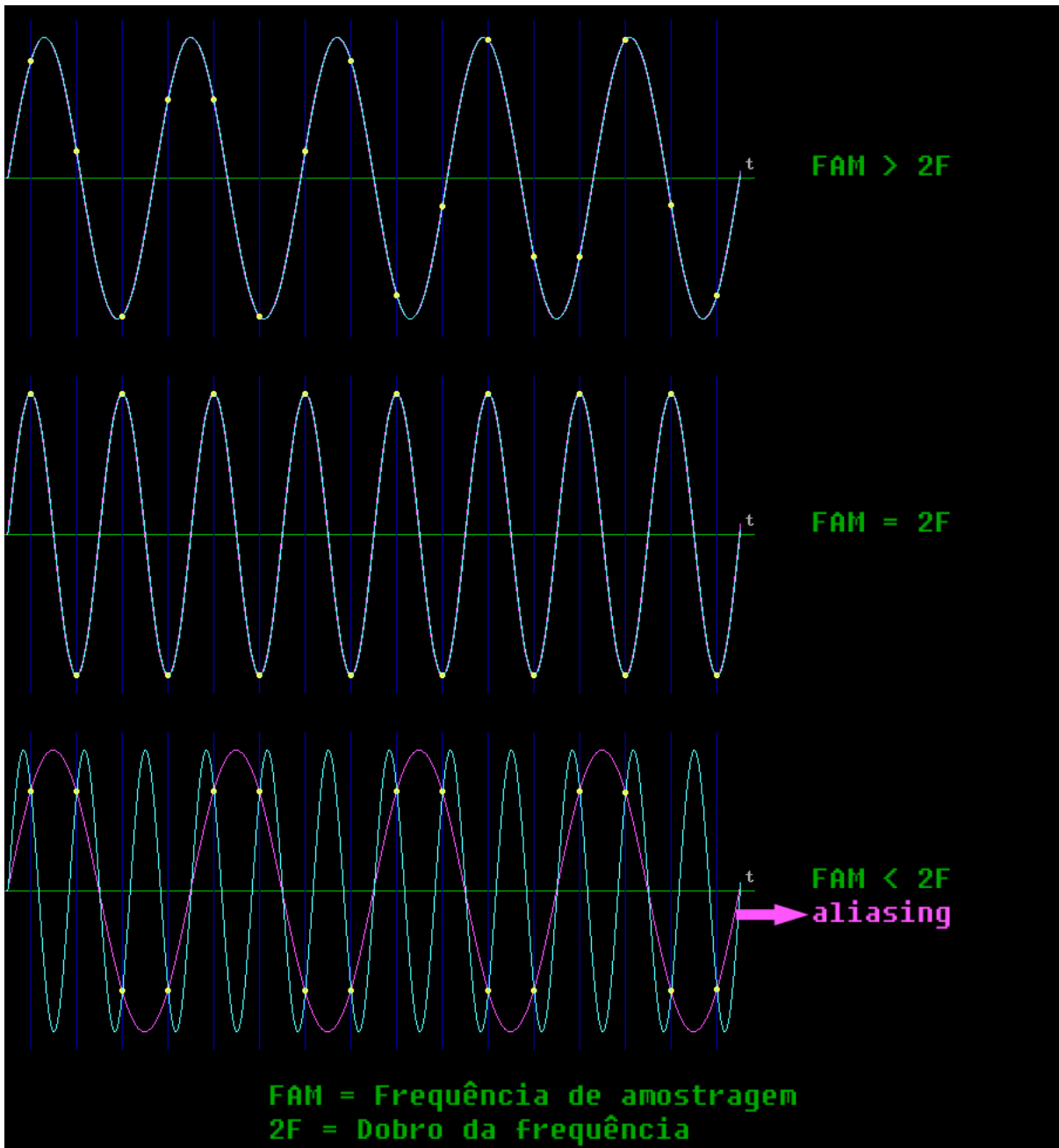


Figura 50: Aliasing
 Fonte: Autor

O *aliasing* é responsável pela quadriculação nas imagens (figura 51), o que significa que a quantidade de dados para a reconstrução do sinal analógico foi insuficiente, pois a seqüência periódica das amostras foi inferior a pelo menos duas vezes a frequência máxima presente no sinal amostrado.

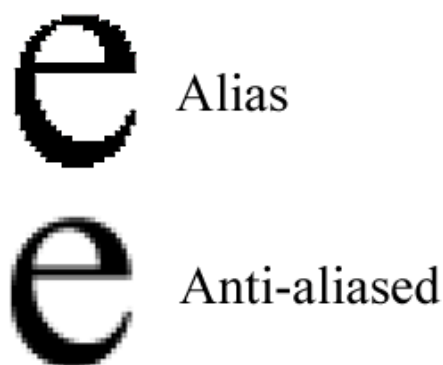


Figura 51: Anti-aliasing
Fonte: Autor

4.4. O processamento da imagem

O processamento da imagem é efetivado quando o algoritmo percorre cada *pixel* dessa imagem transformando-os em outra imagem. No caso do scanner, trabalhando com uma imagem do tipo RGB⁹, o importante é realçar a silhueta vermelha refletida pelo objeto (figura 52). Suponhamos que cada pixel vermelho apresente um determinado valor X. O cálculo realizado para cada pixel o valor deve ser alterado para 255 se apresentar um valor igual ou superior a 45 e o valor deve ser alterado para 0 (zero) se o X encontrado for inferior ao patamar 45 especificado. O valor 45 foi definido a partir dos testes realizados, como sendo o valor que melhor se encaixava na aplicação deste processamento. Desta forma, para cada *pixel*, o canal vermelho foi ressaltado onde houvesse a incidência da cor vermelha de forma intensa e foi colocado um zero onde essa incidência de vermelho fosse muito baixa (figura 53).

⁹ Sistema de cores aditivas Vermelho (Red), Verde (Green) e Azul (Blue).



Figura 52: Imagem original capturada

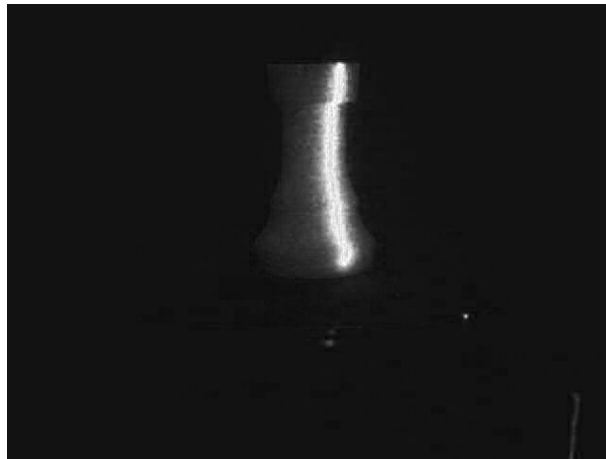


Figura 53: Imagem após o realce da silhueta

Após realizar esse primeiro processamento, ressaltando a linha vermelha da silhueta, o próximo filtro a ser aplicado à imagem é um processamento conhecido como “*Threshold*” (segmentação). De acordo com GUARISE (2007), esse processo refere-se a divisão de uma imagem digital em múltiplas regiões (conjunto de pixels) ou objetos, com o objetivo de simplificar e/ou mudar a representação de uma imagem para facilitar a sua análise, sendo tipicamente utilizada para localizar objetos e formas (linhas, curvas, etc) em imagens (figura 54).



Figura 54: Silhueta destaca após o segundo processamento

4.5. O processamento do eixo central da silhueta

O processamento primário da imagem, onde a silhueta foi destacada através da aplicação de filtros específicos foi essencial para a obtenção da informação de dados espaciais do objeto, pois apenas o destacamento da silhueta na imagem não é suficiente para que se obtenha uma informação com precisão, já que a silhueta destacada é espessa.

Para essa tarefa um algoritmo simples foi idealizado: para cada linha da imagem, foi feita uma busca por pixels de valor 255 (branco). Foi realizada uma média aritmética simples entre a posição do primeiro *pixel* branco e do último. Obteve-se então a posição do pixel que define a silhueta e a linha foi reescrita com apenas 1 *pixel* branco. O resultado foi uma silhueta fina, próxima da silhueta real do objeto (figura 55).

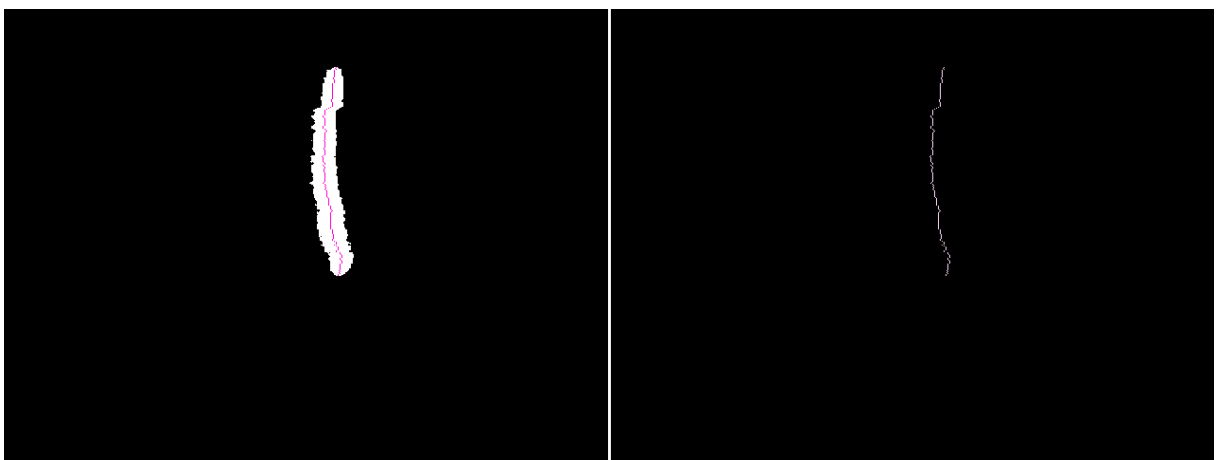


Figura 55: Silhueta fina destacada e resultado final do processamento apresentando apenas a silhueta fina

4.7. A modelagem

Cada conjunto de pixels que compõem o eixo central da silhueta obtida da seqüência de imagens analisadas devem fornecer uma quantidade de pontos “suficientemente densos” para gerar a superfície desejada. Tal quantidade, conhecida como condição de amostragem, deve garantir uma boa aproximação em relação à superfície original, isto é, a superfície reconstruída é homeomorfa e suficiente próxima. (VELHO *et al*, 2005).

Se essa quantidade de pontos é suficiente para representar a superfície, então é necessário aplicar uma solução que seja capaz de interligar corretamente esses pontos para gerar uma malha de polígonos irregulares, os quais definem o modelo da superfície.

A classe *TGLMesh* possui vários métodos para gerar uma malha irregular de superfícies. O método utilizado neste trabalho gera uma malha irregular triangular na tentativa de representar mais fielmente possível a superfície real.

O primeiro conjunto de pontos extraídos do eixo central da primeira silhueta é armazenado em uma estrutura de dados e somente após a obtenção de um segundo conjunto é que o algoritmo faz as ligações entre os pontos e gera a malha correspondente a essa parte da superfície do objeto. Sucessivamente isso é feito, conjunto após conjunto de pontos circunvizinhos até que toda a superfície do objeto esteja coberta.

Após uma seqüência de 200 capturas de imagens (a cada 1,8 graus), o processo de aquisição está concluído e a superfície do objeto está discretizada pelos pontos extraídos das 200 silhuetas correspondentes. O tipo de arquivo escolhido foi o OBJ, pois é um formato de imagem 3D padrão, que pode ser exportado e aberto por vários programas de edição de imagem 3D possuindo a representação textual de um objeto tridimensional através de coordenadas 3D, mapas de textura, e outras informações sobre o objeto e é um formato suportado pelo *GLScene* para o armazenamento de informações.

Desenvolvimento e resultados

Este capítulo apresenta um guia de montagem e utilização do scanner 3D proposto, desde a fabricação da placa de circuito impresso, montagem dos componentes, até a utilização do software controlador do sistema.

5.1. Placa de circuito impresso

A plataforma giratória é um item de hardware importante no projeto, mas não basta apenas conhecer o software embarcado e alguns componentes da placa controladora, sem saber como produzir este artefato.

Utilizando um software de layout e projeto de placas de circuito impresso e o esquema apresentado na figura 44, é simples obter o layout da placa controladora apresentada na figura 45, mas conseguir produzir um circuito impresso, com um bom resultado, sem utilizar um equipamento adequado é uma tarefa mais complicada.

Existem na internet diversos sites que tentam ensinar a produzir um circuito impresso, e após muitas experiências encontrou-se uma alternativa interessante. De posse do desenho adequado do circuito impresso, produzido em um software apropriado, se faz necessário imprimi-lo em um papel *OFF-SET* (folha de revista), utilizando impressora a laser, de forma espelhada, como mostra a figura 56.

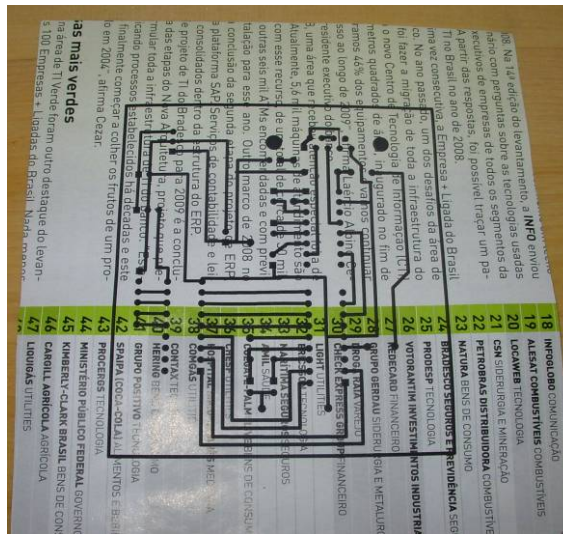


Figura 56: Impressão do circuito em uma folha de revista

Antes de aplicar o desenho do circuito sobre a placa de fenolite¹⁰ ou fibra, é necessário remover as possíveis impurezas. Para isso utiliza-se uma palha de aço e detergente. Após a remoção das impurezas, utiliza-se uma flanela limpa, embebida em álcool, para concluir a limpeza e só então o desenho é aplicado na superfície desta placa.

Utilizando um ferro de passar comum na temperatura mais alta, aplica-se pressão sobre a folha por aproximadamente 5 minutos, alternando a posição do ferro de forma a distribuir o calor (figura 57) e verificando a placa periodicamente para constatar se houve uma boa aderência da folha de revista sobre o cobre, caso contrário é necessário repetir o processo.

¹⁰ O Fenolite é um laminado industrial, duro e denso, feito por aplicação de calor e pressão em camadas de celulose impregnadas com resinas sintéticas (fenólicas). É utilizado para fabricação de placas de circuito impresso assim como a fibra.



Figura 57: Processo de transferência térmica

Após a placa ter esfriado o suficiente para ser manuseada, utiliza-se um pequeno vasilhame, de vidro ou plástico, com certa quantidade de água, suficiente para realizar o processo de esfoliação do papel. Nesse processo ocorre a remoção de parte da folha de revista de forma que sobrem apenas as trilhas impressas na placa (figura 58). Ao mergulhar a placa na água é necessário tomar cuidado ao esfrega-la, para não remover nenhuma trilha acidentalmente.



Figura 58: Processo de esfoliação da placa de circuito

Ao remover todo o papel, utiliza-se uma caneta de retro-projetor para corrigir possíveis falhas no desenho. Essa placa com o desenho corrigido é mergulhada numa solução de 42% de perclorato de ferro, usado para corroer (oxidar) as regiões da lâmina de cobre que não estão protegidas pelas trilhas impressas (figuras 59 e 60).

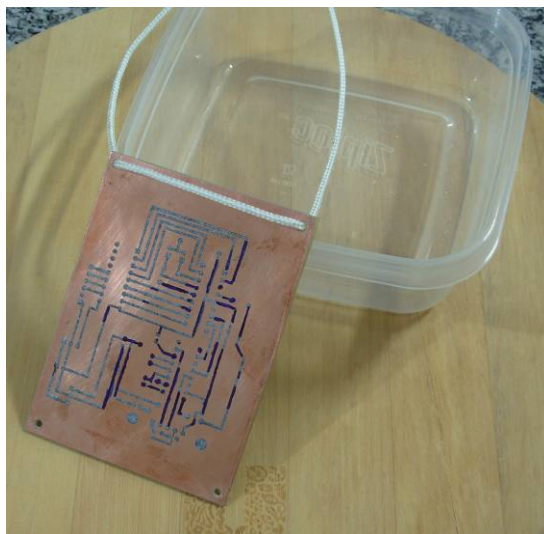


Figura 59: Placa de circuito com correções nos desenhos das trilhas

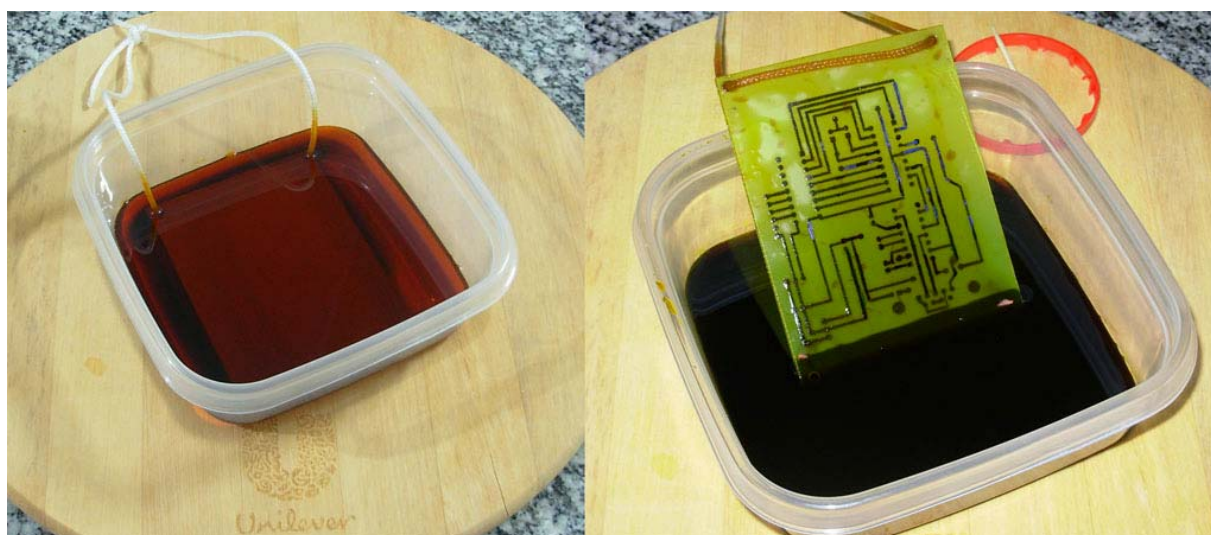


Figura 60: Placa mergulhada em percloro de ferro

Após manufatura, ela está pronta para ser manuseada (realizar furos, encaixar e soldar os componentes) (figura 61).

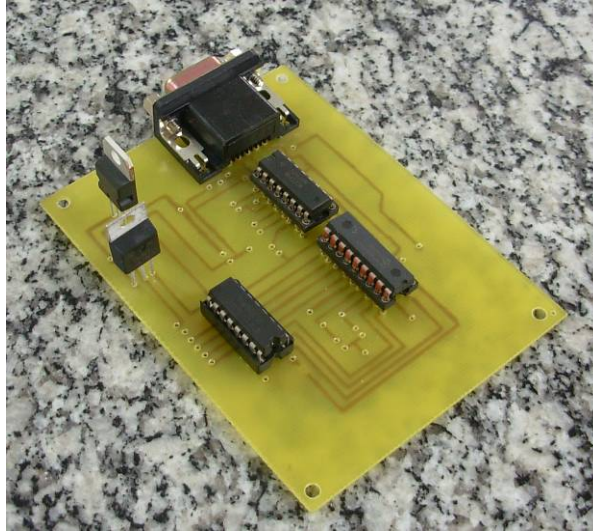


Figura 61: Placa de circuito com alguns componentes soldados

5.2. Plataforma giratória

A plataforma giratória baseia-se em uma caixa de madeira *MDF* (*Médium Density Fiberboard*) com dimensões de 6,5x14x15cm (AxLxP). Sua montagem consiste em colocar o motor preso a tampa, onde um disco de acrílico é fixado posteriormente, e a placa de circuito impresso presa ao fundo da caixa, como mostra a figura 62.

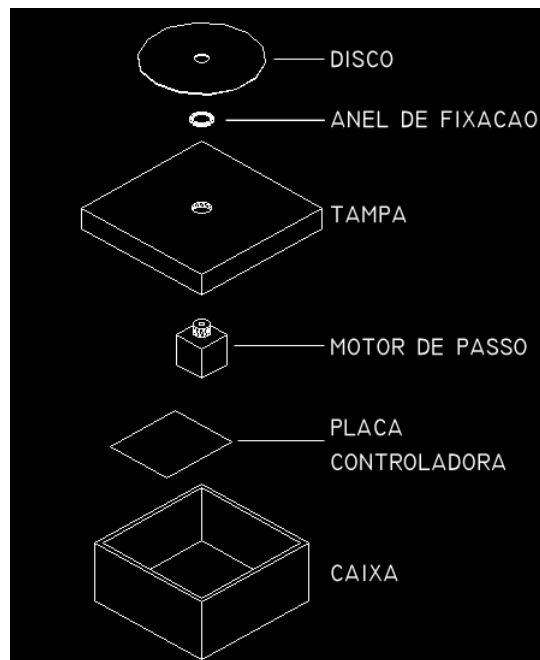


Figura 62: Diagrama de montagem da plataforma giratória

A fixação do motor de passo à tampa é um processo simples, bastando apenas realizar um furo na mesma, com a dimensão do cabeçote preso ao eixo do motor, utilizando uma broca chata. Para finalizar, uma cobertura de papel camurça, de cor preta, é usada para forrar toda a superfície

exterior da caixa. Essa forração minimiza a reflexão do feixe laser que incide sobre ela. Scolado em todo seu contorno, para evitar a reflexão involuntária do laser, se não fosse assim, essa reflexão da caixa poderia ser confundida como sendo parte integrante da silhueta do feixe laser que incide sobre o objeto, fato que resultaria em erro quando essa silhueta for segmentada na imagem adquirida. O processo de furação, fixação do motor e finalização da caixa é ilustrado nas figuras 63 e 64.

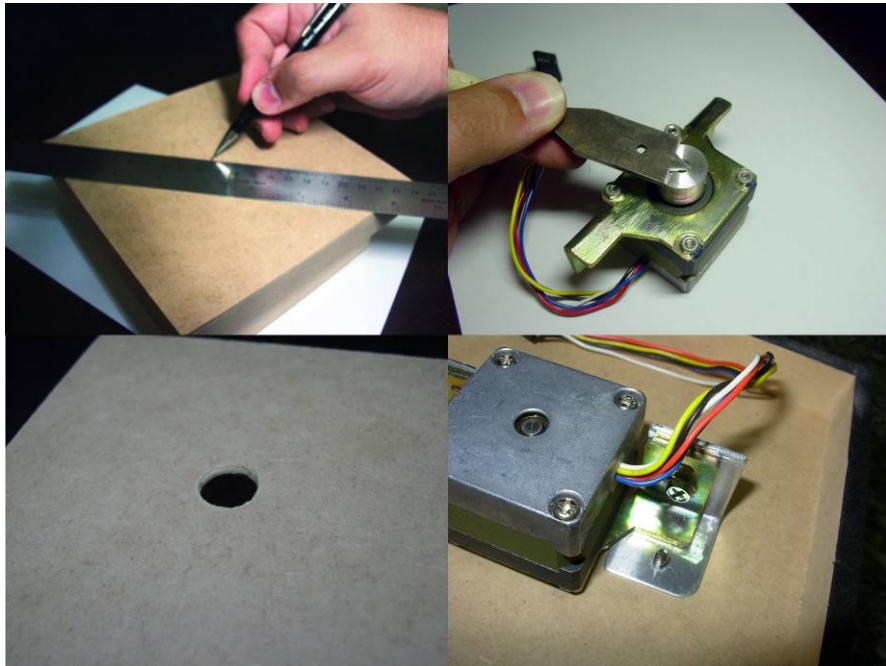


Figura 63: Furação da caixa de MDF e encaixe do motor de passo

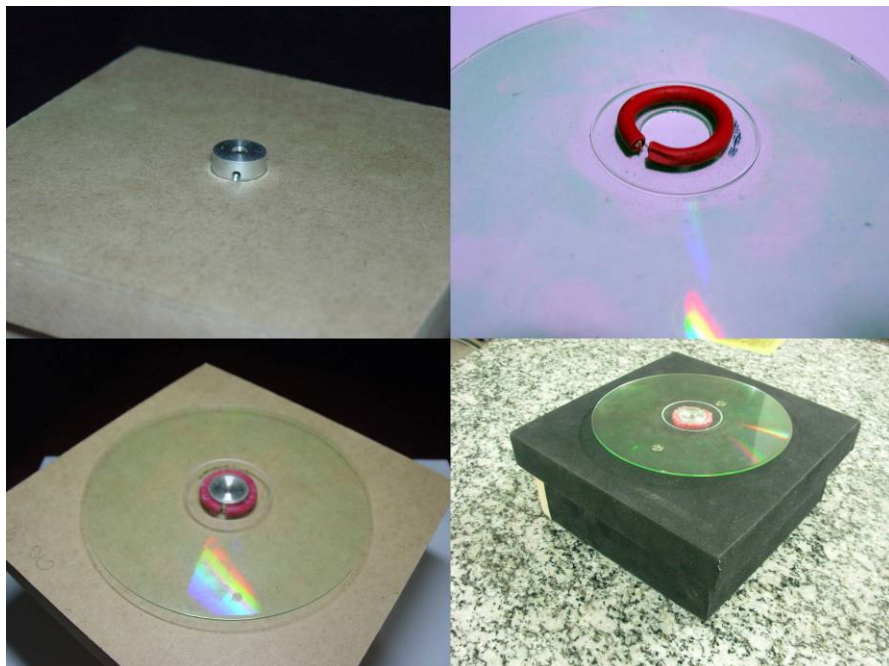


Figura 64: Fixação da plataforma giratória e finalização da plataforma

5.3. Uso do software gerenciador do sistema

O software gerenciador foi projetado para ser simples e intuitivo. Pensando nisso sua interface não dispõe de muitas alternativas de menus ou recursos. A figura 65 ilustra a primeira interface do sistema, na qual é necessário escolher em qual porta serial o scanner está conectado. Caso ocorra algum problema com a porta serial uma mensagem de é retornada, caso contrário, a tela principal do sistema é apresentada ao usuário.

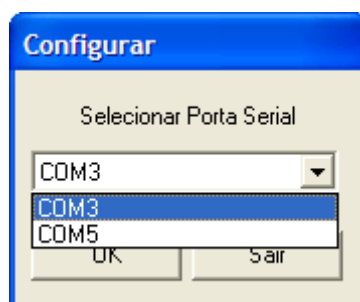


Figura 65: Tela inicial – seleção da porta serial onde o scanner está conectado

Nesse menu, primeiramente o usuário deve selecionar qual dispositivo de captura vai ser utilizado por meio da opção “Cameras”, para logo em seguida deve configurar a resolução em que deseja realizar a captura das imagens, de acordo com as possibilidades do dispositivo de captura, como ilustra as figuras 66 e 67.



Figura 66: Menu de seleção de dispositivo de captura de imagens

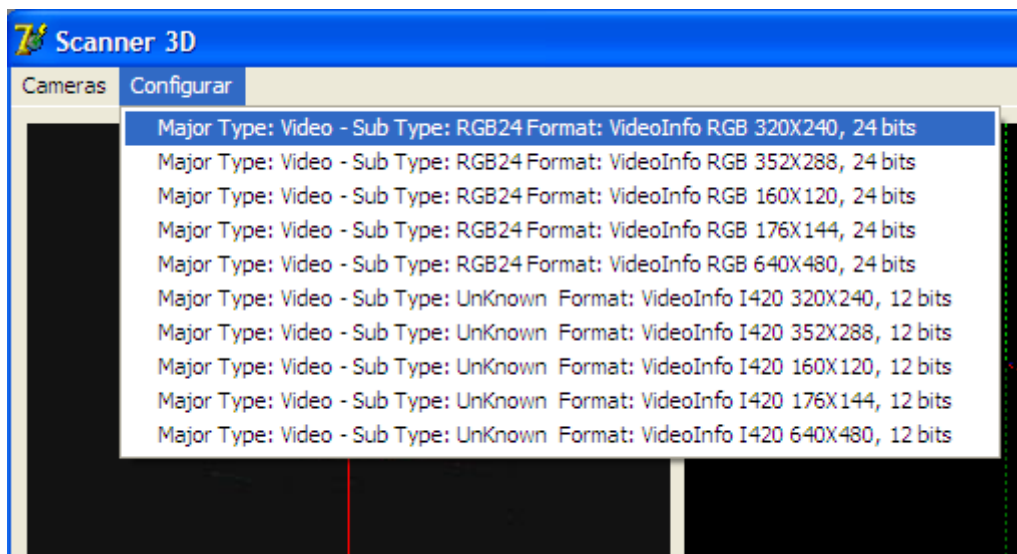


Figura 67: Menu de configuração do dispositivo de captura de imagens

Com o aplicativo configurado e pronto para utilização, é necessário posicionar a câmera de forma que o objeto fique centralizado em relação a linha guia que aparece na tela de captura do lado esquerdo do programa (figura 68).

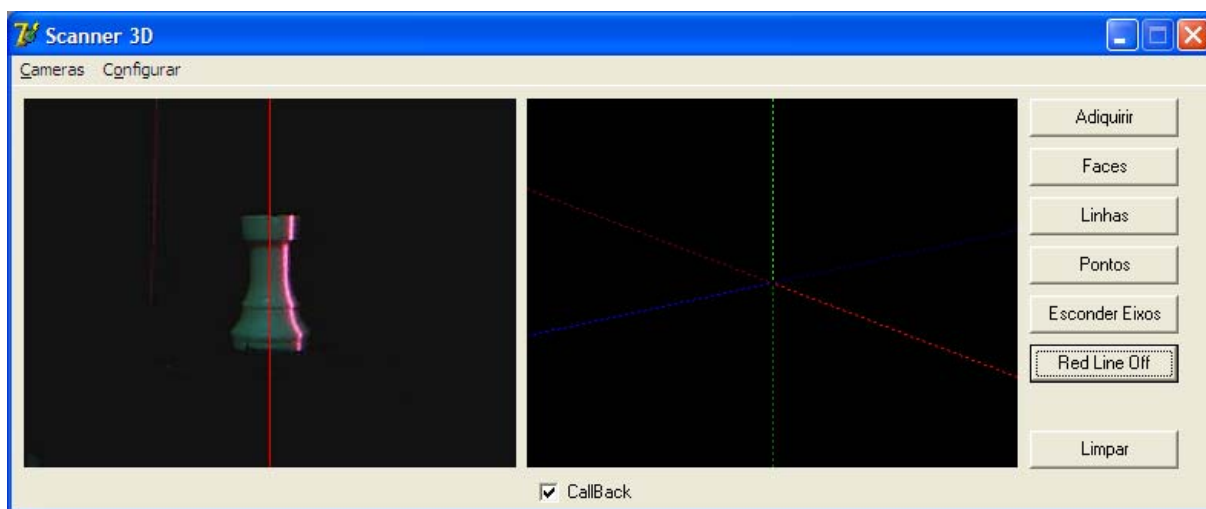


Figura 68: Centralização do objeto em relação a linha guia do sistema

Para iniciar o processo de digitalização basta utilizar o botão “Adquirir”. Neste momento a plataforma começa a girar o objeto, a cada passo do motor a câmera realiza a captura de uma imagem e o software realiza os processamentos referentes a segmentação, refinamento, modelagem e visualização parcial do objeto. Após um giro de 360 graus, dividido em 200 passos discretos do motor utilizado, é possível visualizar o resultado final da digitalização, ilustrado nas figuras 69, 70 e 71.

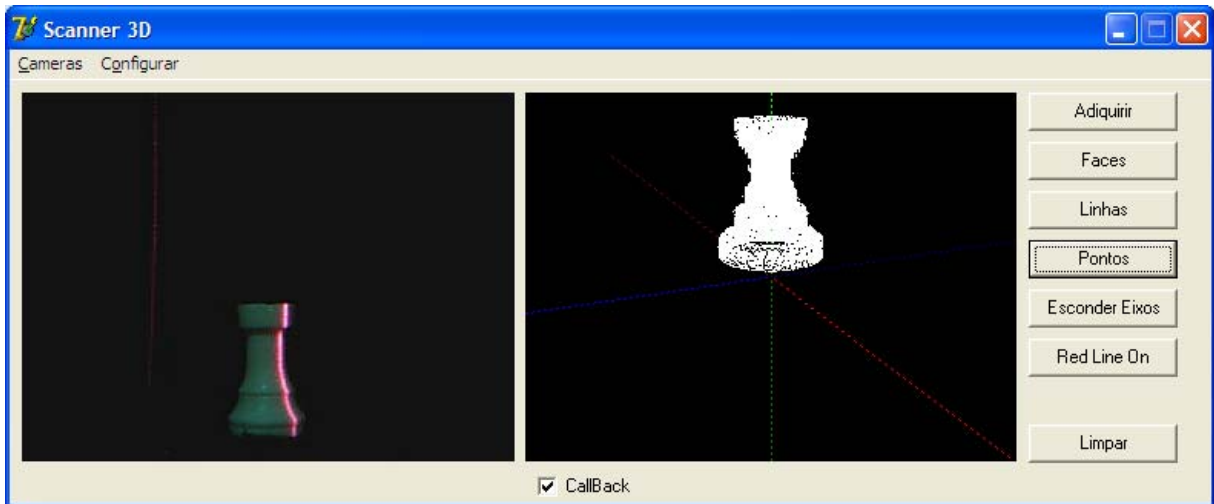


Figura 69: resultado da captura do objeto

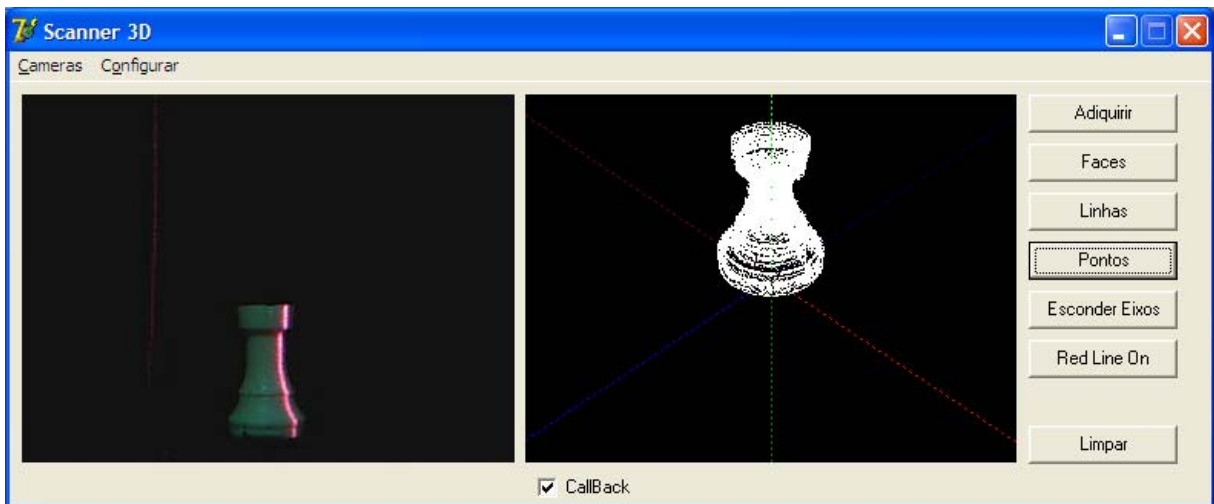


Figura 70: resultado da captura do objeto em outro ângulo

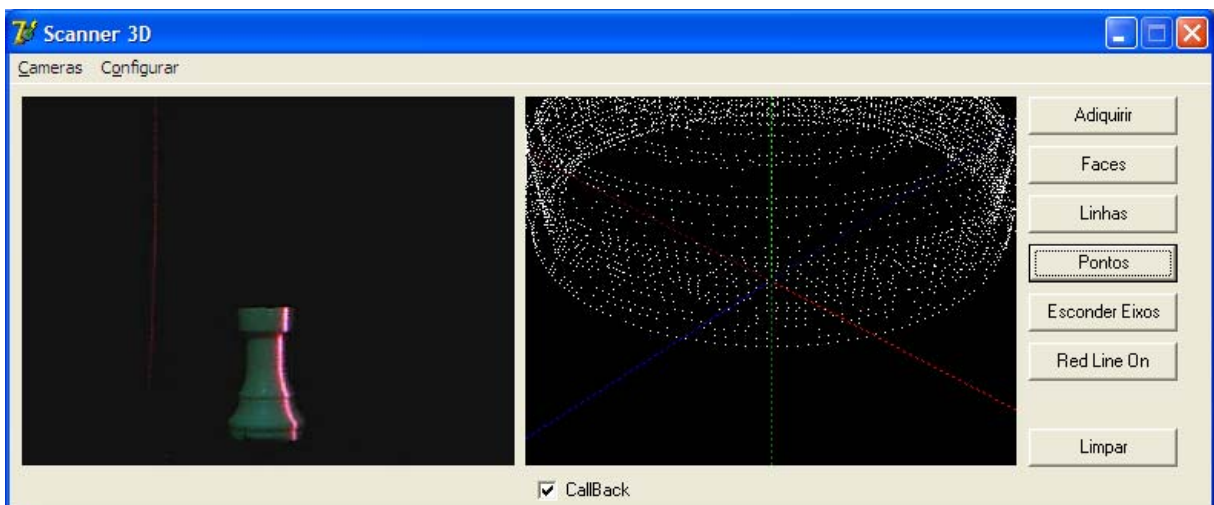


Figura 71: resultado da captura do objeto: nuvem de pontos adquirida pelo sistema

Conclusão

6.1. Limitações da pesquisa

Considerando que o presente trabalho tem como objetivo elaborar uma ferramenta de digitalização 3D simplificada, que viabiliza o uso em situações de ensino de computação gráfica, apresentamos neste trabalho o projeto, a construção e o uso de um dispositivo composto de sistemas físicos e lógicos capazes de realizar a aquisição de dados sobre a superfície do objeto e transforma-los em informações tridimensionais coerentes com o objeto.

Levando em conta o aspecto físico da ferramenta, vale salientar que o tamanho do objeto a ser digitalizado deve obedecer a especificações de tamanho e peso, onde o tamanho máximo deve ser de 20cm, conforme o alcance da *webcam* utilizada; e o peso do objeto não pode exceder 450g, peso máximo suportado pela plataforma giratória atual.

O algoritmo de segmentação das silhuetas do feixe laser projetado sobre a superfície do objeto, utilizado no trabalho, é simples e funcional, apesar de não alcançar resultados exatos ele atende ao objetivo, assim como o algoritmo de cômputo do eixo central da silhueta, que apenas faz um afinamento da silhueta baseado em um ponto médio, sem se preocupar com possíveis deformações ou ruídos.

6.2. Dificuldades encontradas

Durante as pesquisas realizadas, notou-se a existência de diversas ferramentas, tanto para o tratamento de imagens como para modelagem e visualização 3D. Dessa forma a escolha dos componentes utilizados ficou comprometida devido a grande quantidade de ferramentas disponíveis sem estar devidamente documentadas.

Outro problema encontrado foi em relação à compatibilidade dos componentes testados e o sistema operacional *Microsoft Windows Vista*, e por esse motivo optou-se por se realizar o desenvolvimento utilizando o *Microsoft Windows XP*.

O problema mais significativo encontrado foi a assimilação dos trabalhos científicos e a apropriação das informações. Isso se deve ao fato de que os pesquisadores da área de computação gráfica não esclarecem passos importantes no desenvolvimento de seus trabalhos, concentrando-se apenas em apresentar o problema e a solução sugerida, havendo dessa forma pouca informação a respeito do desenvolvimento, ou utilização das ferramentas aplicadas a seus projetos. Apesar da literatura na área de computação gráfica crescer a cada dia, as exposições são complexas e extremamente baseadas em abstrações matemáticas, com modestos registros das relações entre o fenômeno e ocorrências físicas.

6.3. Trabalhos futuros

Como trabalhos futuros, sugerimos a implementação de outras formas de aquisição e processamento dos dados, como por exemplo um método de análise capaz de corrigir as imperfeições causadas pelo reflexo excessivo da luz estruturada sobre a superfície do objeto a ser digitalizado, ou a utilização de métodos clássicos de calibração de câmera.

As sugestões para a realização de trabalhos futuros englobam tanto os pontos que concernem à melhoria da ferramenta, no que diz respeito aos aspectos de hardware e software, quanto sua eficácia como recurso de ensino, para promover estudos sobre a eficácia do uso desse recurso no processo de ensino e aprendizagem da computação gráfica.

6.4. Considerações finais

Na área de ciências humanas, principalmente no curso de formação de professores, é sabido que discussões, pesquisas e reflexões acerca do processo de ensino-aprendizagem, estratégias,

metodologias e recursos de ensino se fazem presentes e que até o momento atual ainda este processo não pode ser tomado como uma realidade precisa e acabada, pois, está sob a égide de teorias, é nessa realidade educacional que esta pesquisa está inserida, ao acreditar que algumas questões educacionais precisam ser discutidas e ampliadas no cotidiano escolar dos cursos pertencentes a área tecnológica.

São recorrentes na literatura específica, sobre computação gráfica, dados e informações que dão pouca ou nenhuma ênfase aos aspectos pedagógicos. O desenvolvimento da ferramenta, objeto de estudo desta pesquisa, foi idealizada para subsidiar a prática pedagógica de aulas teóricas e práticas, para ser utilizada enquanto recurso de como auxílio ao processo de aprendizagem. Esse aspecto assinala a necessidade de discernimento sobre aula teórica entendendo que “tal processo não elimina necessariamente trabalhos concretos, se estes são úteis para o raciocínio”.

Da mesma forma, a aula prática, sendo um momento puramente de “fazer as coisas” ou aplicar o que foi estudado na aula teórica. A aula prática pode vir antes ou depois, desde que ofereça um contato direto com a realidade. Esses aspectos expressam a complexidade do tema abordado e as dificuldades de elaborar ações educativas e seleção de recursos de ensino adequados que venham a intervir e corroborar no processo de apropriação do conhecimento por parte do aprendiz.

Um outro aspecto importante sobre recurso de ensino refere-se a sua aplicabilidade. Este deve ser utilizado e aplicado de forma a viabilizar a capacidade de observação, análise, teorização, síntese, proporcionando a aprendizagem. Pressupõe-se, portanto, não apenas ilustração de um tema, com intuito de facilitar o aprendizado, mas uma ferramenta concreta capaz de auxiliar a mediação em uma área específica.

Referências

ALEMEIDA, L., GAMEIRO, S., MADEIRA, J., TEIXEIRA, J. C., Shoelast 3D — Sistema para Modelação de Formas para Calçado. 9º Encontro Português de Computação Gráfica, Marinha Grande, Portugal, 2000.

AZEVEDO, E.; CONCI, A. *Comutação Gráfica – Teoria e Prática*. Editora Campus: Rio de Janeiro. 2003.

AZEVEDO, T.; TAVARES, J. M. R. S; VAZ, M. A. *Desenvolvimento de uma plataforma computacional para obtenção da forma 3d de objectos usando técnicas de visão activa*. VI Congresso Nacional Mecânica Experimental: FEUP Porto – Portugal. 2005.

BAZAN, W. S.; TOMMASELLI, A. M. G.; GALO, M.; RUY, R. S. *Calibração de um sistema dual de câmeras digitais convergentes*. II Simpósio Brasileiro de Geomática Presidente Prudente. 2007. p. 726-734.

BAZZO, Walter Antonio. *Ciência, tecnologia e sociedade e o contexto da educação tecnológica*. Editora da Univ. Federal de Santa Catarina: Florianópolis, 1998

BORDENAVE, J. D.; PEREIRA, A. M. *Estratégias de Ensino-Aprendizagem*. Petrópolis: Vozes, 1982.

BRAIN, M.. "HowStuffWorks - Como funcionam os motores elétricos". Publicado em 01 de abril de 2000 (atualizado em 14 de abril de 2008) <http://ciencia.hsw.uol.com.br/motor-eletrico1.htm> (21 de novembro de 2009)

CASSAL, M. L. *Geração de sombras emr objetos modelados por geometria sólida construtiva*. Porto Alegre: PPGC da UFRGS, 2001.

CURLESS, B. L. *New methods for surface reconstruction from range images* . Tese (Doutorado em Engenharia Elétrica) Department of Electrical Engineering and the Committee on Graduate Studies, Stanford, 1997.

ENCICLOPÉDIA PRÁTICA JACKSON. V. 4. São Paulo: 1963.

GAGNÉ, R. *Como se realiza a aprendizagem*. Rio de Janeiro: Livros Técnicos e Científicos, 1971.

GUARISE, M. *Processamento de Imagem: Segmentação e K-means*.

Disponível em:

<<http://www.eletrica.ufpr.br/marcelo/TE072/022007/Marcel-Segmentacao.doc>>. Acessado em 16/10/2009

JOSÉ, M. A., *RECONSTRUÇÃO TRIDIMENSIONAL DE BAIXO CUSTO A PARTIR DE PAR DE IMAGENS ESTÉREO*, Dissertação de Mestrado, Escola Politécnica da USP, São Paulo, 2008.

KAWASAKI, H.; FURUKAWA, R.; NAKAMURA, Y. *3D acquisition system using uncalibrated line-laser project Pattern Recognition*, 2006. ICPR 2006. 18th International Conference on Volume: 1 2006 Page(s): 1071-1075

LEITE, B. H. F. C. ; LIMA, E. J. C. ; LEÃO, F. L. ; PRESTES, R. F. *Estudo do Motor de Passo e seu controle digital*.

Disponível em: <http://recreio.gta.ufjf.br/grad/01_1/motor/>. Acessado em 03 de Dez de 2009.

LEOW, K. W.; HUANG, Z.; ZHANG, Y.; SETIONO, R. *Rapid 3D model acquisition from images of small objects*. Geometric Modeling and Processing 2000. Theory and Applications. Proceedings. 2000. Page(s): 33-41

MENEZES JR. L. F. *Controle Automatizado para Scanners de Luz*. Dissertação de Mestrado, Campinas: FEM-UNICAMP. 2004. 84 p.

MIZUKAMI, M. G. N. *Ensino: As abordagens do processo*, São Paulo, Editora Pedagógica Universitária, 1986

MOREIRA, M. A. *Aprendizagem significativa*. Brasília: Editora Universidade de Brasília. 1999.

MORENCY, L. P.; RAHIMI, A.; DARRELL, T. *Fast 3D model acquisition from stereo images*. In: *3D DATA PROCESSING VISUALIZATION AND TRANSMISSION*, 2002. FIRST INTERNATIONAL SYMPOSIUM ON, 1., 2002, Cambridge. Proceedings. Cambridge: 2002. p. 172 - 176.

PALANGANA, I. C. *Desenvolvimento e aprendizagem em Piaget e Vygotsky*. São Paulo: Plexus, 1998.

PEIXOTO, A.; GATTASS, M. *Reconstrução de Superfícies a partir de Seções Bidimensionais*. Qualificação MCC28/00, Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro. 2000.

PERTENCE, A. E. M. ; SANTOS, D. M. C. ; JARDIM, H. V. *desenvolvimento de modelos didáticos através de plataforma cad/cam utilizando o conceito de prototipagem rápida*. In: XVI Congresso Brasileiro de Engenharia Mecânica, 2001, Uberlândia. Anais do COBEM 2001. Uberlândia : UFU, 2001.

PILETTI, C. *Didática Geral*. São Paulo: Editora Ática. 1997.

PINHO, S. M. *Visão Tridimensional*.

Disponível em <<http://www.inf.pucrs.br/~pinho/TACCII/Aula3-Visao3D.pdf>>. Acesso em: 14 set. 2009.

PRATT, W. K. *Digital Image Processing*: PIKS Scientific Inside. Wiley-Interscience: Los Latos – California. 2007

RAPOSOS, A. *Computação Gráfica Interativa*.

Disponível em:

<http://www.tecgraf.puc-rio.br/~abraposo/INF1366/2007/01_Introduc%C3%A3o.pdf>. Acesso em 14 de Set. de 2009.

TSAI, R. Y. *A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses*. IEEE Journal of Robotics and Automation, 3(4):323–344, 1987.

VALENTIN, L. G., CONSULARO, L. A. *DIGITALIZADOR 3D DE BAIXO CUSTO*, XI Encontro Anual de Iniciação Científica, Maringá/PR, 2002

VELHO, L.; CARVALHO, P. C.; SÁ, A.; MEDEIROS, E.; MONTENEGRO, A. A.; PEIXOTO, A.; ESCRIBA, L. A. R. *Fotografia 3D*, 25º Colóquio Brasileiro de Matemática, Rio de Janeiro, 2005.

ZHANG, Z. *Flexible camera calibration by viewing a plane from unknown orientations*. International Conference on Computer Vision (ICCV), pages 666–673, 1999.

Bibliografia consultada

BERNARDINO, A.; SANTOS–VICTOR, J. *Scanner 3D*. Lisboa: Instituto Superior Técnico - Departamento de Engenharia Electrotécnica e de Computadores, 2002. 10 p.

Apêndice A

Código fonte do programa embarcado

```
#include <16F628.h>
#include <stdlib.h>
#include <delay.h>
#include <rs232.h>
#include <fuses.h>
#include <ports.h>

int
    tipo='F'; //inicia com passo completo
byte
    passo=1, //executa 1 passo para inicialização
    motor=9; //inicia o conjunto de bobinas na primeira posição em ambos os
casos

#include <int_rda.h>
int_rda_isr() { //interrupção dizendo que o buffer da porta serial tem dados
    tipo=fgetc(dados); //recebe dados da porta serial
    passo=1; //diz ao sistema para executar 1 passo
}

void main() {
    setup_counters(RTCC_INTERNAL,RTCC_DIV_2);
    setup_timer_1(T1_DISABLED);
    setup_timer_2(T2_DISABLED,0,1);
    setup_comparator(NC_NC_NC_NC);
    setup_vref(FALSE);
    enable_interrupts(INT_RDA);
    enable_interrupts(global);
    set_uart_speed(19200);
    while (true){ //loop infinito
```

```

        if (passo){ //verifica se há passos a serem executados
            switch (tipo){
//verifica o tipo de passo: completo 1 bobina (200 passos) ou 2 bobinas (190)
//passos
                //executa passo completo (full step 190 passos)
            case 'F': {
bobinas
                switch (motor){ //verifica a posição de acionamento das

                    case 3:
                        output_low(PIN_B4);
                        output_high(PIN_B5);
                        output_low(PIN_B6);
                        output_high(PIN_B7);
                        motor=6;
                        break;

                    case 6:
                        output_low(PIN_B4);
                        output_high(PIN_B5);
                        output_high(PIN_B6);
                        output_low(PIN_B7);
                        motor=12;
                        break;

                    case 12:
                        output_high(PIN_B4);
                        output_low(PIN_B5);
                        output_high(PIN_B6);
                        output_low(PIN_B7);
                        motor=9;
                        break;

                    case 9:
                        output_high(PIN_B4);
                        output_low(PIN_B5);
                        output_low(PIN_B6);
                        output_high(PIN_B7);
                        motor=3;
                        break;

                    default: break;
                }
            }
            break; }

//executa passo completo (weak full step 200 passos)
            case 'W': {

```

bobinas

```

switch (motor){ //verifica a posição de acionamento das
    case 3:
        output_high(PIN_B4);
        output_high(PIN_B5);
        output_low(PIN_B6);
        output_high(PIN_B7);
        motor=6;
        break;
    case 6:
        output_low(PIN_B4);
        output_high(PIN_B5);
        output_high(PIN_B6);
        output_high(PIN_B7);
        motor=12;
        break;
    case 12:
        output_high(PIN_B4);
        output_high(PIN_B5);
        output_high(PIN_B6);
        output_low(PIN_B7);
        motor=9;
        break;
    case 9:
        output_high(PIN_B4);
        output_low(PIN_B5);
        output_high(PIN_B6);
        output_high(PIN_B7);
        motor=3;
        break;
    default: break;
} break; }
case 'C': output_high(PIN_B4);
        output_high(PIN_B5);
        output_high(PIN_B6);
        output_high(PIN_B7);
        break;

    default: break;
}
putc(tipo);
passo=0; //zera o contador de passos

```

```
        tipo='X'; //muda o tipo de passo para um tipo inválido
    }else{
        #asm
            nop
        #endasm
    }
}
}
```

Anexo A

Linha temporal da área de computação gráfica

Azevedo e Conci(2003) descrevem uma linha temporal para o melhor entendimento dos avanços e os importantes fundamentos na área de computação gráfica, a saber:

- Euclides [300-250a.C.] – desenvolveu toda a geometria que norteou seu desenvolvimento até o século XVIII.
- Brunelleschi [1377-1446] – arquiteto e escultor italiano que usou de forma criativa a noção de percepção visual, e criou em 1425 a perspectiva.
- Descartes [1596-1650] – matemático e filósofo francês que formulou a geometria analítica e os sistemas de coordenadas 2D e 3D.
- Euler [1707-1783] – o mais produtivo matemático do século XVIII, que, entre outros, criou o conceito de senos, tangentes, a expressão que relaciona o número de vértices, arestas e faces de poliedros etc.
- Monge [1746-1818] – matemático francês que desenvolveu a geometria descritiva como um ramo da geometria.
- Sylvester [1814-1897] – matemático inglês que inventou as matrizes e a notação matricial, uma das ferramentas mais comuns da computação gráfica.
- Hermite [1822-1901] – matemático francês que provou a transcendência do número e (usado como base para os logaritmos naturais) desenvolveu funções elípticas e curvas, entre outros.
- Em 1959, surge o termo Computer Graphics, criado por L. Hudson da Boeing;
- No final da década de 1950, as universidades e empresas americanas, como a Boeing, começam a usar computadores para testar idéias e novas aplicações;
- Em 1961, no MIT é criado o primeiro jogo de computador (Spacewars) para o computador DEC PDP-1;

- Em 1963, Sutherland apresenta um sistema de desenho interativo de primitivas gráficas 2D baseado em caneta luminosa;
- Em 1963, Zajac produz nos laboratórios da Bell o primeiro filme gerado por computador (imagens formadas de linhas e texto);
- Em 1963, surge o primeiro sistema comercial de CAD (DAC-1);
- Em 1963, Coons inventa a teoria de representação de superfícies curvas através de “retalhos” baseados em aproximações polinomiais;
- Em 1965, Roberts cria um algoritmo de remoção de partes invisíveis de segmentos de reta e introduz a noção de coordenadas homogêneas na representação geométrica de objetos;
- Em 1966, é lançado no mercado o primeiro console caseiro de jogos Odyssey;
- Em 1966, surge a primeira empresa de produção computacional de animações e efeitos especiais, a MAGI;
- Em 1967, Rougelet cria o primeiro simulador de vôo interativo da NASA;
- Em 1969, a MAGI produz para a IBM o primeiro comercial baseado em técnicas de computação gráfica;
- Em 1969, é criado entre os grupos da ACM o Special Interest Group on Graphics SIGGRAPH;
- Em 1969, nos laboratórios da Bell, é construída a primeira matriz de pixels (cada pixel representado por 3 bits);
- Em 1972, A. Kay, no Xerox PARC, produz o computador gráfico Alto;
- Em 1972, Bushnell funda a empresa ATARI;
- Em 1973, Metcalf desenvolve a tecnologia Ethernet e é editado o primeiro livro que aborda detalhadamente os algoritmos e métodos da computação gráfica (autores Newman e Sproull);
- Em 1974, Catmull desenvolve o algoritmo Z-Buffer;
- Em 1977, a Academia de Artes e Ciências Cinematográficas de Hollywood cria a categoria de Oscar de Efeitos Especiais;
- Em 1979, G. Lucas contrata Catmull, Ray Smith e outros para uma nova empresa denominada Lucas Film;

Anexo B

Tabela de caracteres ASCII completa

Tabela 4: Tabela de caracteres ASCII completa

Decimal	Binário	Hex	Referência
0	00000000	00	Null - NUL
1	00000001	01	Start of Heading - SOH
2	00000010	02	Start of Text - STX
3	00000011	03	End of Text - ETX
4	00000100	04	End of Transmission - EOT
5	00000101	05	Enquiry - ENQ
6	00000110	06	Acknowledge - ACK
7	00000111	07	Bell, rings terminal bell - BEL
8	00001000	08	BackSpace - BS
9	00001001	09	Horizontal Tab - HT
10	00001010	0A	Line Feed - LF
11	00001011	0B	Vertical Tab - VT
12	00001100	0C	Form Feed - FF
13	00001101	0D	Enter - CR
14	00001110	0E	Shift-Out - SO
15	00001111	0F	Shift-In - SI
16	00010000	10	Data Link Escape - DLE
17	00010001	11	Device Control 1 - D1
18	00010010	12	Device Control 2 - D2
19	00010011	13	Device Control 3 - D3
20	00010100	14	Device Control 4 - D4
21	00010101	15	Negative Acknowledge - NAK
22	00010110	16	Synchronous idle - SYN
23	00010111	17	End Transmission Block - ETB
24	00011000	18	Cancel line - CAN
25	00011001	19	End of Medium - EM
26	00011010	1A	Substitute - SUB
27	00011011	1B	Escape - ESC
28	00011100	1C	File Separator - FS

29	00011101	1D	Group Separator - GS
30	00011110	1E	Record Separator - RS
31	00011111	1F	Unit Separator - US
32	00100000	20	Space - SPC
33	00100001	21	!
34	00100010	22	"
35	00100011	23	#
36	00100100	24	\$
37	00100101	25	%
38	00100110	26	&
39	00100111	27	'
40	00101000	28	(
41	00101001	29)
42	00101010	2A	*
43	00101011	2B	+
44	00101100	2C	,
45	00101101	2D	-
46	00101110	2E	.
47	00101111	2F	/
48	00110000	30	0
49	00110001	31	1
50	00110010	32	2
51	00110011	33	3
52	00110100	34	4
53	00110101	35	5
54	00110110	36	6
55	00110111	37	7
56	00111000	38	8
57	00111001	39	9
58	00111010	3A	:
59	00111011	3B	;
60	00111100	3C	<
61	00111101	3D	=
62	00111110	3E	>
63	00111111	3F	?
64	01000000	40	@
65	01000001	41	A
66	01000010	42	B
67	01000011	43	C
68	01000100	44	D
69	01000101	45	E
70	01000110	46	F
71	01000111	47	G
72	01001000	48	H
73	01001001	49	I
74	01001010	4A	J
75	01001011	4B	K

76	01001100	4C	L
77	01001101	4D	M
78	01001110	4E	N
79	01001111	4F	O
80	01010000	50	P
81	01010001	51	Q
82	01010010	52	R
83	01010011	53	S
84	01010100	54	T
85	01010101	55	U
86	01010110	56	V
87	01010111	57	W
88	01011000	58	X
89	01011001	59	Y
90	01011010	5A	Z
91	01011011	5B	[
92	01011100	5C	\
93	01011101	5D]
94	01011110	5E	^
95	01011111	5F	_
96	01100000	60	`
97	01100001	61	a
98	01100010	62	b
99	01100011	63	c
100	01100100	64	d
101	01100101	65	e
102	01100110	66	f
103	01100111	67	g
104	01101000	68	h
105	01101001	69	i
106	01101010	6A	j
107	01101011	6B	k
108	01101100	6C	l
109	01101101	6D	m
110	01101110	6E	n
111	01101111	6F	o
112	01110000	70	p
113	01110001	71	q
114	01110010	72	r
115	01110011	73	s
116	01110100	74	t
117	01110101	75	u
118	01110110	76	v
119	01110111	77	w
120	01111000	78	x
121	01111001	79	y
122	01111010	7A	z

123	01111011	7B	{
124	01111100	7C	
125	01111101	7D	}
126	01111110	7E	~
127	01111111	7F	Delete
128	10000000	80	Ç
129	10000001	81	ü
130	10000010	82	é
131	10000011	83	â
132	10000100	84	ä
133	10000101	85	à
134	10000110	86	å
135	10000111	87	ç
136	10001000	88	ê
137	10001001	89	è
138	10001010	8A	è
139	10001011	8B	ì
140	10001100	8C	î
141	10001101	8D	ì
142	10001110	8E	Ä
143	10001111	8F	Å
144	10010000	90	É
145	10010001	91	æ
146	10010010	92	Æ
147	10010011	93	ô
148	10010100	94	ö
149	10010101	95	ò
150	10010110	96	û
151	10010111	97	ù
152	10011000	98	ÿ
153	10011001	99	Ö
154	10011010	9A	Ü
155	10011011	9B	ø
156	10011100	9C	£
157	10011101	9D	∅
158	10011110	9E	×
159	10011111	9F	f
160	10100000	A0	á
161	10100001	A1	ù
162	10100010	A2	ó
163	10100011	A3	ú
164	10100100	A4	ñ
165	10100101	A5	Ñ
166	10100110	A6	ª
167	10100111	A7	º
168	10101000	A8	¿
169	10101001	A9	®

170	10101010	AA	¬
171	10101011	AB	½
172	10101100	AC	¼
173	10101101	AD	i
174	10101110	AE	«
175	10101111	AF	»
176	10110000	B0	⋯
177	10110001	B1	⋮
178	10110010	B2	■
179	10110011	B3	
180	10110100	B4	└
181	10110101	B5	Á
182	10110110	B6	Â
183	10110111	B7	À
184	10111000	B8	©
185	10111001	B9	⋈
186	10111010	BA	
187	10111011	BB	┘
188	10111100	BC	┘
189	10111101	BD	ç
190	10111110	BE	¥
191	10111111	BF	┘
192	11000000	C0	L
193	11000001	C1	⊥
194	11000010	C2	T
195	11000011	C3	└
196	11000100	C4	—
197	11000101	C5	+
198	11000110	C6	ã
199	11000111	C7	Ã
200	11001000	C8	ℓ
201	11001001	C9	ℓ
202	11001010	CA	ℓ
203	11001011	CB	ℓ
204	11001100	CC	ℓ
205	11001101	CD	=
206	11001110	CE	ℓ
207	11001111	CF	α
208	11010000	D0	ö
209	11010001	D1	Ð
210	11010010	D2	Ê
211	11010011	D3	È
212	11010100	D4	È
213	11010101	D5	ı
214	11010110	D6	Í
215	11010111	D7	Î
216	11011000	D8	Ï

217	11011001	D9	Ј
218	11011010	DA	Г
219	11011011	DB	■
220	11011100	DC	■
221	11011101	DD	∴
222	11011110	DE	İ
223	11011111	DF	■
224	11100000	E0	Ó
225	11100001	E1	ß
226	11100010	E2	Ô
227	11100011	E3	Ò
228	11100100	E4	ø
229	11100101	E5	Õ
230	11100110	E6	μ
231	11100111	E7	þ
232	11101000	E8	Ɔ
233	11101001	E9	Ú
234	11101010	EA	Û
235	11101011	EB	Û
236	11101100	EC	Ý
237	11101101	ED	Ÿ
238	11101110	EE	—
239	11101111	EF	˘
240	11110000	F0	
241	11110001	F1	±
242	11110010	F2	≡
243	11110011	F3	$\frac{3}{4}$
244	11110100	F4	¶
245	11110101	F5	§
246	11110110	F6	÷
247	11110111	F7	˙
248	11111000	F8	◦
249	11111001	F9	∴
250	11111010	FA	•
251	11111011	FB	1
252	11111100	FC	3
253	11111101	FD	2
254	11111110	FE	■
255	11111111	FF	