

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

DEPARTAMENTO ACADÊMICO DE ELETRÔNICA

ANDRÉ DE CASTILHO COSTA PINTO

MARCOS VINÍCIUS ZIMMERMANN OTANI

MARIANA MOHR LOLIS

**ANALISADOR DE FALHAS EM PLACAS DE CIRCUITO
IMPRESSO**

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA

2012

ANDRÉ DE CASTILHO COSTA PINTO

MARCOS VINÍCIUS ZIMMERMANN OTANI

MARIANA MOHR LOLIS

**ANALISADOR DE FALHAS EM PLACAS DE CIRCUITO
IMPRESSO**

Trabalho de Conclusão de Curso como requisito parcial à obtenção do título de Bacharel do Departamento Acadêmico de Eletrônica, da Diretoria de Graduação e Educação Profissional, da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Gustavo Borba

CURITIBA

2012

Folha destinada à inclusão da Ficha Catalográfica (elemento obrigatório somente para as dissertações) a ser solicitada ao Departamento de Biblioteca do Campus UTFPR (prazo: 3 dias) e posteriormente impressa no verso da Folha de Rosto (folha anterior).

AGRADECIMENTOS

Ao nosso orientador, Prof. Dr. Gustavo Borba, pelo entusiasmo, bom humor e apoio técnico, que foram sempre muito além do estritamente necessário, e sem os quais não conseguiríamos ter atingido nossos objetivos no curto período de tempo que possuíamos.

Ao Prof. Dr. Fernando Castaldo, por fornecer material, tempo e paciência para que fosse possível testar o algoritmo em placas de verdade.

Às nossas namoradas e namorado, sem os quais este trabalho, definitivamente, não poderia ter sido desenvolvido.

Às nossas famílias, pela paciência, apoio e confiança.

Aos colegas engenheiros Guilherme Kira e Henrique Gubert, pelos valiosos conselhos de quem já havia passado pelo que passamos nestes últimos meses.

Ao Chico.

“Pois vista todos têm;
visão é que são elas.”

Millôr Fernandes

RESUMO

COSTA PINTO, André C.; LOLIS, Mariana M.; OTANI, Marcos V. Z.. Analisador de Placas de Circuito Impresso. 2012. 112 fls. Trabalho de Conclusão de Curso (Engenharia Industrial Elétrica – Ênfase em Eletrônica e Telecomunicações) - Universidade Tecnológica Federal do Paraná. Curitiba, 2012.

Inspecionar visualmente a produção de placas de circuito impresso, além de ser uma tarefa extenuante, não é um processo confiável. O nível de miniaturização alcançado pela indústria chega, muitas vezes, a inclusive inviabilizar este tipo de inspeção. Este trabalho apresenta uma solução completa de inspeção de placas de circuito impresso, baseada em visão computacional, denominada Abutre, capaz de reconhecer curto entre trilhas, trilhas abertas, e se há trilhas ou espaçamentos mais finos que o previsto no projeto da placa. Analisam-se, também, os aspectos de gestão do projeto desenvolvido e é apresentado um plano de negócios para uma potencial empresa a ser fundada no ramo de inspeção industrial, que teria como primeiro produto o Abutre.

Palavras-chave: **Inspeção Óptica Automatizada. Placa de Circuito Impresso.**

ABSTRACT

Visually inspecting the production of printed circuit boards, besides being an extenuating task, is not a reliable process. The miniaturization level achieved by this industry can even make such inspection approach unfeasible. This work presents a complete solution for PCB inspection, based on computer vision, called *Abutre* (Portuguese for Vulture), that is capable of recognizing short circuited and open tracks; and telling if the tracks' width and the spacing between them are within specifications. The management aspect of the development of this project and a business plan of a potential company in the field of industrial inspection are also present in this work.

Keywords: **Automated Optical Inspection; Printed Circuit Board.**

LISTA DE ILUSTRAÇÕES

Figura 1 – Árvore taxonômica de Moganti et al. (1996)	19
Figura 2 - Figura 2 - (a) Imagem da PCI segmentada; (b) Imagem após operação de 'fill'; (c) Imagem dos furos após a operação (a) – (b); (d) centroides dos furos.	24
Figura 3 - (a) furos da placa de referência enumerados; (b) regiões de Voronoi da placa de referência; (c) Trilhas da placa de referência enumeradas.....	26
Figura 4 - (a) placa de teste binarizada; (b) centroide dos furos da placa testada (c) trilhas enumeradas da placa testada. Devido aos defeitos do circuito aberto, a enumeração agora é diferente daquela obtida na placa de referência (Figura 3(a)).	26
Figura 5 - Método de inspeção de PCB baseada em conectividade.	27
Figura 6 - Diagrama em blocos do sistema desenvolvido.....	28
Figura 7 - Placa utilizada nos primeiros testes de segmentação.	29
Figura 8 - Resultado da segmentação utilizando o algoritmo de Otsu para definir o nível de threshold	30
Figura 9- Resultado da segmentação com adapthisteq.....	31
Figura 10 - Resultados k-means com 4 regiões e sem valores iniciais (a) RGB (b) Log (c) Gray	32
Figura 11 - Resultados k-means com 3 regiões e sem valores iniciais (a) RGB (b) Log (c) Gray	32
Figura 12 - Resultados k-means com 3 regiões e com valor inicial (a) RGB (b) Log (c) Gray	32
Figura 13 - Imagem da placa original, utilizada como referência.....	34
Figura 14 - Imagem com falhas inseridas	35
Figura 15 - Placa de referência.....	35
Figura 16 - Placa com defeitos inseridos manualmente	36
Figura 17 - Resultado do algoritmo de Tatibana et al.	36
Figura 18 - Exemplo do algoritmo de intersecção.....	38
Figura 19 – Aplicação do filtro de gradiente.....	40
Figura 20 – Aplicação do filtro de Sobel	40
Figura 21 - Aplicação do operador Laplaciano 5x5.....	41
Figura 22- Uso da HEAP ao longo do programa	45

Figura 23 - Sistema de aquisição com SoftBox	51
Figura 24 - Mesa estativa para fotografia comercializada pela empresa Tecnocroma.	52
Figura 25 - Desenho do sistema de aquisição de imagens	54
Figura 26 - Sistema de Aquisição de Imagens	55
Figura 27 - Canon EOS T2i	56
Figura 28 - Duas fotografias tiradas com o sistema criado, da mesma placa, só que em (a) com a lente EF 50mm f/1.8 e em (b) com a lente macro profissional MP-E 65mm f/2.8	56
Figura 29 - Resultados da análise do sistema de aquisição de imagens	58
Figura 30 - Diagrama em Blocos do Software Embarcado	59
Figura 31 - Exemplo de implementação de um estado com o <i>Qt State Machine Framework</i>	62
Figura 32 - Estrutura do banco de dados.....	63
Figura 33 - Screenshot do aplicativo desenvolvido.....	65
Figura 34 - Beagleboard xM	67
Figura 35 - Screenshot do <i>site</i> Narcissus	68
Figura 36- Logomarca da empresa Valar.	76
Figura 37 - Participação das exportações no faturamento das Indústrias de Eletroeletrônicos.....	82
Figura 38 - Expectativas e exportações da indústria de eletroeletrônicos	82
Figura 39 - Uso de telefones celulares de 2007 a 2010.....	83
Figura 40 - Divisão das empresas produtoras de PCI de acordo com o tamanho.	86
Figura 41 - Proposta de valor do Abutre.....	88

LISTA DE TABELAS

Tabela 1 - Erros percentuais de cada variação do algoritmo de k-means	33
Tabela 2 - Relação de cada trilha na placa de teste a uma trilha na placa de referência	38
Tabela 3- Relação de cada trilha na placa de referência a uma trilha na placa de teste	38
Tabela 4- Requisitos para a estrutura física do sistema de aquisição de imagens	50
Tabela 5 - Comparação das soluções com os requisitos de projeto.....	53
Tabela 6 - Tabela comparativa de sistemas de processamento.....	66
Tabela 7 - Custos do projeto.....	73
Tabela 8 - Projeção de indicadores do setor eletroeletrônico para 2012.....	84
Tabela 9 - Faturamento dos setores da Indústria de Eletroeletrônicos.....	85
Tabela 10 - Produção de PCIs no Brasil, subdividida por número de camadas. .	87
Tabela 11 - Quadro de pessoal	92
Tabela 12 - Projeção de índices e taxas.....	92
Tabela 13 - Encargos sociais.....	93
Tabela 14 - Orçamento de Produção e de Matérias-Primas.....	93
Tabela 15 - Orçamento de mão-de-obra direta.....	94
Tabela 16 - Orçamento de despesas indiretas de produção	95
Tabela 17 - Orçamento de despesas administrativas	95
Tabela 18 - Orçamento de investimentos na produção	96
Tabela 19 - Orçamento de investimentos na administração	96
Tabela 20 - Movimentações do imobilizado na produção	97
Tabela 21 - Movimentações no imobilizado da administração.....	97
Tabela 22 - Depreciação acumulada - Produção.....	98
Tabela 23 - Depreciação acumulada - administração.....	99
Tabela 24 - Orçamento de financiamentos	99
Tabela 25 - Orçamento de Aplicações Financeiras	100
Tabela 26 - Movimentação de contas a receber.....	101
Tabela 27 - Movimentação de fornecedores.....	101

Tabela 28 - Movimentação de impostos a recolher	102
Tabela 29 - Movimentação de folha de pagamento e encargos sociais	103
Tabela 30 - Movimentação de contas a pagar	103
Tabela 31 - Fluxo de caixa ao longo dos três primeiros anos.....	104
Tabela 32 - Demonstração do resultado em exercício	105
Tabela 33 - Calculo do valor presente liquido dos superavits (déficits) futuros .	106
Tabela 34 - Estimativo da TIR	106

LISTA DE ABREVIATURAS

ABRACI: Associação Brasileira de Circuitos Impressos

ABINEE: Associação Brasileira da Indústria Elétrica e Eletrônica

AOI: *Automatic Optical Inspection*

ARM: *Advanced RISC Machines*

GNU: *GNU is Not Unix*

MVC: *Model-View-Controller*

PCB: *Printed Circuit Board*

PCI: Placa de Circuito Impresso

RAM: *Random Access Memory*

SIFT: *Scale Invariant Fast Transform*

SMD: *Surface Mount Devices*

SoC: *System on Chip*

SURF: *Speeded Up Robust Feature*

XML: *Extensible Markup Language*

SUMÁRIO

1. INTRODUÇÃO	16
1.1. OBJETIVO E JUSTIFICATIVA	16
1.2. METODOLOGIA	17
2. FUNDAMENTAÇÃO TEÓRICA	18
2.1. TRABALHOS REALIZADOS POR MOGANTI, ET AL. (1996)	18
2.1.1. Imagens por Raios-X e outras tecnologias	19
2.1.2. Inspeção Ótica/Visual Automática.....	20
2.2. TRABALHOS REALIZADOS POR (TATIBANA, ET AL., 1997)	23
3. DESENVOLVIMENTO.....	28
3.1. VISÃO GERAL	28
3.2. ALGORITMO PARA DETECÇÃO DE FALHAS EM PCIs	28
3.2.1. Estudos em MATLAB.....	28
3.2.2. Estudos em Outros Softwares.....	39
3.2.3. Implementação com o OpenCV	41
3.2.4. Análise do Algoritmo	44
3.3. SISTEMA DE AQUISIÇÃO DE IMAGENS.....	46
3.3.1. Estudo preliminar para o projeto da estrutura física.....	46
3.3.2. Concepção do Sistema de Aquisição.....	50
3.3.3. Construção do Sistema de Aquisição de Imagens.....	53
3.3.4. Escolha da Câmera Fotográfica.....	55
3.3.5. Análise do desempenho do sistema	56
3.4. SOFTWARE EMBARCADO.....	59
3.4.1. A Biblioteca Qt	60
3.4.2. Núcleo do Sistema (State Machine).....	61
3.4.3. Modelos (interação com Banco de Dados)	62
3.4.4. Views (Interface com o Usuário)	63
3.4.5. Comunicação com a Câmera.....	64
3.4.6. GUI.....	64
3.5. PLATAFORMA EMBARCADA	65
3.5.1. Estudos Iniciais da Plataforma.....	65

3.5.2. Implementação da Toolchain	68
4. GESTÃO	70
4.1. ESCOPO.....	70
4.2. CRONOGRAMA	70
4.3. CUSTOS.....	72
4.4. RISCOS.....	73
4.5. CONSIDERAÇÕES	74
5. PLANO DE NEGÓCIOS.....	75
5.1. SUMÁRIO EXECUTIVO.....	75
5.2. DEFINIÇÃO DO NEGÓCIO.....	76
5.3. OBJETIVOS.....	77
5.4. PRODUTOS E SERVIÇOS	78
5.4.1. O Abutre.....	78
5.4.2. Análise Comparativa	79
5.4.3. Tecnologias.....	80
5.4.4. Produtos e Serviços Futuros.....	80
5.5. ANÁLISE DO MERCADO RESUMIDA.....	82
5.5.1. Resumo do Macro Ambiente.....	82
5.5.2. Segmentação do Mercado	84
5.5.3. Segmento Alvo – Indústria Produtora de Placas de Circuito Impresso	85
5.5.4. Definição da Proposta de Valor.....	86
5.6. ESTRATÉGIA	88
5.6.1. Diferenciais Competitivos.....	89
5.6.2. Vendas.....	89
5.6.3. Preço.....	89
5.6.4. Promoção.....	90
5.6.5. Distribuição	90
5.6.6. Futuro.....	90
5.7. GESTÃO DA EMPRESA	91
5.7.1. Estrutura Organizacional.....	91
5.7.2. Quadro de Pessoal	92

5.8. PLANO FINANCEIRO	92
5.8.1. Projeção de Índices e Taxas	92
5.8.2. Orçamento de Produção e de Matérias-Primas	93
5.8.3. Orçamento de Mão-de-Obra Direta.....	94
5.8.4. Orçamento de Custos Indiretos de Fabricação	94
5.8.5. Orçamento de Despesas Administrativas	95
5.8.6. Orçamento de Investimentos	95
5.8.7. Depreciação	98
5.8.8. Orçamento de Aplicações Financeiras e de Financiamento	99
5.8.9. Projeção de Resultados	100
5.8.10. Projeção do fluxo de caixa	103
5.8.11. Demonstração do Resultado em Exercício (DRE)	104
5.8.12. Payback	105
5.8.13. Análise de VPL (Valor Presente Líquido)	105
5.8.14. TIR (Taxa Interna de Retorno)	106
7. CONCLUSÃO	107
7.1. ASPECTOS TÉCNICOS	107
7.2. TRABALHOS FUTUROS	108

1. INTRODUÇÃO

Placas de circuito impresso são utilizadas para fixar e conectar eletricamente componentes eletrônicos. A placa é feita de um material isolante revestido por uma ou mais camadas de cobre, que são corroídas para formar as trilhas condutoras. Como essas trilhas são da ordem de grandeza de alguns milímetros, qualquer falha nesse processo de corrosão pode resultar na má conexão dos componentes eletrônicos. Erros em placas de circuito impresso, se não detectados e chegarem a versões finais de produtos, podem causar prejuízos materiais consideráveis e, em algumas aplicações, tais como o mercado de eletrônica automotiva, colocam em risco a vida de outras pessoas.

Este trabalho apresenta uma solução completa de inspeção de placas de circuitos impressos, capaz de reconhecer erros de curto-circuito e trilhas abertas, além de larguras de trilhas e espaçamento entre trilhas fora da especificação de projeto da placa. A análise é feita na placa corroída, sem verniz e sem serigrafia, ou então antes da corrosão, na placa com a emulsão fotossensível (*photoresist*) já revelada. É importante ressaltar que o processo de inspeção visual é mais fácil de ser aplicado a placas com camada simples ou dupla camada, mas é possível adaptá-lo para a inspeção de placas multicamada, caso sejam analisadas cada uma das camadas corroídas antes de prensá-las juntas numa placa só.

1.1.OBJETIVO E JUSTIFICATIVA

O reconhecimento de falhas no processo de corrosão de placas de circuito impresso por visão computacional apresenta algumas vantagens sobre os métodos de inspeção humana e os testes elétricos. A principal delas, talvez, é a maior simplicidade: o sistema baseado em visão não requer contato físico com a placa, podendo ser adaptado a placas de diversos tamanhos e formas sem a necessidade de fixação específica, além de ser facilmente instalado e não sofrer desgaste mecânico, tornando seus custos de manutenção muito baixos. Outro ponto importante é que a inspeção visual automatizada pode ser facilmente adaptada para outras necessidades (como a análise estética da placa final, por exemplo), já que a

lógica de inspeção depende fundamentalmente do algoritmo empregado. Por fim, em comparação ao processo de inspeção humana, o processo automatizado possui potencialmente maior confiabilidade, e permite o rastreamento e análise futura dos erros e falhas encontrados.

1.2.METODOLOGIA

O projeto de desenvolvimento de um detector de falhas em circuitos impressos pode ser dividido em três grandes partes: o algoritmo, a interface com o usuário e o hardware.

O desenvolvimento do algoritmo de detecção de falhas, por sua vez, foi dividido em duas etapas: a de segmentação da imagem e a de detecção de falhas na imagem já binarizada. Testes preliminares com placas envernizadas mostraram que seria inviável segmentar as placas ao final do processo de produção. Por isso se decidiu pela análise das placas sem verniz, que poderiam ser mais facilmente segmentadas. A segmentação é feita com algoritmos de tratamento de imagens bem estabelecidos no ramo de visão computacional. Para a segunda etapa, por outro lado, foi feita extensa pesquisa da produção científica na área de análise de PCIs, e foram estudadas as metodologias encontradas que pareceram mais relevantes. Ao final deste processo, desenvolveu-se um algoritmo original, mas que sintetiza algumas ideias interessantes dos métodos pesquisados.

A interface com o usuário foi desenvolvida com uma biblioteca própria para esse fim, sempre levando em conta a usabilidade do programa para que a experiência do usuário fosse a melhor possível.

A parte do projeto de hardware divide-se, por sua vez, em duas partes: a estrutura para aquisição de imagens e o sistema embarcado que executará o algoritmo. O sistema de aquisição foi todo projetado e construído pelo grupo, e foi dada atenção especial ao controle da iluminação do ambiente. O sistema embarcado utilizado foi o open-source *BeagleBoard-xM*.

2. FUNDAMENTAÇÃO TEÓRICA

2.1. TRABALHOS REALIZADOS POR MOGANTI, ET AL. (1996)

Em (Moganti, et al., 1996), é apresentado um levantamento dos métodos de detecção de falhas em placas de circuito impresso conhecidos até então. Neste estudo foi feita uma análise de grande parte dos artigos publicados sobre o assunto e as técnicas foram classificadas de acordo com a finalidade e a metodologia.

A primeira classificação definida no artigo separa os métodos em métodos elétricos/com contato - onde se encontram os mais tradicionais métodos para análises de falhas em placa, como a cama de agulhas - e em métodos não elétricos/sem contato. A Figura 1 apresenta um diagrama em árvore destes métodos não elétricos/sem contato.

Os tópicos a seguir descrevem cada um destes métodos, com ênfase naqueles do tipo inspeção óptica/visual automática (em inglês *Automatic Visual/Optical Inspection*, AOI), já que estão diretamente relacionados ao escopo deste projeto.

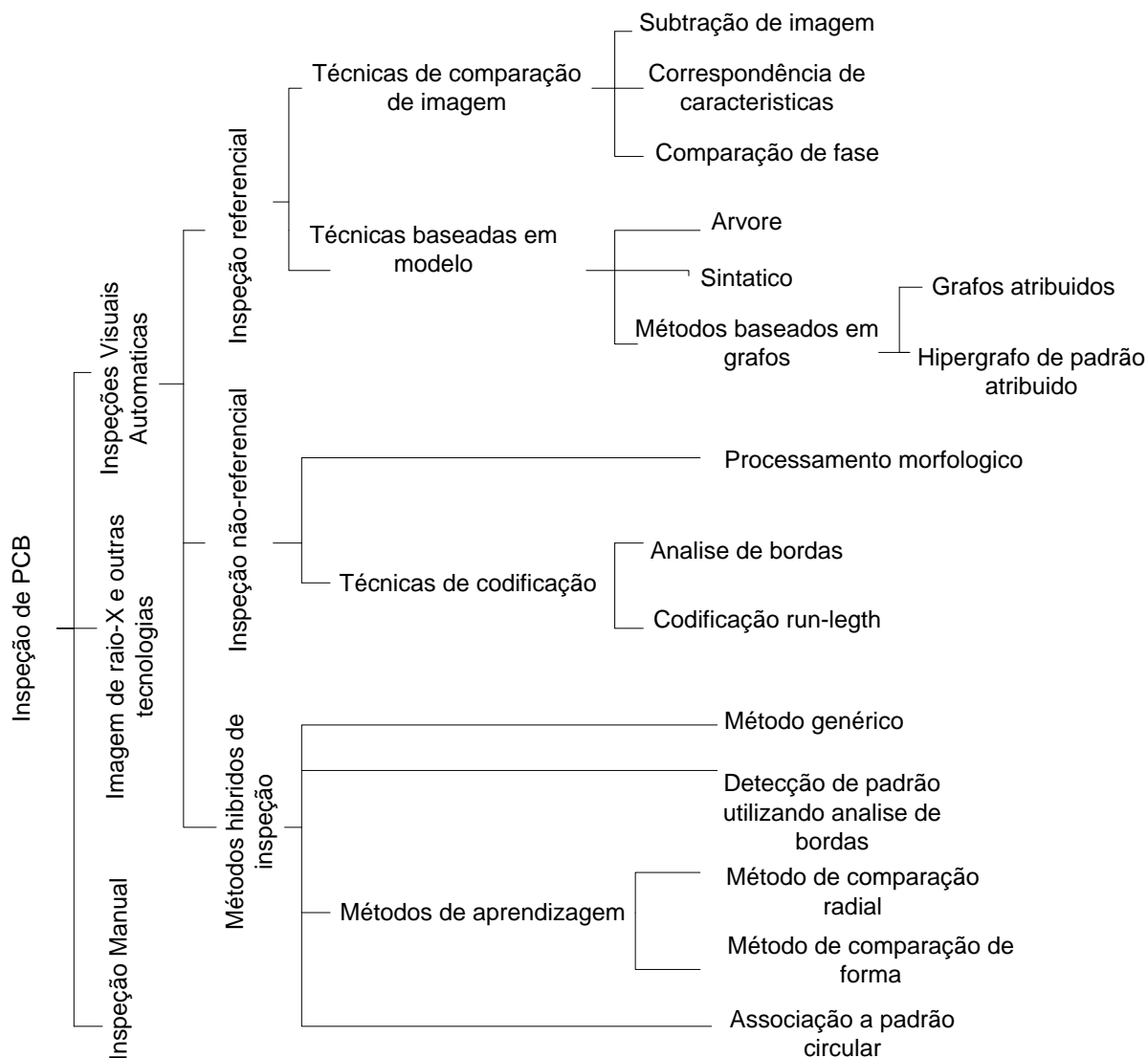


Figura 1 – Árvore taxonômica de Moganti et al. (1996)
 Fonte: Autoria própria. Baseado em (Moganti, et al., 1996)

2.1.1. Imagens por Raios-X e outras tecnologias

Não apenas sensores sensíveis à luz visível são utilizados em inspeção automática de PCBs, outros métodos de imageamento também são empregados por possuírem certas vantagens específicas. Os citados em (Moganti, et al., 1996) são os listados a seguir.

- Sistema de imagem por raios-x: usada para rápida e precisa medição de placas multicamadas;
- Imagem de ultrassom: especialmente útil para detectar defeitos em juntas de solda;
- Imagem térmica: capazes de detectar falhas em placas de circuito impresso em operação (reconhecendo áreas super- ou subaquecidas).

2.1.2. Inspeção Ótica/Visual Automática

Esses métodos caracterizam-se pela capacidade de detectar, além dos tipos de falhas detectáveis com a inspeção manual, outros tipos de falhas ou potenciais falhas, como espessuras de trilha erradas, e até mesmo defeitos estéticos.

2.1.2.1. Inspeção referencial

Executa uma comparação ponto a ponto com uma referência, que pode ser uma imagem ou um desenho de uma placa sem defeitos.

- **Técnicas por comparação de imagem**

Subtração de imagem: é o tipo de algoritmo mais simples e a abordagem mais direta para o problema de inspeção de PCIs. A imagem da placa a ser testada é simplesmente subtraída da placa de referência e o resultado indica os defeitos encontrados. É extremamente sensível a diferenças no alinhamento das placas.

Correspondência de características: é uma forma aperfeiçoada de subtração de imagem, na qual as características extraídas da placa testada e as características do modelo são comparadas e não a imagem inteira. Esse método reduz a sensibilidade e aumenta a robustez do sistema.

Comparação de fase: compara as informações de fase da imagem de referência com a imagem da placa testada. Nesse método são as transformadas de

Fourier normalizadas das imagens de referência e da placa testada que são comparadas, e não as imagens no domínio do espaço.

- **Técnicas baseadas em modelo**

São técnicas que comparam a imagem da placa a ser testada com um conjunto de modelos. Subdividem-se em métodos baseados em grafos, em árvores, ou métodos sintáticos.

Os métodos baseados em grafos utilizam as propriedades estruturais, topológicas e geométricas da imagem.

- **Grafos atribuídos:** O grafo atribuído é um grafo de relações espaciais na forma de atributos direcionados, que descrevem a conectividade e a relação entre a vizinhança. Uma função de similaridade é então utilizada para mensurar o quão bem o grafo de cena corresponde ao grafo do modelo. É um algoritmo custoso computacionalmente.
- **Hipergrafo de padrão atribuído:** Nessa representação, as regiões segmentadas são representadas por um grafo atribuído regional e a informação é representada em dois níveis diferentes: o mais baixo, que liga componentes dentro de uma mesma região e o mais alto, que liga as diferentes regiões. Essa separação torna o algoritmo mais eficiente.

2.1.2.2. Inspeção não referencial

Esses métodos funcionam verificando se alguma característica da placa não está de acordo com o padrão de design especificado para a placa. Erros como trilhas fora de um intervalo definido de espessuras, por exemplo, podem ser detectadas a partir desse método.

- **Processamento morfológico**

Envolve as operações de expansão e contração. São utilizados para verificar se as trilhas atendem à especificação mínima de espessura e mínimo espaçamento entre trilhas, e se os furos estão corretamente posicionados.

- **Técnicas de codificação**

Subdividem-se em técnicas de análise de borda e codificação *run-length*.

Técnicas de análise de borda: se baseiam na representação das bordas de uma forma tratável, seguidas de um processo de verificação. Defeitos são definidos como sendo bordas com características distintas do padrão de uma trilha.

A codificação *run-length*: conta pixels de traços contínuos ao longo de cada coluna e linha da imagem da PCB e constrói um histograma horizontal e vertical. A largura mínima do condutor é verificada checando se o *run-length* de pixel é menor do que um certo limite.

2.1.2.3. Métodos híbridos

Servem-se tanto da análise referencial quanto da análise não referencial para detectar falhas.

- **Método genérico**

É uma combinação de algoritmos referenciais e não-referenciais. A imagem é transformada para obter uma imagem do esqueleto, a partir do qual defeitos e boas características de circuito podem ser facilmente detectados. As características da placa testadas são então comparadas com uma referência e características conflitantes indicam defeito.

- **Detecção de padrões usando análise de borda**

Para falhas no condutor. O algoritmo detecta áreas com possíveis defeitos a partir da técnica de análise de bordas. Essas áreas são, em seguida, um algoritmo de detecção de padrões é aplicado para medir a largura do condutor.

- **Correspondência de padrão circular**

Uma técnica baseada na codificação radial para inspeção de alta velocidade. Uma janela de 32x32 é movida ao longo da imagem da PCI da esquerda para a direita e de cima para baixo. Para detectar se a largura de um conector é menor do que d , um círculo com raio d é desenhado com o centro do círculo na borda da trilha. Se ao longo da trilha, o círculo continuar sempre dividido ao meio, então a trilha não apresenta defeito.

- **Métodos de aprendizagem**

Subdivide-se em algoritmos de comparação radial e método de comparação de forma.

Algoritmos de comparação radial: nesse método o usuário insere amostras de aprendizagem de PCIs sem defeito e o sistema converte esses padrões em características conhecidas.

Método de comparação de forma: o usuário insere uma amostra para aprendizagem e o sistema quebra a imagem de entrada em pequenos segmentos. O sistema guarda pequenos padrões de cada segmento em um arquivo de referência. A mesma operação é realizada com a placa a ser testada. Se ela não apresentar padrões que correspondam aos padrões da referência, o sistema julga o produto defeituoso.

2.2. TRABALHOS REALIZADOS POR (TATIBANA, ET AL., 1997)

Tatibana, et al. (1997) apresentam um novo método de inspeção de placas de circuito impresso baseado na conectividade das trilhas. Esse método se aplica somente para detecção de falhas do tipo curto-circuito e circuito aberto.

A ideia central do método é verificar se dois furos conectados por trilhas na placa de referência também estão conectados na placa de teste. Para isso, o primeiro passo do algoritmo é de detectar os furos na imagem de referência. Isso é

feito operando a segmentação da imagem da placa, seguida de uma operação de preenchimento de furos (*fill*, em morfologia (Gonzalez, et al., 2008)). Subtraindo a imagem segmentada da imagem preenchida, obtém-se a imagem dos furos na placa. As imagens resultantes após cada operação na placa de referência são mostradas na Figura 2. A Figura 2 (a) ilustra uma área da placa segmentada, com os furos abertos, e em Figura 2(b) após a operação de preenchimento dos furos. A imagem Figura 2(c) é o resultado da operação de subtração Figura 2(b) - Figura 2(a). Para as etapas seguintes do algoritmo é necessária também a posição dos centroides dos furos, conforme ilustrado na imagem Figura 2(d).

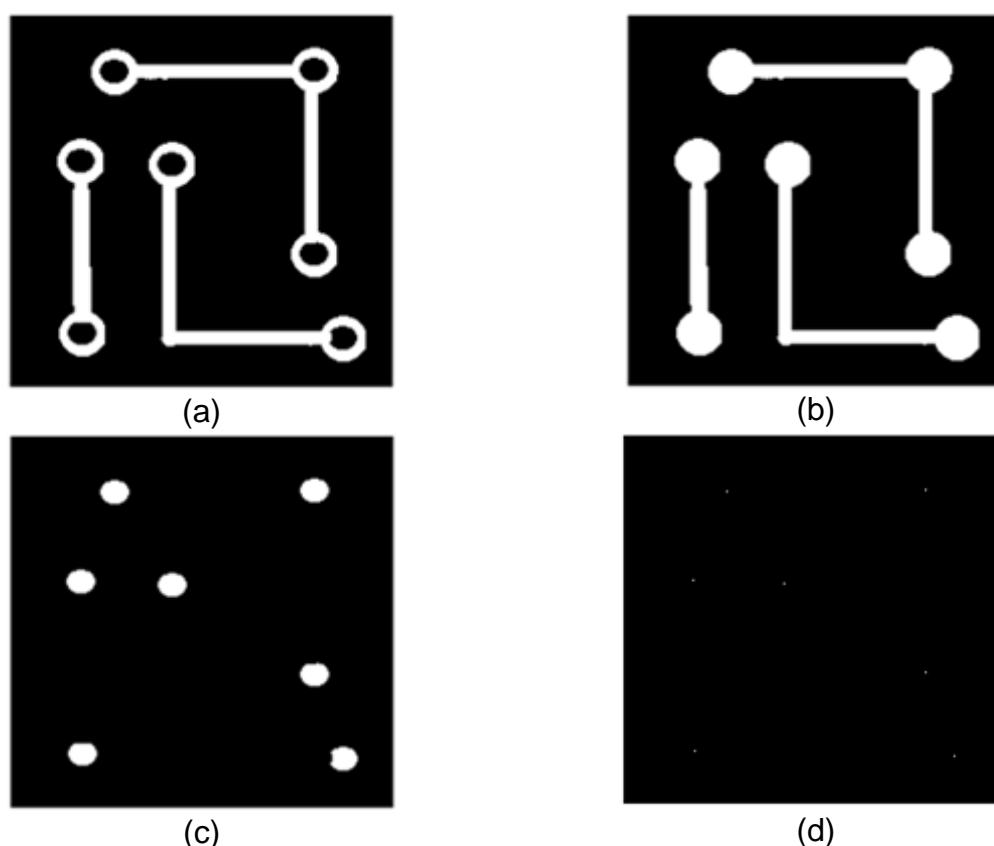


Figura 2 - Figura 2 - (a) Imagem da PCI segmentada; (b) Imagem após operação de 'fill'; (c) Imagem dos furos após a operação (a) - (b); (d) centroides dos furos.

Fonte: Autoria própria

Em seguida, é feita uma análise de componentes conectados na imagem preto-e-branca, e é atribuído um *label* para cada um dos objetos presentes na imagem. Dessa forma, cada trilha da placa é enumerada e recebe um identificador único.

A partir da imagem dos centroides dos furos é feita uma operação de subdivisão da imagem em regiões de Voronoi, que utiliza como critério a menor distância euclideana a um objeto (Theodoridis, et al., 2008). Neste caso, os objetos são os centroides dos furos da placa. Em seguida, uma vez que cada furo possui sua região definida, é possível fazer a comparação das regiões com as trilhas enumeradas, construindo-se então uma tabela relacionando furos (regiões) e trilhas. A figura 4 apresenta um exemplo destes passos. Na figura 4(a) podem ser observados os furos enumerados, em 4(b) as suas respectivas regiões e em 4(c) as trilhas enumeradas (*labels*). É importante observar que as trilhas podem atravessar regiões diferentes, mas cada trilha deve receber apenas um *label*. O objetivo final deste processo é relacionar cada trilha da placa ao furo a ela conectado.

A partir das tabelas relacionais, as seguintes verificações são feitas:

- Se o número de furos na placa de referência e na placa testada for diferente, verifica-se que há falta (ou excesso) de furos;
- Se a uma região na placa de referência corresponder a mais de um furo na placa testada, ou então, se nenhum furo na placa testada corresponder a uma região na placa de referência, a placa testada apresenta um defeito;
- Se duas regiões que estão conectadas por uma trilha na placa de referência não estão conectadas na placa testada, verifica-se que a trilha está aberta;
- Se, pelo contrário, duas regiões que não estão conectadas na placa de referência estão conectadas na placa testada, verifica-se que há um curto-circuito nessa placa.

Para ilustrar a detecção de um circuito aberto através deste conjunto de regras, pode-se considerar o exemplo da Figura 3 como a placa de referência e o exemplo da Figura 4 como a placa a ser testada, na qual há duas trilhas abertas (Figura 4(a)). Na placa de referência, analisando-se as Figura 3(b) e (c), conclui-se que as regiões 1, 2 e 5 estão conectadas pela trilha 1, as regiões 3 e 6 estão conectadas pela trilha 2 e as regiões 4 e 7 estão conectadas pela trilha 3. Já na

placa testada, a região 1 está desconectada das regiões 2 e 5, e a região 3 está desconectada da região 6.

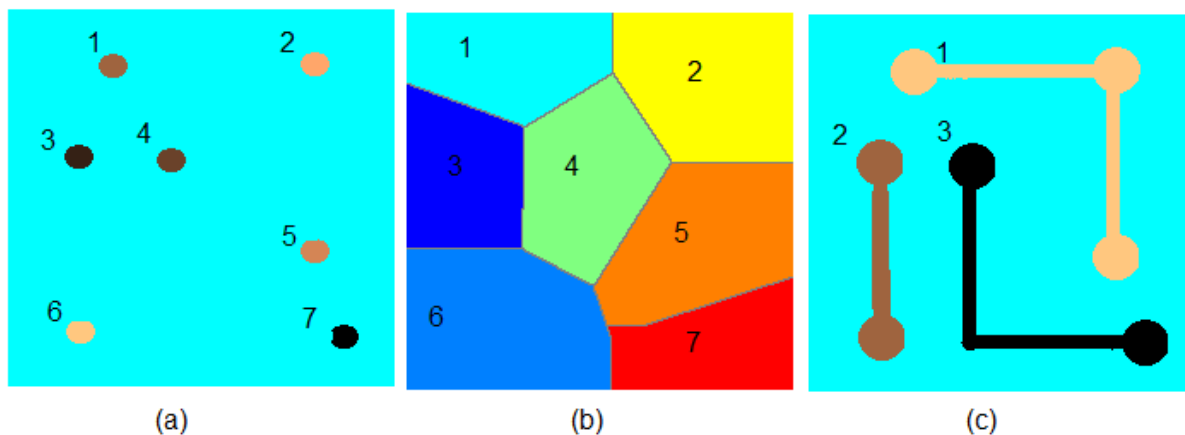


Figura 3 - (a) furos da placa de referência enumerados; (b) regiões de Voronoi da placa de referência; (c) Trilhas da placa de referência enumeradas
 Fonte: Autoria própria

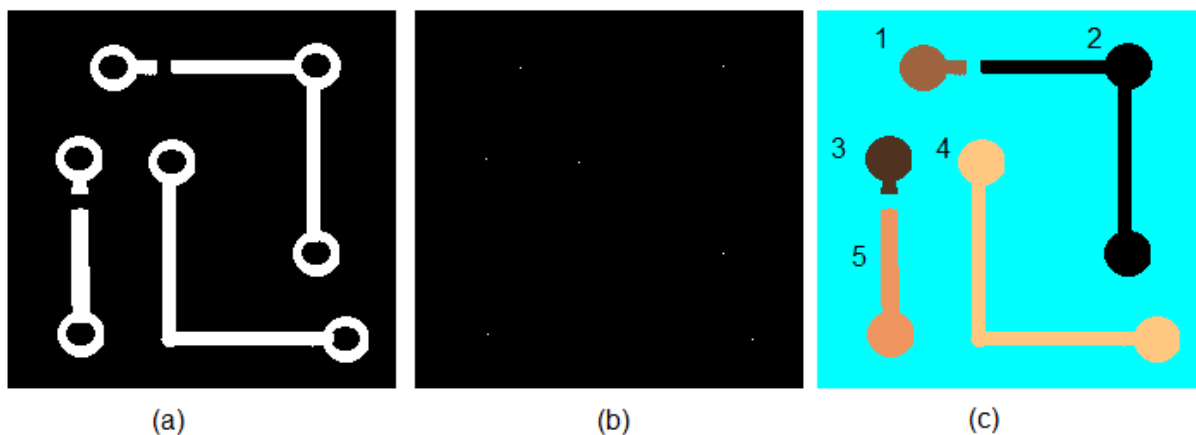


Figura 4 - (a) placa de teste binarizada; (b) centroide dos furos da placa testada (c) trilhas enumeradas da placa testada. Devido aos defeitos do circuito aberto, a enumeração agora é diferente daquela obtida na placa de referência (Figura 3(a)).
 Fonte: Autoria própria

A desvantagem deste método é que ele se baseia na suposição de que toda trilha terá ao menos dois furos. Essa abordagem não permite avaliar placas antes da corrosão ou placas que possuam trilhas não perfuradas, o que é cada vez mais comum, já que a placa em questão pode possuir componentes SMD, que cada vez mais são utilizados em detrimento dos componentes *through hole*.

A Figura 5 apresenta um diagrama em blocos que resume o método proposto em (Tatibana, et al., 1997), explicado nesta seção.

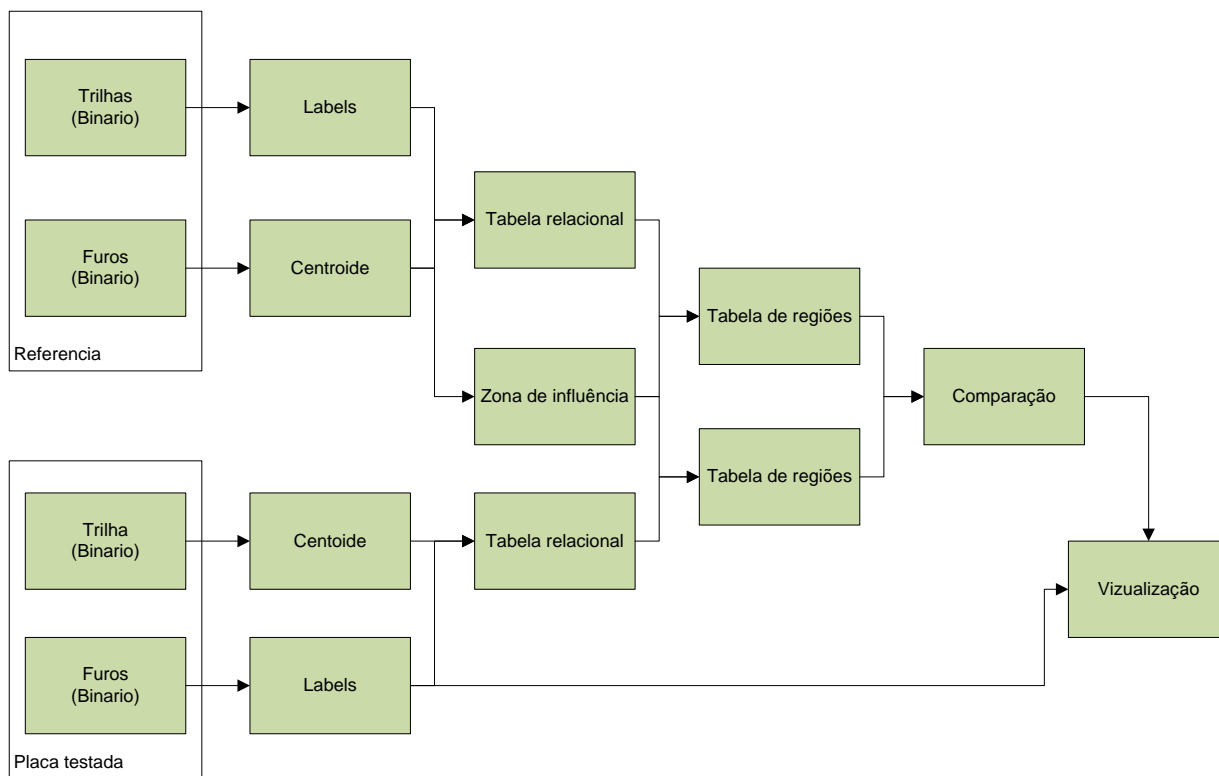


Figura 5 - Método de inspeção de PCB baseada em conectividade.
Fonte: Autoria própria, baseada em (Tatibana, et al., 1997)

3. DESENVOLVIMENTO

3.1. VISÃO GERAL

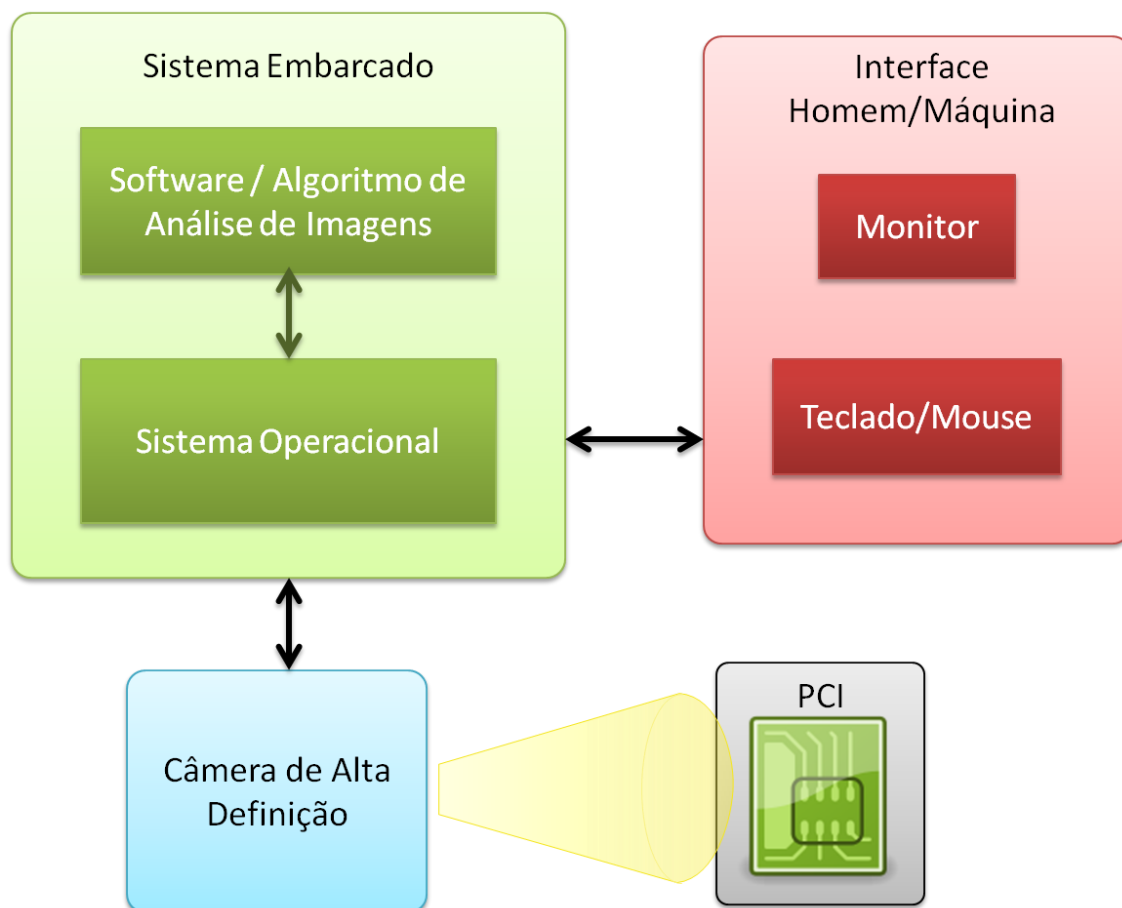


Figura 6 - Diagrama em blocos do sistema desenvolvido

O diagrama em blocos da Figura 6 apresenta a visão geral do sistema. Os tópicos a seguir descrevem cada um dos blocos funcionais.

3.2. ALGORITMO PARA DETECÇÃO DE FALHAS EM PCIS

3.2.1. Estudos em MATLAB

Em processamento de imagens, a ferramenta MATLAB é extremamente útil quando se deseja testar um algoritmo. Ela permite testar as funcionalidades de um algoritmo de uma forma simples, por conta do grande número de funções de

tratamento de sinais e imagens disponibilizadas no programa, e da facilidade de manipulações de matrizes e vetores.

3.2.1.1. Discussão sobre a segmentação da imagem

Em um primeiro momento foram utilizadas placas de circuito impresso envernizadas, como mostrado na Figura 7. O primeiro problema identificado ao trabalhar com esse tipo de imagem foi a sensibilidade a diferenças de iluminação de todos os algoritmos testados. Por esse motivo foi necessário construir um sistema de aquisição das imagens da placa com iluminação controlada, descrito em um capítulo específico. Os próximos tópicos descrevem os algoritmos testados e os problemas encontrados.

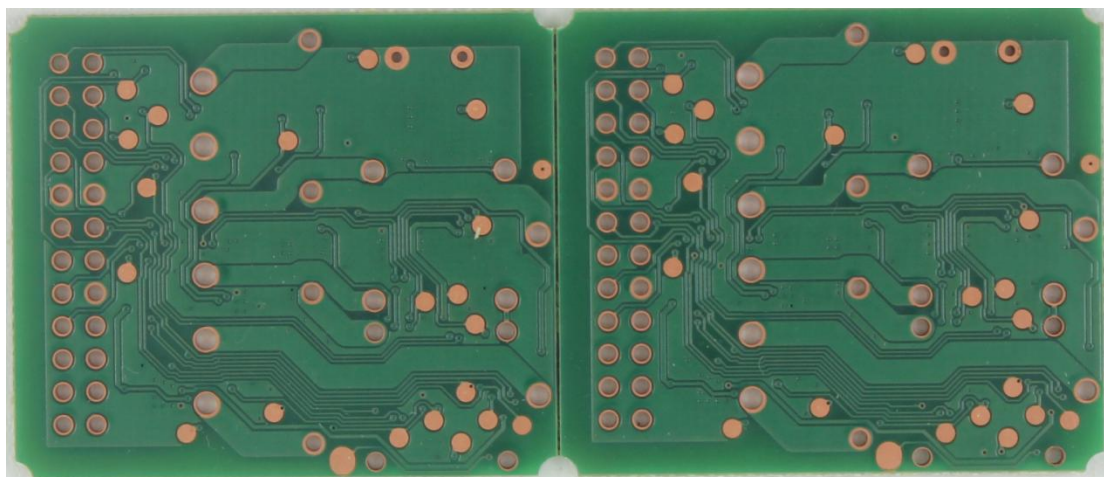


Figura 7 - Placa utilizada nos primeiros testes de segmentação.
Fonte: Autoria própria

Nas placas envernizadas, os condutores tem aspecto muito semelhante ao background. A segmentação por limiar (ou *threshold*) foi descartada logo no início, já que demonstrou péssimos resultados. Nesse ponto é importante notar que, como as trilhas têm uma pequena espessura, operações de morfologia, como open e close, se usadas para tentar melhorar o resultado da segmentação, poderiam mascarar ou até mesmo inserir problemas na placa.

As primeiras tentativas de segmentação foram feitas com o algoritmo de Otsu sobre a imagem em níveis de cinza, que obtém o valor do limiar de segmentação através da minimização da variância intraclasse das duas regiões que serão

segmentadas pela operação de *threshold* (Otsu, 1979). Otsu prova que isso equivale a achar o valor de k que maximiza a equação da variância interclasse (**Erro! Fonte de referência não encontrada.**):

$$\sigma_B^2 = \frac{[\mu_T \omega(k) - \mu(k)]^2}{\omega(k)[1 - \omega(k)]} \quad (1)$$

Infelizmente o contraste da imagem não é suficiente para separar as trilhas do *background* da imagem.

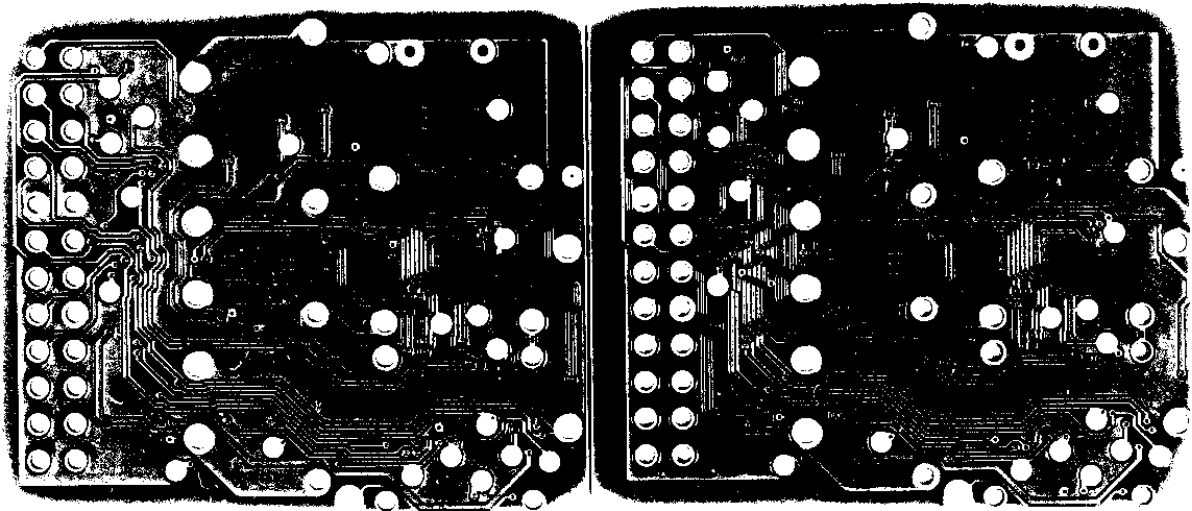


Figura 8 - Resultado da segmentação utilizando o algoritmo de Otsu para definir o nível de threshold
Fonte: Autoria própria

Então foram feitos testes com a função *adapthisteq*, do *Image Processing Toolbox* do MATLAB (Mathworks Inc., 2011). A função *adapthisteq* é um algoritmo que aumenta o contraste de blocos na imagem equalizando o histograma de cada bloco da imagem. Essa sequência, seguida de uma operação de *threshold* resultou na Figura 9. É possível perceber que os contornos, mesmo com a operação de aumento de contraste, não são bem definidos e as cores das trilhas e do *background* se misturam.

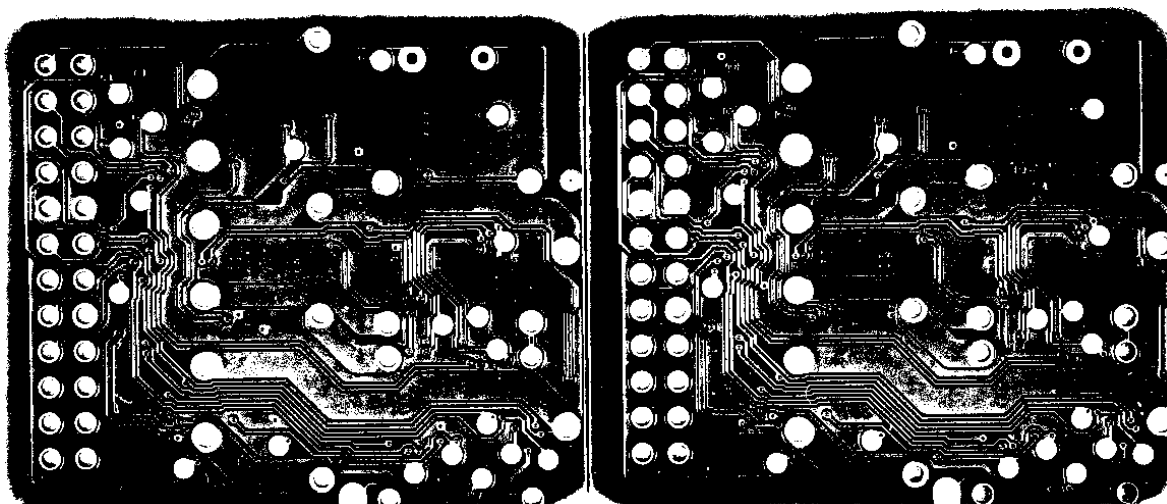


Figura 9- Resultado da segmentação com adapthisteq
Fonte: Autoria própria

Assim, optou-se pelo teste de algoritmos que operam diretamente sobre os canais de cor da imagem para realizar a segmentação. O algoritmo *k-means* (Theodoridis, et al., 2008), por exemplo, é bastante utilizado para separar uma imagem em regiões, e faz isso com procurando k centroides, onde k é o número de regiões em que se deseja dividir a imagem, de forma que a soma das variâncias dos pixels de cada região para a sua média seja a menor possível. Esse algoritmo é pouco flexível, isso porque o número de regiões em que a imagem será dividida deve ser predefinido, então se for decidido por separar a imagem em duas regiões, ela deve conter apenas trilhas (verde claro) e background (verde escuro). Neste caso, se um pedaço da carcaça aparecer, ela será confundida com uma das regiões ou o algoritmo pode nem ao menos funcionar.

Foram realizados diversos testes, variando os três seguintes parâmetros no algoritmo (Figura 10, 10 e 11):

- O número de *clusters* k , com 3 ou 4 regiões;
- O espaço de cores da imagem, em RGB, monocromática (tons de cinza) ou a imagem em tons de cinza passada por uma transformada logarítmica;
- A inicialização do algoritmo, com ou sem valores iniciais de médias setados pelo usuário.

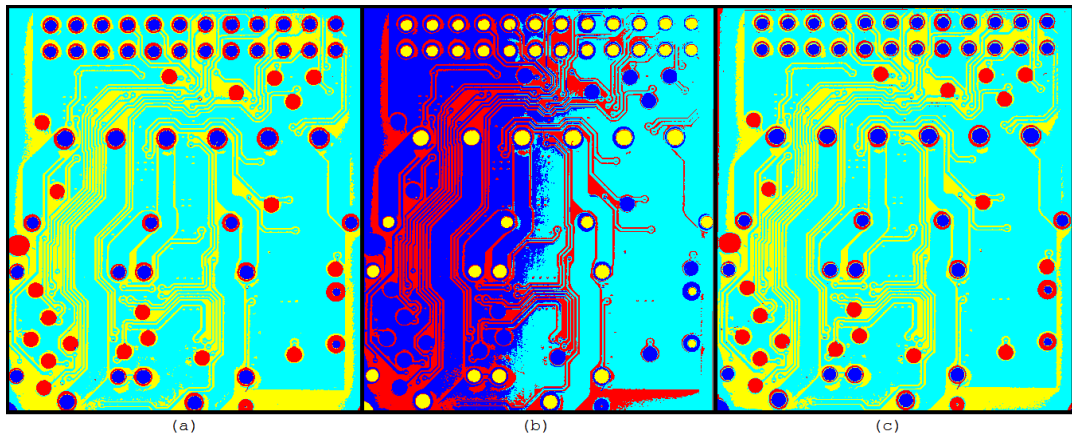


Figura 10 - Resultados k-means com 4 regiões e sem valores iniciais (a) RGB (b) Log (c) Gray
Fonte: Autoria própria

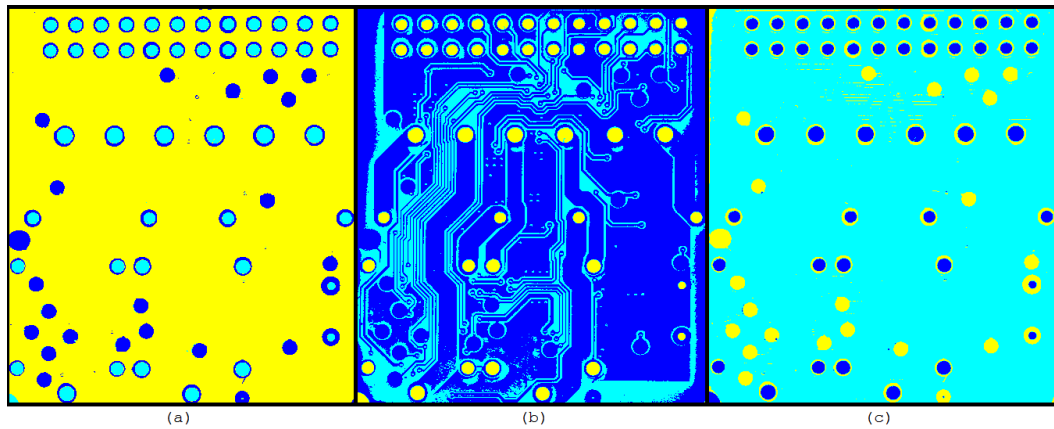


Figura 11 - Resultados k-means com 3 regiões e sem valores iniciais (a) RGB (b) Log (c) Gray
Fonte: Autoria própria

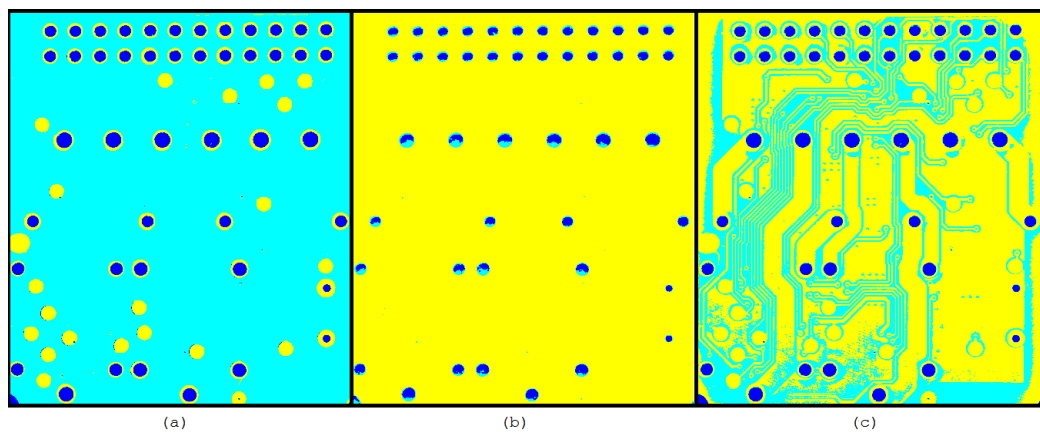


Figura 12 - Resultados k-means com 3 regiões e com valor inicial (a) RGB (b) Log (c) Gray
Fonte: Autoria própria

A Tabela 1 mostra os erros percentuais de cada variação do algoritmo comprada com o *ground-truth*. Os erros percentuais são dados em percentagem dos pixels classificados incorretamente, comparando o que foi classificado como trilha, fundo, ou buraco com o *ground-truth*, que foi desenhado manualmente por cima da imagem original.

Tabela 1 - Erros percentuais de cada variação do algoritmo de k-means

K	Tipo de imagem	Valor inicial	Erros
4	RGB	Não	6,78%
4	Log	Não	14,05%
4	Gray	Não	6,75%
3	RGB	Não	12,89%
3	RGB	Sim	12,89%
3	Log	Não	5,07%
3	Log	Sim	10,91%
3	Gray	Não	12,86%
3	Gray	Sim	5,46%

Fonte: Autoria própria

No melhor dos casos, o algoritmo *k-means* aplicado a uma imagem no espaço logarítmico, com três regiões e sem inicialização das médias, resulta em um erro de 5,07%. Pelas Figura 10, Figura 11 e Figura 12, é visível que o maior erro está nas bordas e é resultado de uma iluminação não uniforme somado ao fato de as trilhas terem cores muito semelhantes ao fundo. Esse erro, mesmo que pequeno percentualmente, impossibilita qualquer algoritmo de detecção de falhas em placas de circuito impresso, pois insere falhas na imagem segmentada que não existem na imagem original. Nesse caso, por exemplo, mesmo a olho nu, fica claro que um curto-circuito inexistente seria detectado.

Além dos algoritmos conhecidos em processamento de imagens, alguns métodos propostos em artigos científicos também foram testados, como é o caso dos algoritmos de (Jian, et al., 2010), de (Chen, et al., 2010) e de (Gokturk, et al.), mas ainda assim não foi possível obter um resultado satisfatório para placas envernizadas.

Tendo em vista que nenhum algoritmo de segmentação apresentou um resultado satisfatório, decidiu-se por analisar placas de circuito impresso sem verniz, ainda em uma etapa intermediária do processo produtivo, e o que foi verificado é

que neste caso não é necessário utilizar algoritmos tão complexos como o *k-means* para a segmentação das trilhas, bastando que a iluminação seja uniforme.

3.2.1.2. Algoritmo desenvolvido para detecção de falhas

Após um estudo dos algoritmos apresentados em diversos artigos científicos, o algoritmo de (Tatibana, et al., 1997), citado também no artigo (Leta, et al., 2008) foi escolhido para ser testado. Esse algoritmo, detalhado no capítulo 2, se baseia na conectividade de trilhas para encontrar falhas de circuito aberto e curto-circuito.

Antes de desenvolver uma solução própria, o algoritmo apresentado em (Tatibana, et al., 1997) foi implementado e testado, em um primeiro momento, comparando pares de figuras geradas a partir de arquivos Gerber, um sem nenhum defeito, e outro com defeitos inseridos manualmente. O primeiro teste foi feito com uma placa simples, igual à placa mostrada no artigo (Figura 13).

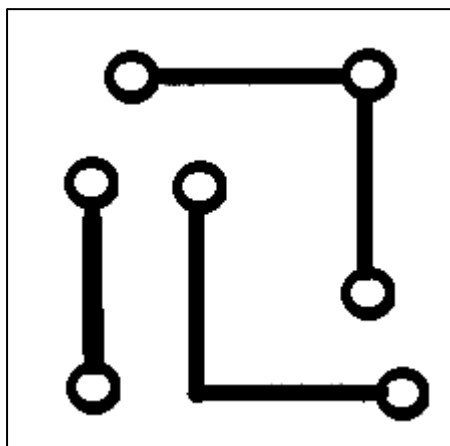


Figura 13 - Imagem da placa original, utilizada como referência
Fonte: Autoria própria

Nesta placa foram inseridos erros manualmente, que podem ser observados na Figura 14.

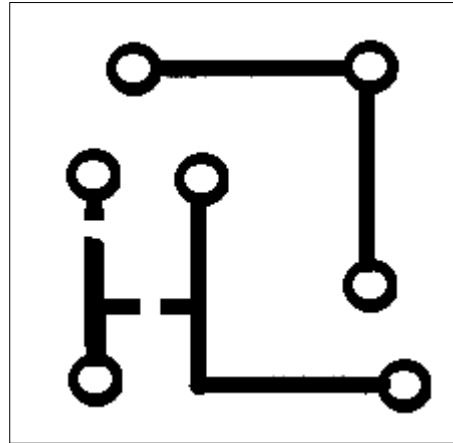


Figura 14 - Imagem com falhas inseridas
Fonte: Autoria própria

Após um resultado positivo na placa simples (todos os erros foram detectados corretamente), uma placa mais complexa foi testada. Os resultados da comparação da placa sem (Figura 15) e com defeito (Figura 16) são mostrados na Figura 17.

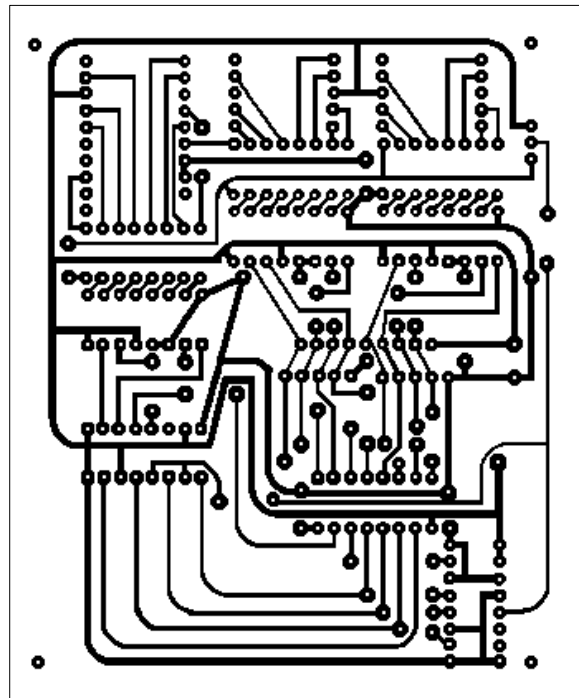


Figura 15 - Placa de referência
Fonte: Internet (2012°)

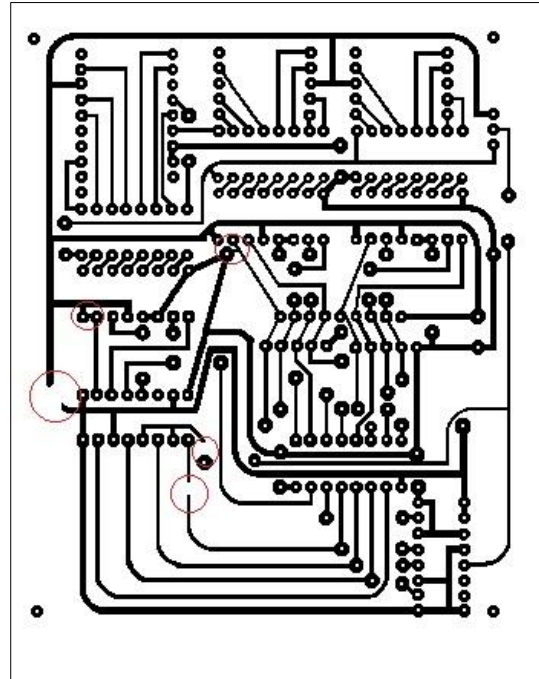


Figura 16 - Placa com defeitos inseridos manualmente
 Fonte: Internet (2012)

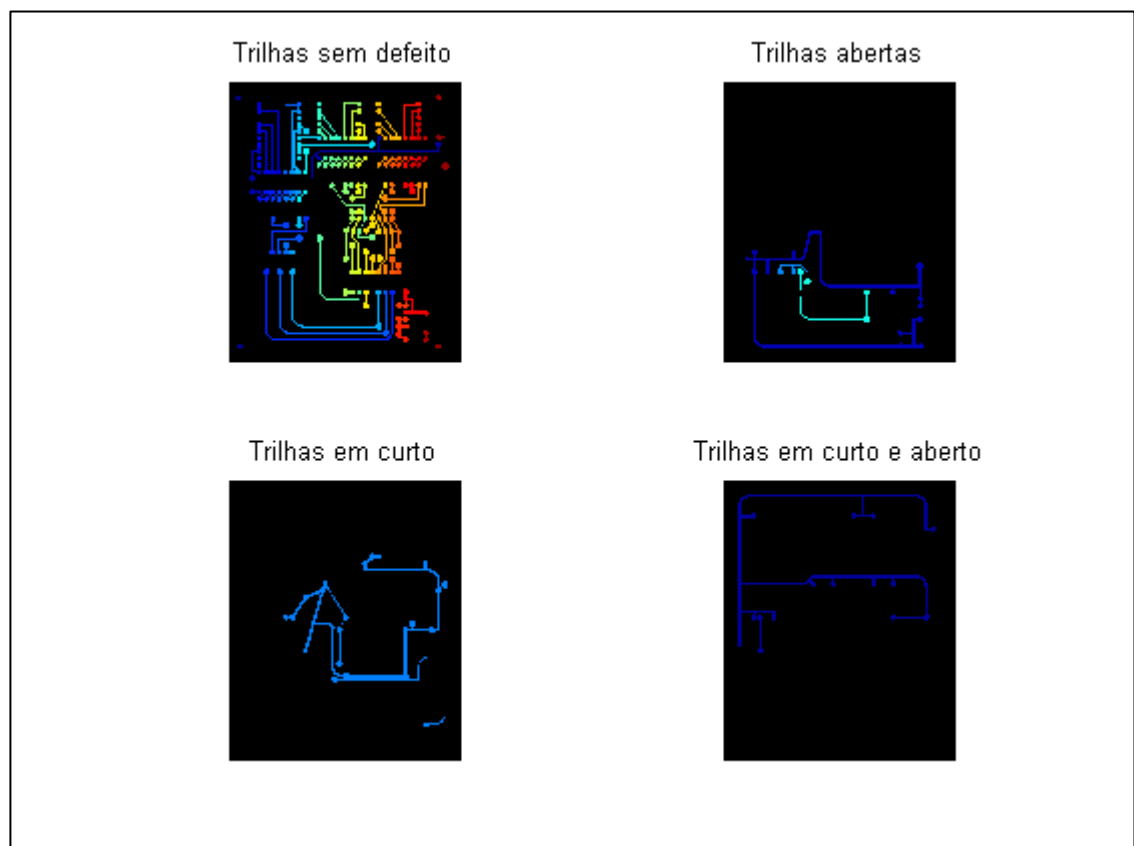


Figura 17 - Resultado do algoritmo de Tatibana et al.
 Fonte: Autoria própria

Este algoritmo se apresentou eficiente na detecção de erros, entretanto, como mencionado no capítulo 2, apresenta limitações de ordem prática, já que para que ele funcione, é necessário que todas as trilhas tenham ao menos duas extremidades perfuradas. Isso limita a aplicação a placas destinadas a conectar apenas componentes do tipo *through-hole*, o que na maioria das vezes não é o caso.

Tendo em vista essa limitação, uma nova ideia foi desenvolvida utilizando o mesmo fundamento do algoritmo acima apresentado. Nesse algoritmo, ao invés de utilizar os furos como referência para separar a imagem em regiões, as próprias trilhas são utilizadas. Dessa forma, a robustez do algoritmo quanto ao alinhamento da placa é aumentada e não é necessário que todas as trilhas sejam perfuradas.

O algoritmo funciona da seguinte forma: quando uma placa é testada, o algoritmo separa trilha por trilha da imagem segmentada e faz uma intersecção de cada uma com a placa de referência rotulada. Calcula-se, então, o histograma da imagem de intersecção e verifica-se qual rótulo é mais frequente em cada uma das regiões. Com isso, é possível criar uma tabela que relaciona cada trilha na placa de testes a uma trilha na placa de referência. A mesma operação, com a placa a ser testada no lugar da placa de referência e a placa de referência no lugar da placa a ser testada, é realizada para obter a tabela que relaciona cada trilha na placa de referência a uma trilha na placa de testes.

Apenas com essas duas tabelas é possível verificar se uma trilha está aberta ou curto-circuitada. Isso porque se, na primeira tabela, duas ou mais trilhas na placa que está sendo testada estiverem relacionadas a uma mesma trilha na placa de referência, essa trilha está aberta. Agora se, por sua vez, na segunda tabela, duas ou mais trilhas na placa de referência estiverem relacionadas a uma trilha na placa que está sendo testada, é verificado que essas trilhas estão curtadas. A Figura 17 e as Tabelas 2 - Relação de cada trilha na placa de teste a uma trilha na placa de referência, Tabela 2 e Tabela 3 ilustram um exemplo.

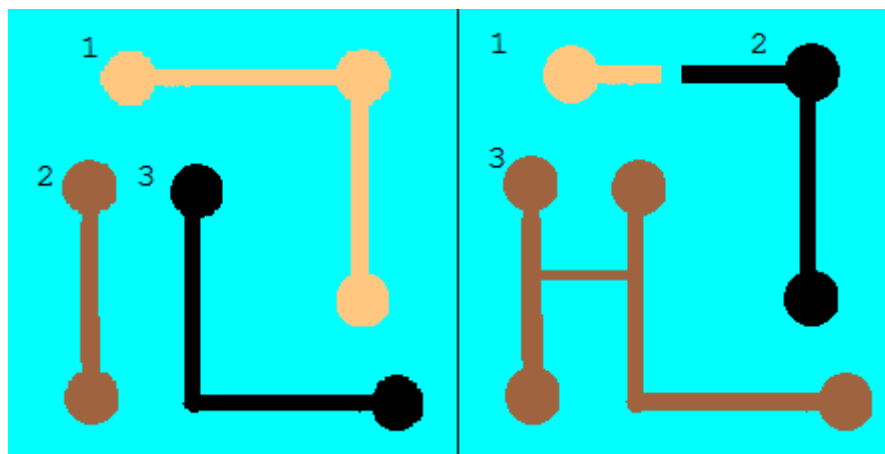


Figura 18 - Exemplo do algoritmo de intersecção
Fonte: Autoria própria

Nesse exemplo as duas tabelas criadas teriam a seguinte forma:

Tabela 2 - Relação de cada trilha na placa de teste a uma trilha na placa de referência

Trilha (placa testada)	Trilha (placa de referência)
1	1
2	1
3	3

Fonte: Autoria própria

Tabela 3- Relação de cada trilha na placa de referência a uma trilha na placa de teste

Trilha (placa referência)	Trilha (placa testada)
1	2
2	3
3	3

Fonte: Autoria própria

Segundo a Tabela 2, a trilha 1 está aberta, pois na placa testada ela se divide em duas. Já segundo a Tabela 3, as trilhas 2 e 3, que deveriam estar separadas, estão unidas na placa de teste, o que indica um curto-circuito dessas duas trilhas. A região que está ligando - no caso de um curto-circuito - ou separando - no caso de um circuito aberto - duas trilhas é a borda que separa as duas regiões da coluna direita das tabelas acima na imagem de intersecção.

Para detectar a localização exata da abertura da trilha, as mascaras das trilhas de referência são sobrepostas a imagem de contorno das regiões de Voronoi (regiões de mínima distância em relação a cada trilha) da placa que está sendo testada. Já para detectar a localização exata de um curto, as mascaras das trilhas curtadas na placa testada são sobrepostas a imagem de contorno de regiões da placa de referência. Assim, no primeiro caso, por exemplo, apenas os contornos das áreas que dividem duas trilhas abertas irão restar e será possível delimitar uma “caixa” que contenha aqueles pixels, exatamente no ponto de falha.

3.2.2. Estudos em Outros Softwares

Além do *MATLAB*, dois outros softwares foram utilizados para realizar um estudo dos algoritmos de segmentação. O primeiro deles foi o *imLab* (Scuri, 2009), software que permite realizar operações de tratamento de imagem sem a necessidade de programar as operações desejadas. Nele é possível testar diferentes operações em diferentes sequências. É uma forma mais prática do que o *MATLAB* de verificar o efeito de diferentes operações em uma imagem. Ela foi utilizada principalmente para testar algoritmos de detecção de borda.

Para binarizar as imagens após o tratamento, o software *Image_EII*, desenvolvido no *Institut National de Sciences Appliquées* de Rennes - FR, foi utilizado. O funcionamento desse software é igual ao *imLab*, mas ele possui uma função de *threshold* adaptável que o *imLab* não possui.

O primeiro teste foi feito com o gradiente, que detecta bordas pela derivada de primeira ordem da imagem nas duas direções, horizontal e vertical. O resultado é mostrado na Figura 19.

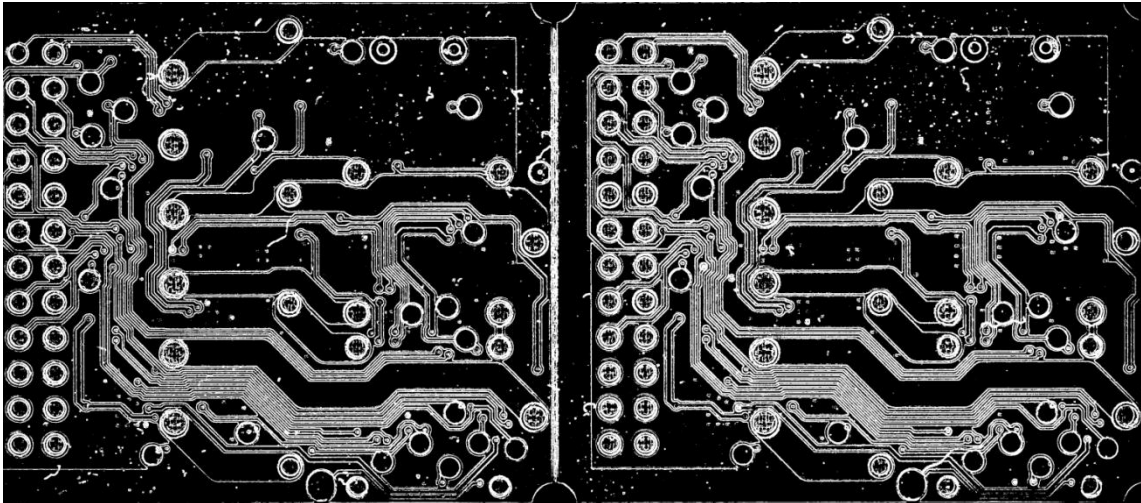


Figura 19 – Aplicação do filtro de gradiente
 Fonte: Autoria própria

O filtro de Sobel também é um operador discreto diferencial, e utiliza as seguintes matrizes para filtrar a imagem nos eixos x e y, respectivamente:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \quad G_y = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$$

O resultado da aplicação deste filtro, seguida de binarização, é mostrado na Figura 20.

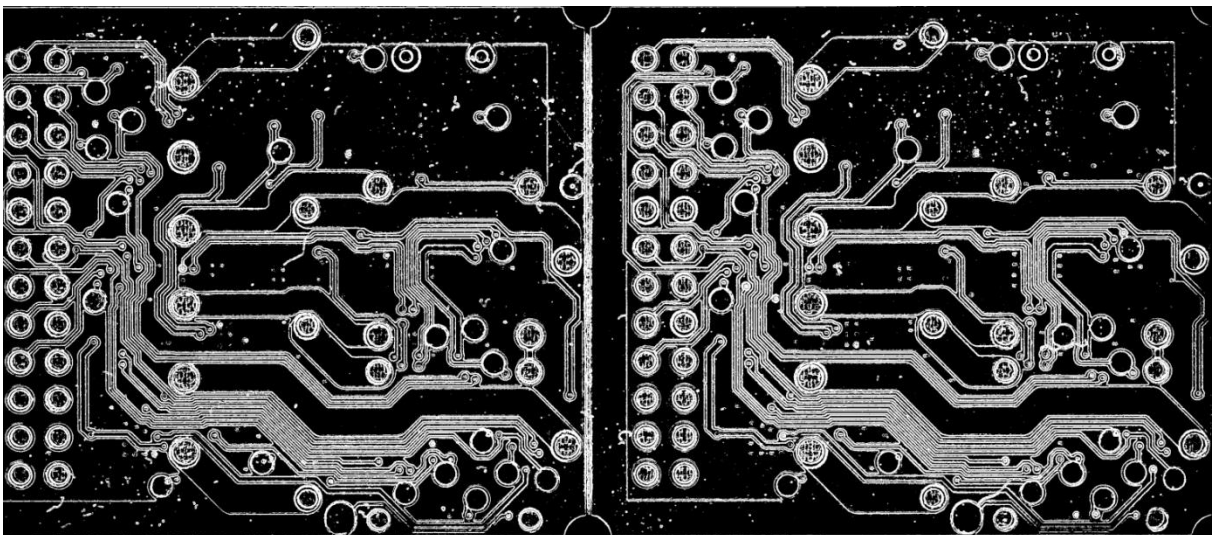


Figura 20 – Aplicação do filtro de Sobel
 Fonte: Autoria própria

Outra tentativa foi a de usar o filtro Laplaciano 5x5, que realiza a convolução da imagem com uma matriz da forma:

$$G = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 24 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$

O resultado da convolução é mostrado na Figura 21.

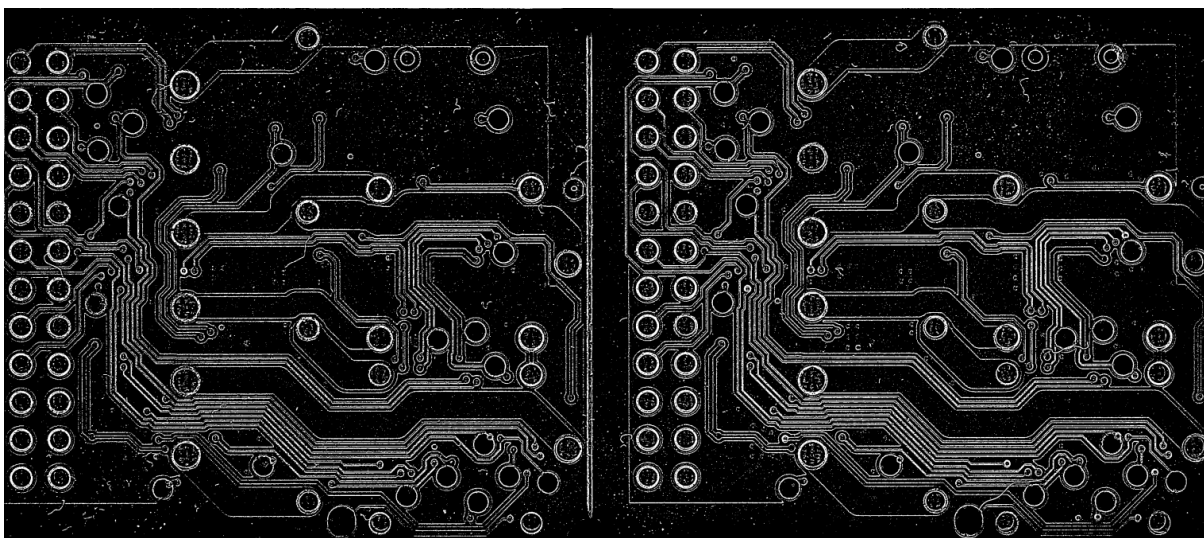


Figura 21 - Aplicação do operador Laplaciano 5x5
Fonte: Autoria própria

3.2.3. Implementação com o OpenCV

O algoritmo adaptado baseado na intersecção de regiões explicado na seção 2.2, depois de testado no MATLAB, foi implementado em C++. Para isso, foram utilizadas muitas das funções da biblioteca OpenCV (Bradski, 2000), que já contém os algoritmos mais conhecidos de tratamento de imagens implementados de uma forma otimizada. Essa biblioteca é uma referência para os que desejam trabalhar com tratamento de imagens e é bastante difundida.

3.2.3.1. Binarização da imagem

A binarização, como foi mostrado nos estudos feitos no Matlab, é uma tarefa simples se houver contraste suficiente na imagem. Ao limitar a aplicação à placas

ainda não corroídas e a placas corroídas mas não envernizadas, a tarefa de segmentar a imagem é simplificada. Não foi então necessário utilizar os algoritmos mais complexos de segmentação de imagens que haviam sido estudados. Foi verificado que basta que a iluminação da placa seja uniforme para que a binarização da placa seja capaz de separar o condutor do background da placa.

3.2.3.2. Enumeração das trilhas

A função `cvFindContours` da biblioteca OpenCV retorna uma sequência de contornos encontrados na imagem. Essa função utiliza o método descrito em (Suzuki, et al., 1985). A partir dessa sequência, uma imagem, na qual cada contorno é preenchido com um número diferente, é criada. Para isso, a função `cvDrawContours` é utilizada e a cada contorno é atribuído um número sequencial. Com estes passos, é feito o *labeling* de cada uma das trilhas das placas.

3.2.3.3. Separação da placa em regiões

A separação da placa em regiões foi feita utilizando a função `cvDistTransform` do OpenCV. Essa função separa a imagem em regiões de mínima distância, também conhecidas como regiões de Voronoi utilizando o método descrito em (Borgefors, 1986). No algoritmo em questão, a distância é calculada em relação à trilha mais próxima e os pontos nas fronteiras das regiões têm distâncias iguais em relação a duas ou mais trilhas.

3.2.3.4. Detecção de curto-circuito e circuito-aberto

A detecção de circuito aberto e fechado, como mencionado no estudo no MATLAB, foi feita com a imagem da intersecção das trilhas de referência com a imagem enumerada da placa testada e a intersecção das trilhas da placa testada com a imagem enumerada da placa de referência. Para a operação de intersecção foi utilizada a função `mul`, que multiplica ponto a ponto a máscara com a imagem enumerada.

A partir dessa intersecção, a trilha correspondente é determinada pelo número da trilha que aparece mais vezes na imagem da intersecção. Para

determinar qual número de trilha aparece mais vezes, foi utilizada a função `cvCalcHist`, que divide a imagem em um histograma de valores. Nesse histograma, o valor que se repetir mais vezes é o número da trilha correspondente à trilha da máscara.

3.2.3.5. Detecção de *over etch* e *under etch*

Under-etch e *over-etch* são os nomes dados à falha de produção de PCIs quando há sub-corrosão ou sobre-corrosão de alguma região cobreada da placa. Esses erros levam à produção de placas com trilhas mais finas ou mais grossas do que o especificado em projeto. A detecção destas falhas foi feita com o mesmo algoritmo de detecção de circuito aberto e curto-circuito. Para isso é necessário, em primeiro lugar, estabelecer a mínima distância entre trilhas e a mínima espessura de trilha especificadas no projeto da placa a ser testada. Essa informação, em conjunto com a relação entre a quantidade de pixels na imagem e a distância em centímetros na placa, permite transformar essas distâncias em um número de pixels.

A operação morfológica *open* com um elemento estruturante de $n \times n$ pixels é equivalente à aplicar uma operação de erosão seguida de dilatação. Essa operação faz com que objetos de espessura menor do que n sejam eliminados. Já a operação morfológica de *close*, com o mesmo elemento estruturante de $n \times n$ pixels, realiza a operação de dilatação seguida de erosão, o que faz com que dois objetos, antes separados por uma distância menor do que n , se unam.

Esses dois princípios de morfologia, aplicados separadamente à imagem da placa segmentada, cada um com um elemento estruturante adequado às mínimas distâncias antes estabelecidas, fazem com que as trilhas com espessura menor do que a mínima estabelecida sejam abertas e que as trilhas separadas por uma distância menor do que a especificada sejam curto-circuitadas. Por esse motivo, como foi citado no início do texto, é possível utilizar o mesmo algoritmo para detectar falhas de *over etch* e *under etch*. É importante notar que, nesta configuração, o algoritmo não permite estabelecer limites diferentes para diferentes trilhas em uma mesma placa.

3.2.4. Análise do Algoritmo

Depois da implementação do algoritmo foi feita uma análise de seu funcionamento com o software Valgrind (Nethercote, et al., 2006) (Valgrind-developers, 2011). Esse software é capaz de verificar se toda memória alocada ao longo do programa foi corretamente liberada ou se algum ponteiro foi perdido. Em geral, em programas de tratamento de imagem, trabalha-se muito com esse tipo de operação. A biblioteca OpenCV utiliza diversas variáveis referenciais e, como parte dela está escrita em C, utiliza as funções de malloc e free para alocar e liberar memória. Por isso foi dada certa atenção para prevenir problemas deste tipo, que causam falhas difíceis de ser detectadas e corrigidas.

Além dessa funcionalidade, esse software permite a visualização da ocupação da memória ao longo da execução do programa. O resultado para o programa de detecção de falhas em placa é mostrado na Figura 22. Nessa análise é visível que a memória ocupada, de 17MiB, é bem menor do que memória RAM disponível no kit de desenvolvimento utilizado no projeto (1GiB).

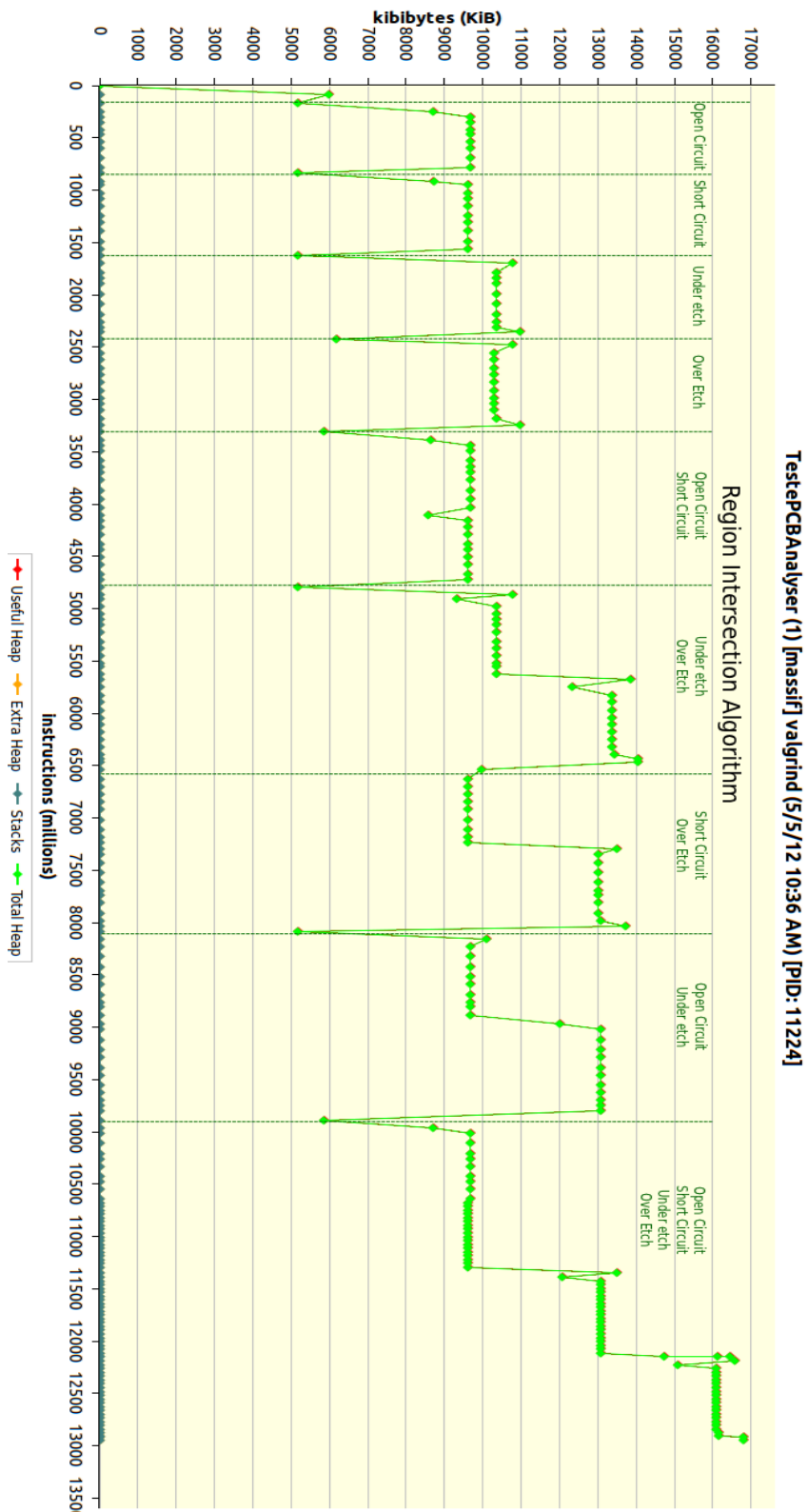


Figura 22- Uso da HEAP ao longo do programa
Fonte: Autoria própria

3.3. SISTEMA DE AQUISIÇÃO DE IMAGENS

O sistema de aquisição de imagens é composto de dois componentes básicos: a estrutura física, que engloba posicionamento da câmera, da PCI, e sistema de iluminação, e a câmera fotográfica empregada. O projeto deste sistema foi uma etapa importante para o andamento do projeto, porque só depois dele pronto pode-se configurar o algoritmo de processamento das imagens. Conforme descrito na seção anterior, a escolha do algoritmo empregado é intimamente dependente de parâmetros tais como iluminação ambiente, posição da PCI em relação à câmera ou se a placa poderia estar rotacionada com relação à imagem padrão adquirida num primeiro passo da etapa de análise. Foi verificado que modificações relativamente simples no sistema de aquisição permitiram uma grande simplificação na complexidade do algoritmo para detecção dos erros. Detalhes sobre a estrutura de aquisição concebida, sobre a construção da estrutura e sobre as escolhas de projeto feitas serão descritas detalhadamente neste capítulo.

3.3.1. Estudo preliminar para o projeto da estrutura física

Algoritmos de visão computacional são extremamente dependentes de conhecimentos prévios sobre o problema a ser resolvido, e esses conhecimentos podem ser de natureza semântica (por exemplo, regras rígidas, como um objeto do tipo A nunca pode estar conectado a um objeto do tipo B) ou sobre aspectos ambientais de como a exposição fotográfica foi feita. Problemas complicados de se resolver com algoritmos de visão podem ser solucionados de maneira trivial quando se muda o ponto de vista pelo qual se encara o problema. Um exemplo clássico são os refinados algoritmos capazes de reconhecer objetos mesmo que ele esteja rotacionado ou escalonado, como o SURF (Bay, et al., 2008) e o SIFT (Lowe, 2004) (para citar dois algoritmos bem conhecidos). O problema resolvido por essas abordagens é naturalmente válido e bastante útil até, mas somente a partir do momento que não se tem controle ou conhecimento nenhum sobre as imagens de entrada do algoritmo. A partir do momento que se possui algum tipo de conhecimento sobre a disposição do objeto de interesse, ou se pode adicionar restrições para que exista somente um número pequeno de cenários possíveis para as imagens de entrada, o problema pode se tornar muito simples.

Por esse motivo, tomou-se cuidado especial em pensar e desenvolver uma estrutura física que simplificasse ao máximo os cenários e possibilidades para as imagens de entrada, e que por meio de suposições simples sobre elas fosse possível diminuir ao máximo o trabalho de programação e projeto do algoritmo.

Para isso, foram listadas as características da aquisição que podem variar entre duas exposições fotográficas consecutivas, foram analisados os impactos da variação de cada uma delas na complexidade final do algoritmo, e finalmente foram concebidas soluções que minimizassem a variação das características tidas como críticas pela análise. A análise foi feita pensando-se no efeito da variação de uma dessas características, mantidas todas as outras constantes. A lista das variáveis analisadas se encontra abaixo:

3.3.1.1. Distância entre a placa e a câmera

A variação da distância entre a placa e a câmera provocaria variações de tamanho da placa entre duas exposições fotográficas diferentes. Como a fotografia é uma projeção de uma realidade tridimensional em um sensor de duas dimensões, um objeto ocupa uma área tanto maior na fotografia quanto menor é a distância entre ele e o sensor, até o ponto em que não fosse possível, devido a limitações da lente, focalizar o objeto. Como o algoritmo concebido, conforme descrito no capítulo anterior, toma como base uma imagem de referência, se houver diferenças entre o tamanho da PCI sob teste e a PCI de referência, seria necessário desenvolver um algoritmo que fosse imune a esse tipo de variação, ou redimensionar alguma das duas imagens.

3.3.1.2. Ângulo entre o plano que contém a PCI e o plano que contém o sensor da câmera

O princípio fundamental da fotografia estabelece que é a projeção de uma realidade tridimensional num sensor bidimensional, a rotação do plano que contém um objeto bidimensional e o plano que contém o sensor de uma câmera fotográfica provoca distorções no tamanho e nas proporções do objeto fotografado. Quando se fotografa um quadrado perfeito, por exemplo, esse quadrado só aparece na fotografia como um quadrado caso ele esteja paralelo ao sensor no momento da

exposição. Caso o quadrado esteja inclinado com relação a apenas um eixo, ele aparecerá na foto como um trapézio, cujo lado menor é o lado do quadrado mais afastado da câmera, e o lado maior é o lado do quadrado mais próximo da câmera. Caso haja rotação em dois eixos, o quadrado pode acabar sendo representado na fotografia como um quadrilátero irregular. Por isso, para que seja possível inclusive chegar a conclusões sobre a largura das trilhas e dos espaçamentos entre as trilhas na fotografia, é necessário que não haja rotação entre esses planos. Caso existisse, seria necessário desenvolver um algoritmo mais complicado para transformar a imagem de modo que as distorções fossem compensadas. Um problema dessa abordagem seria ter de corrigir a resolução espacial dos pixels obtidos após a transformação, possivelmente tendo de recorrer a interpolações complexas. Uma alternativa seria recorrer à geometria epipolar e técnicas de visão estéreo, que são muito bem descritas em (Hartley, et al., 2004), que envolvem mais de uma câmera e complicados métodos de calibração, ou então fluxo de imagem, o que só pode ser feito quando se tem a liberdade de variar alguma característica espacial da câmera (rotacioná-la ou transladá-la). É, portanto, importante que essa característica varie o mínimo possível.

3.3.1.3. Rotação da PCI com relação ao eixo perpendicular ao sensor da câmera

A variação dessa característica sofre dos mesmos problemas descritos na análise da característica anterior, com a ressalva que, quando não é modificada a distância entre nenhum dos pontos da PCI e o sensor da câmera, a correção da resolução espacial dos pixels é bem mais simples.

3.3.1.4. Posicionamento da placa

Assim como a rotação da PCI com relação a qualquer um dos eixos, se houver diferença entre a posição da placa dentro da fotografia entre fotos consecutivas, com todas as outras variáveis mantidas constantes, ainda assim deveriam ser tomadas atitudes para reconhecer alguma característica da placa e a partir dela inferir a posição da PCI na fotografia. Outra maneira seria pedir que o usuário indicasse ao algoritmo, por meio de alguma interface simples (cliques de mouse, por exemplo), a posição da PCI. Como a correção da posição não necessita

de correções na resolução espacial dos pixels da fotografia (o que não envolve nenhum tipo de algoritmo de interpolação), desde que sejam corrigidas as distorções causadas pela lente da câmera (o que quase toda câmera digital moderna já faz internamente), isso seria uma solução simples de se implementar, já que a resolução espacial da fotografia é uniforme em toda sua extensão.

3.3.1.5. Iluminação uniforme

Esta foi uma variável cuja análise inicial apontava a necessidade apenas de se manter uma iluminação constante entre sucessivas exposições fotográficas. Pensou-se, a princípio, que se a iluminação fosse constante, seria possível utilizar o mesmo método de segmentação da imagem em diferentes imagens, ou pelo menos se poderia “calibrar” o algoritmo para que ele fosse capaz de segmentar de maneira robusta fotografias diferentes da mesma placa. Isso se mostrou um problema maior que o esperado: a qualidade da segmentação depende diretamente da qualidade da iluminação da placa. Em algumas placas iluminadas de maneira não uniforme, não foi possível conceber um algoritmo capaz de segmentá-las; principalmente nos casos em que pontos pertencentes a regiões distintas possuíam coordenadas de cor muito parecidas. Pensou-se em utilizar métodos de *clustering* para possibilitar a segmentação dessas placas problemáticas, mas isso inseriria complicação adicional e mais uma fonte de incerteza no sistema. Se houvesse um método simples e mais confiável de segmentar a imagem, variando o aparato de iluminação do sistema, ela seria muito bem vinda.

A Tabela 4 resume todas as variáveis analisadas, o impacto da variação de cada uma, com as outras mantidas constantes, na complexidade do algoritmo, e um parecer sobre a necessidade de cada umas das variáveis ser ou não controlada.

n.	Variável	Impacto no algoritmo	Necessidade de ser Controlada
1	Distância entre plano da PCI e plano do sensor	Ter que compensar diferenças de tamanho e resoluções espaciais	Alta
2	Rotação entre plano da PCI e plano do sensor	Ter de compensar diferenças de tamanho e resoluções espaciais, com algoritmos refinados de correção de distorção	Alta
3	Rotação entre a PCI e o eixo do sensor	Ter de encontrar uma maneira de reconhecer a angulação da placa de teste e de referência, e rotacionar uma das duas	Média
4	Posição da PCI	Achar onde a placa se encontra, ou fazer o usuário informar a posição. Não há necessidade de interpolação para correção de resolução espacial.	Baixa
5	Iluminação controlada e uniforme.	Afeta diretamente a segmentação, um ponto crítico do algoritmo.	Alta

Tabela 4- Requisitos para a estrutura física do sistema de aquisição de imagens
Fonte: Autoria própria

3.3.2. Concepção do Sistema de Aquisição

Tendo em mãos uma lista das variáveis que deveriam ser controladas, puderam-se avaliar as opções existentes para sistemas de iluminação e aquisição das imagens.

3.3.2.1. Soluções Encontradas

As alternativas encontradas foram:

3.3.2.1.1. *Soft Boxes com iluminação e câmera fixadas com tripés*

Foram encontrados diversos sistemas comerciais baseados em softboxes, que nada mais são que “caixas” de pano ou de outro material semi-translúcido (foram encontradas alternativas na internet, e até mesmo em (Borba, 1994), que utilizavam papel, o que é inadmissível para um ambiente industrial). A foto de um sistema como esse pode ser vista na Figura 23 - Sistema de aquisição com SoftBoxe custava US\$153,00.



Figura 23 - Sistema de aquisição com SoftBox
Fonte: (Amazon.com, Inc., 2012)

Os problemas dessa montagem são fundamentalmente dois: a posição relativa entre a câmera, as fontes luminosas e a PCI são difíceis de controlar, e a iluminação é influenciada por variações da iluminação ambiente. Além disso, a PCI teria que ser apoiada verticalmente na caixa, o que é uma solução bastante insatisfatória devido à instabilidade e à dificuldade de operação que isso acarretaria, ou então ter que conceber uma adaptação para que a câmera pudesse ser posicionada acima da caixa.

3.3.2.1.2. *Mesa estativa para fotografia*



**Figura 24 - Mesa estativa para fotografia comercializada pela empresa Tecnocroma.
Fonte: (Tecnocroma Imagem e Comércio Ltda., 2012)**

Foram encontradas mesas estativas comerciais, sistemas como o ilustrado na Figura 24, que se prestam basicamente a facilitar a digitalização de objetos tais como fotografias analógicas, livros e radiografias. É um sistema robusto, com controle fino de posicionamento da câmera, mas cuja iluminação é difícil de ser controlada e é extremamente dependente da iluminação ambiente. Além disso, o custo de uma mesa como esta varia, no mercado nacional, entre R\$500,00 e R\$700,00 (segundo orçamento feito pelo grupo com a empresa Tecnocroma).

Aliado a isso, esse sistema é mais propenso a erros já que o usuário deveria saber, de alguma forma, qual foi a posição da câmera utilizada para se fotografar a placa de referência.

3.3.2.1.3. Construção de uma caixa rígida com apoio para a câmera e iluminação interna

Uma caixa rígida, de madeira ou metal, com apenas um orifício com diâmetro pouco maior que o diâmetro da lente empregada, na qual se pudesse fixar a câmera, e que possuísse um sistema de iluminação interna, seria um sistema que se adequaria a todos os requisitos levantados na seção anterior. O posicionamento da placa poderia ser feito com pequenas marcações no fundo da caixa, a posição relativa entre a câmera e a PCI seria constante, e a iluminação seria perfeitamente

controlável e independente do meio ambiente externo, o que é adequado ao ambiente industrial em que o sistema será utilizado.

É evidentemente mais trabalhoso construir a caixa do zero do que comprar uma solução pronta, mas com a escolha adequada dos componentes e das peças utilizadas na construção, pode-se construir um sistema mais eficiente por um custo menor.

3.3.2.2. Comparação dos três sistemas

A Tabela 5 sumariza as conclusões tiradas quanto às três soluções encontradas:

Solução	Requisito n. da Tabela 4 é atendido?				
	1	2	3	4	5
SoftBox	não	sim	não	não	não
Estativa	sim	sim	sim	sim	não
Construção de uma caixa	sim	sim	sim	sim	sim

Tabela 5 - Comparação das soluções com os requisitos de projeto.

Fonte: Autoria própria

Como visto na tabela, a solução que melhor atende aos requisitos, conquanto que mais trabalhosa, é a construção de uma caixa projetada especialmente para a tarefa. Como pontos positivos, essa também é a solução mais barata de todas, e com maior facilidade de alteração no decorrer do projeto.

3.3.3. Construção do Sistema de Aquisição de Imagens

Um rascunho da solução concebida para a aquisição de imagens se encontra na Figura 25. Foi construída uma caixa de madeira de 60cm x 40 cm x 40cm, com abertura frontal. O tampo superior possui um furo de 10 cm de diâmetro e um suporte, também de madeira, para fixar a câmera, que também foi revestido de borracha a fim de não a danificar. A caixa foi toda revestida por dentro com papel refletor prateado. Foram presas às paredes laterais duas lâmpadas fluorescentes cilíndricas do tipo T8, e ao tampo superior, em volta do orifício para passagem da lente da câmera, uma lâmpada fluorescente circular. Para difundir a luz foram utilizados tecidos brancos com malha fina, presos como cortinas às laterais e ao

topo da caixa. A câmera é fixada à caixa através de um parafuso padrão, do mesmo tamanho utilizado em tripés comerciais.

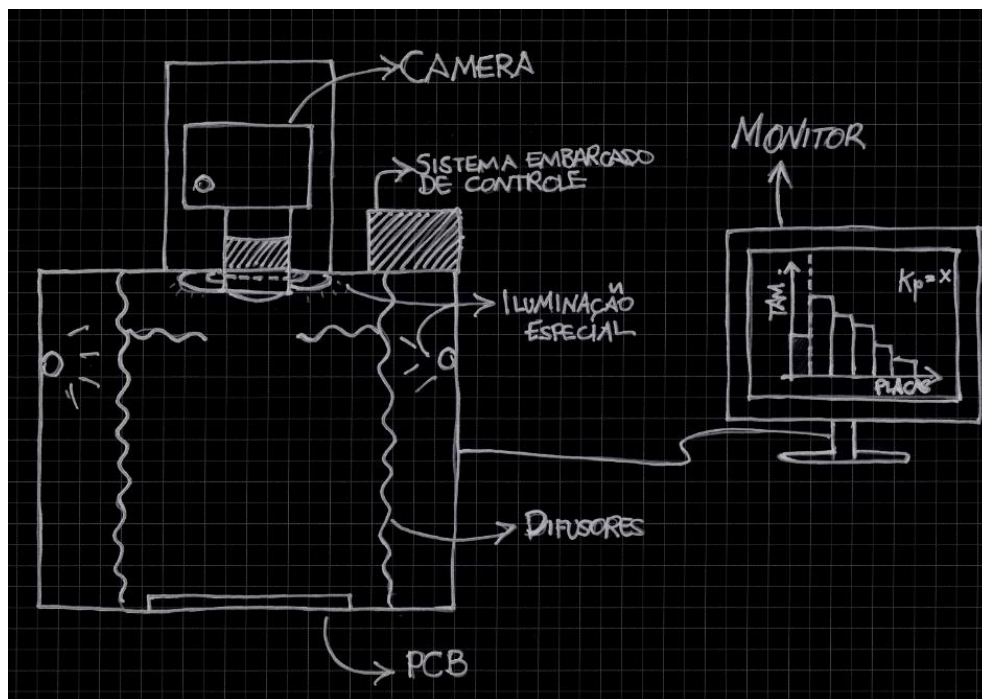


Figura 25 - Desenho do sistema de aquisição de imagens
Fonte: Autoria própria

O trabalho de marcenaria foi todo feito pelos integrantes do grupo, diminuindo o custo e viabilizando a adaptação incremental do protótipo conforme o andamento do projeto. A Figura 26(a) apresenta uma fotografia do sistema final de aquisição de imagens construído, e a Figura 26(b) um detalhe de como a câmera é fixada no sistema.



(a)



(b)

Figura 26 - Sistema de Aquisição de Imagens
Fonte: Autoria Própria

3.3.4. Escolha da Câmera Fotográfica

Uma das especificações do projeto estabelecia que ele fosse flexível e facilmente adaptável à realidade de indústrias e ambientes diferentes. Por isso, se pensou desde o princípio em construir um sistema que pudesse ser utilizado com diversas câmeras fotográficas comerciais. Desta forma, por causa da maneira com que a comunicação entre o sistema embarcado e a câmera foi pensada, a escolha da câmera empregada no protótipo não era de vital importância. Por outro lado, seria interessante que a câmera utilizada fosse boa o suficiente para facilitar ao máximo o projeto do algoritmo, para impedir que, por conta de complicações derivadas da aquisição da imagem, a etapa de processamento de imagens do sistema fosse prejudicada.

A câmera utilizada no protótipo foi a Canon T2i, modelo DSLR de lente intercambiável, com sensor CMOS de 18 megapixels, lançada em 2010 pela Canon (Figura 27) (Canon, Inc., 2012).



Figura 27 - Canon EOS T2i
Fonte: <http://www.canon.com>

Uma vantagem desta câmera é a possibilidade de se utilizar diversas lentes diferentes para a aquisição da imagem, o que aumenta ainda mais a flexibilidade do sistema. Na Figura 28 se encontram duas fotos tiradas da mesma placa, com a mesma câmera, utilizando o mesmo sistema de iluminação construído para o projeto, mas com duas lentes diferentes. A Figura 28 (b) possui ampliação de 5x e foi tirada com uma lente macro profissional da Canon modelo MP-E 65mm f/2.8; a Figura 28 (a) foi tirada com a lente EF 50mm f/1.8, a lente mais barata a venda pela Canon.

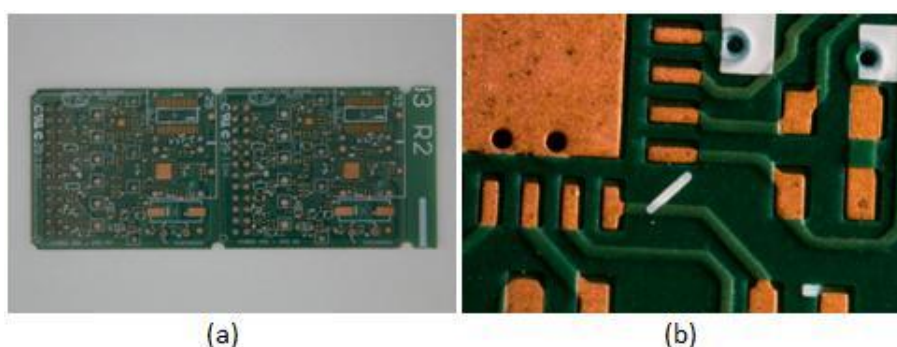


Figura 28 - Duas fotografias tiradas com o sistema criado, da mesma placa, só que em (a) com a lente EF 50mm f/1.8 e em (b) com a lente macro profissional MP-E 65mm f/2.8
Fonte: Autoria própria

3.3.5. Análise do desempenho do sistema

Para verificar os resultados do sistema construído, foram tiradas fotografias com diferentes esquemas de iluminação. Posteriormente, essas fotografias, sem

nenhum pré-processamento, foram binarizadas pelo método de Otsu. A partir do resultado desta análise se pode perceber claramente a importância do controle da iluminação na simplificação do algoritmo necessário para segmentar as trilhas de uma PCI.

A Figura 29 contém os resultados da análise. A figura (a) apresenta a fotografia tirada com iluminação direta (luz ao redor da lente) e com difusores, e o resultado da binarização é mostrado em (b).

A figura (c) foi tirada com iluminação lateral e com difusores. Claramente foi a fotografia que apresentou menor contraste entre as regiões de trilha e de fundo. O resultado da binarização por Otsu, (d), não é aproveitável, e para conseguir segmentar este tipo de imagem seria necessário um algoritmo mais sofisticado. Além disso, por conta da iluminação não uniforme vinda das laterais da foto, a borda da imagem ficou mais escura que o centro, o que pode ser notado com um pouco de dificuldade na fotografia original, mas fica bem evidente na imagem binarizada, em que a borda direita é mais escura.

A fotografia (e) foi tirada com iluminação frontal, só que sem o difusor. O resultado da binarização foi bem superior ao da fotografia com iluminação lateral, mas ainda assim, devido ao reflexo da lâmpada na placa, existem regiões muito claras que prejudicaram a binarização. Neste caso específico, o canto superior esquerdo da imagem não foi corretamente segmentado (como pode ser visto em (f)).

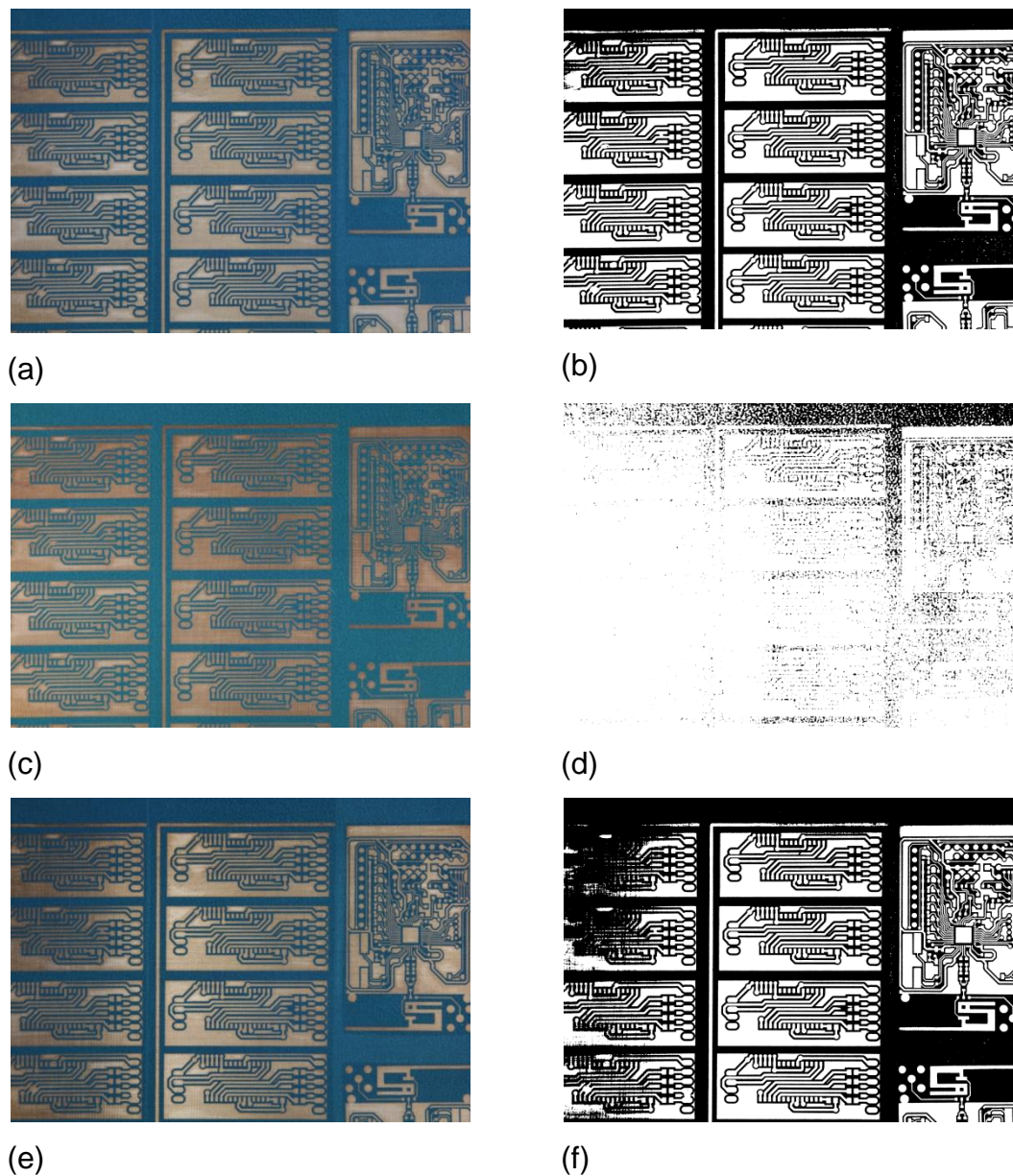


Figura 29 - Resultados da análise do sistema de aquisição de imagens
Fonte: Autoria própria

Conclui-se, portanto, a partir da análise feita, que as fotografias obtidas com o sistema de aquisição de imagens concebido para o projeto são altamente propícias para a aplicação em questão. Pode-se afirmar, também, que o esquema de difusores utilizado - panos brancos, com malha fina, à frente das lâmpadas - foi eficiente em distribuir a luz mais uniformemente pela placa; e que a posição da fonte luminosa, ao redor da lente, se mostrou muito mais favorável que as outras possibilidades testadas.

3.4. SOFTWARE EMBARCADO

Este capítulo apresenta a explicação do software desenvolvido para a interface do usuário com o algoritmo de reconhecimento de falhas, e o sistema de comunicação com a câmera fotográfica. Detalhes sobre a plataforma computacional utilizada (*BeagleBoard XM* (BeagleBoard.org, 2010; Trygve, 1979)) e o motivo pelo qual ela foi escolhida podem ser encontrados no tópico 3.5.

Um diagrama em blocos do sistema como um todo pode ser visto na Figura 30. Os itens subsequentes deste capítulo explicarão em detalhes cada um dos blocos do diagrama.

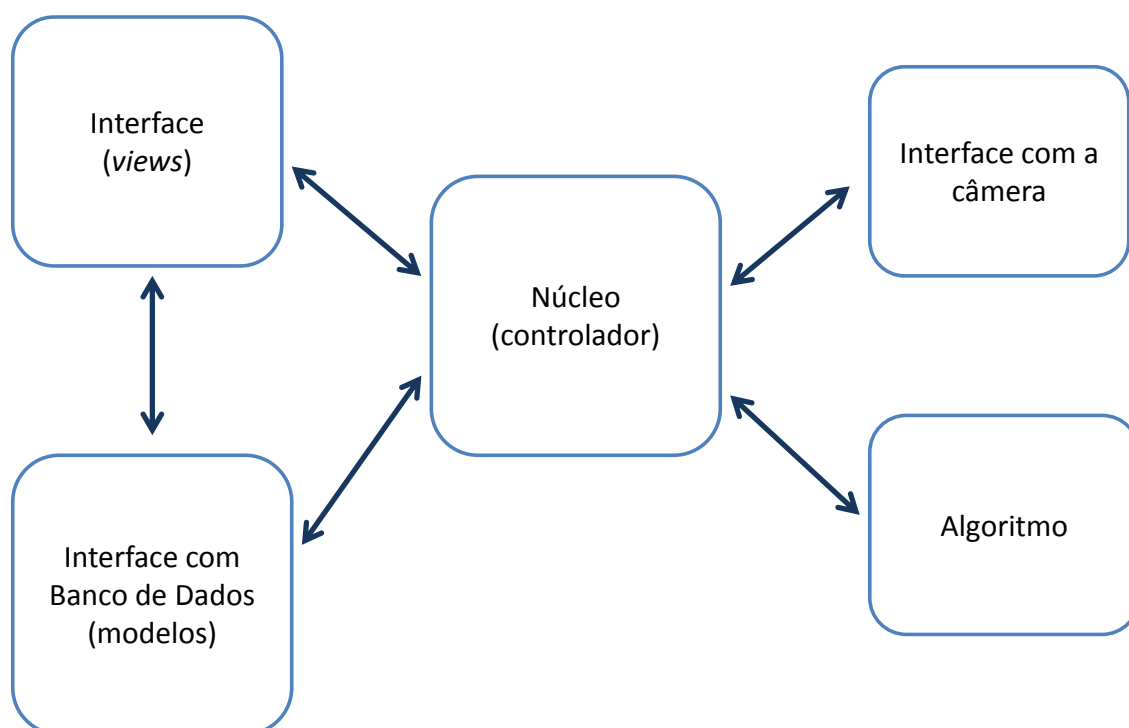


Figura 30 - Diagrama em Blocos do Software Embarcado
Fonte: Autoria própria

Foi utilizado o padrão de software MVC (*model-view-controller*), descrito em detalhes em (Trygve, 1979), para o controle da interface com o usuário. O controlador se relaciona diretamente com o algoritmo de detecção de falhas e com o sistema de comunicação com a câmera, criando uma camada de abstração entre a parte de interação com o usuário e a lógica do software.

O controlador foi implementado utilizando o framework de Máquina de Estados Finitas do Qt (Nokia Corporation, 2011). O banco de dados utilizado foi o *mysql*, em conjunto com a API de interação com o *mysql* do Qt. A interface com a câmera fez uso da biblioteca *opensource* libgphoto2 (libgphoto2 Development Team, 2007), e o algoritmo, conforme descrito na seção 3.2, foi programado com auxílio da biblioteca OpenCV.

A versão final do código possui mais de 6 mil linhas de código, e por isso, neste relatório, optou-se por resumir as partes relevantes do funcionamento do programa ao invés de detalhar e entrar nas minúcias de implementação do código.

3.4.1. A Biblioteca Qt

Todo o software foi desenvolvido utilizando a biblioteca Qt (Nokia Corporation, 2012) e as ferramentas de desenvolvimento disponíveis para ela.

O Qt (pronuncia-se 'cute', como a palavra em inglês para "bonitinho") é uma biblioteca C++ desenvolvida na década de 90 pela companhia norueguesa Trolltech. Ela se popularizou quando foi escolhida como plataforma de desenvolvimento básica para o projeto KDE (um conjunto de programas gráficos sob licença GNU, para serem distribuídos junto de versões do Linux), e acabou sendo comprada pela Nokia em 2008 (Nokia Corporation, 2008).

A característica distintiva do Qt, a que motivou inclusive a criação da biblioteca, é a capacidade fornecida a certos objetos de conhecerem, programaticamente, o nome e a assinatura de algumas de suas funções e variáveis. Dessa forma, pode-se passar como parâmetro para uma classe uma string que contém o nome de uma função, por exemplo, para que em certas ocasiões especiais a função com aquele nome seja executada.

Esse sistema é denominado Meta-Object System, e é implementado totalmente em C++ padrão. Essa adição do Qt alia às vantagens do C++ (robustez, flexibilidade e performance) certas características de linguagens de programação modernas, como Ruby e Python (principalmente o suporte à metaprogramação, além de implementações diferenciadas de estruturas de dados comuns, por exemplo).

O grande motivador do desenvolvimento desse sistema foram as facilidades que ele traz para o paradigma de programação assíncrono, orientado a eventos, tipicamente encontrado em programação de interação com o usuário. Com o Meta-Object System, o Qt oferece um sistema eficiente e fácil de utilizar de propagação de eventos (signals and slots system, em inglês), que pode ser utilizado, por exemplo, para implementar facilmente padrões de software complicados tais como um Observer, padrão descrito em (Gamma, et al., 1994)

Além disso, o Qt fornece funções para desenho e geração de GUIs independentes de plataforma, o que facilita enormemente a portabilidade de código entre sistemas Windows, Mac e Linux.

3.4.2. Núcleo do Sistema (State Machine)

O núcleo do sistema fez o papel de controlador, sob o ponto de vista do padrão *Model-View-Controller*.

A abordagem do problema com máquina de estados finita foi escolhida pela facilidade do modelamento do comportamento do programa, pela facilidade de se poder discutir e entender os problemas relativos à implementação antes de se começar a produzir código propriamente dito, e por ser fácil modificar o modelo caso ocorra algum problema. É, portanto, uma abordagem de modelamento facilmente modificável, descritiva do comportamento efetivo do software, e com a qual todos os integrantes do grupo possuíam familiaridade. Dito de maneira mais simples: era mais fácil discutir sobre o projeto do software baseado nos diagramas de estado do que vendo código efetivo em C++.

O framework de *Finite State Machines* do Qt permitiu que a máquina projetada fosse facilmente traduzida em código C++. Para a biblioteca, cada estado pode ser representado como uma classe, derivada de um tipo básico denominado `QState`. As funções *onEntry* e *onExit*, são funções virtuais definidas para o tipo básico, e para adicionar comportamento específico para alguma delas basta reescrevê-las na classe derivada. A Figura 31 apresenta um *snippet* de como um estado foi implementado com as funções do Qt.

```

1. // stinitial.h
2. #ifndef STINITIAL_H
3. #define STINITIAL_H
4. #include "genericstate.h"
5.
6. class StInitial : public GenericState
7. {
8.     Q_OBJECT
9. public:
10.     explicit StInitial(CoreStates *st, Core *c, QState *ps = 0, MainWindow *w = 0);
11.
12.     virtual void createTransitions();
13. signals:
14.     void analysisCreated();
15.     void analysisLoaded();
16. public slots:
17.     void createNewAnalysis(void);
18.     void openAnalysisPopUp(void);
19. private:
20.     virtual void onEntry(QEvent* event);
21. };
22.
23. #endif // STINITIAL_H
24.
25. // stinitial.cpp
26. #include "stinitial.h"
27.
28. #include "InfoBar/ibinitial.h"
29.
30. StInitial::StInitial(CoreStates *st, Core *c, QState *ps, MainWindow *w):
31.     GenericState(st, c, ps, w)
32. {
33. }
34.
35. void StInitial::onEntry(QEvent *event)
36. {
37.     win->changeInfoBar(tibvInitial);
38.     win->changeActionBar(tabvInitial);
39.
40.     connect(win->ibInitial, SIGNAL(registerBoard()), this, SLOT(createNewAnalysis()));
41.     connect(win->ibInitial, SIGNAL(startAnalysis()), this, SLOT(openAnalysisPopUp()));
42. }
43.
44. void StInitial::createTransitions()
45. {
46.     this->addTransition(this, SIGNAL(analysisCreated()), states->RegisterBoard);
47.     this->addTransition(this, SIGNAL(analysisLoaded()), states->PerformAnalysisAdjustPosition);
48. }
49.
50. void StInitial::createNewAnalysis (void)
51. {
52.     core->createNewAnalysis();
53.     emit(analysisCreated());
54. }
55.
56. void StInitial::openAnalysisPopUp()
57. {
58.     OpenAnalysisDialog d(core, win);
59.     d.exec();
60.     d.result();
61.     emit(analysisLoaded());
62. }

```

Figura 31 - Exemplo de implementação de um estado com o Qt State Machine Framework

3.4.3. Modelos (interação com Banco de Dados)

Para representar em código os modelos conceituais utilizados no programa, foram utilizadas classes adaptadoras para as tabelas do banco de dados. Elas possuem funções para salvar e ler dados da tabela relativa a ela, fazem a validação

dos dados salvos para garantir que o banco de dados nunca será posto em um estado inválido, e fornecem funções que permitem às *views* exibirem dados contidos nos modelos.

A estrutura do banco de dados utilizada no software está representada na Figura 32.

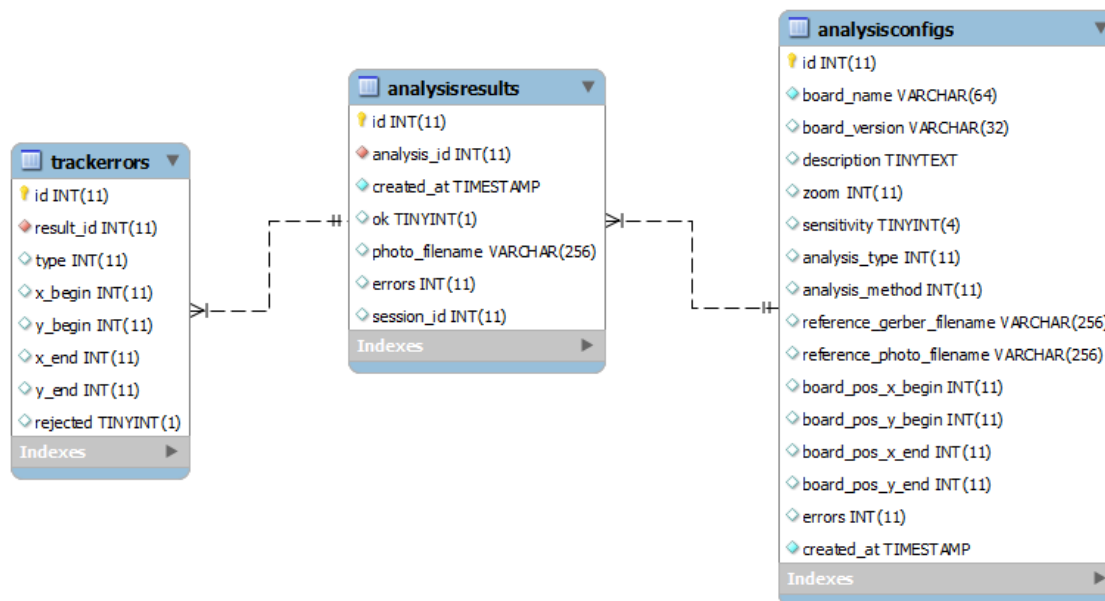


Figura 32 - Estrutura do banco de dados
Fonte: Autoria própria

3.4.3.1. Modelo conceitual do Software

Como pode ser visto da estrutura do banco de dados criada, o software foi construído em torno do conceito de “análise”. Uma análise é o ato de verificar se uma placa de circuito impresso possui um ou mais tipos de erros. A configuração da análise é a estrutura lógica que agrupa todas as características de um tipo específico de análise. O resultado da análise é o resultado retornado pelo algoritmo; esta tabela guarda informações sobre a posição e o tipo dos erros encontrados.

3.4.4. Views (Interface com o Usuário)

Para facilitar a programação e tornar a lógica da interface com o usuário independente da lógica do funcionamento do programa, mantendo a filosofia do

padrão *Model-View-Controller*, as telas do programa foram escritas utilizando o padrão XML do Qt . Esses arquivos XML são convertidos automaticamente em código C++ por uma ferramenta do Qt, e são compilados junto do programa.

3.4.5. Comunicação com a Câmera

A comunicação com a câmera foi desenvolvida utilizando as funções da biblioteca *opensource libgphoto2* (libgphoto2 Development Team, 2007). Esta é escrita em C, e possui funções para listar câmeras conectadas ao computador via USB, estabelecer conexão com a câmera, manipular, salvar e fazer download de arquivos presentes na memória da câmera, de enviar comando remoto para captura de imagens e controlar alguns parâmetros da máquina fotográfica.

A libgphoto2 é utilizada em diversos programas do pacote Gnome, um sistema de interface gráfica GNU/Linux, sendo o Gimp, talvez, o mais conhecido destes.

3.4.6. GUI

Como o sistema seria utilizado, primordialmente, para fazer análises de placas de circuito impressos por um usuário que potencialmente não teria intimidade com o sistema, a interface teria de ser simples e demandar pouco ou quase nenhum conhecimento prévio do usuário sobre o funcionamento do programa. Por este motivo, todos os procedimentos de registro ou cadastramento de informações no programa são baseados no modelo de *wizard*, no qual existe uma sequência de telas simples que explicam parte do processo para o usuário e recebem poucas informações do usuário. Isso não sobrecarrega o usuário na hora de cadastrar uma análise, por exemplo, com uma tela única a ser preenchida com informações como “tipo da análise”, “exposição da câmera” e “posição da placa”. Além disso, o sistema de wizard previne notadamente erros de configuração, porque o usuário é levado a pensar individualmente em cada uma das propriedades do que está cadastrando, podendo voltar e modificar informações inseridas de maneira errada em qualquer ponto do processo.

A Figura 33 mostra um screenshot do programa..

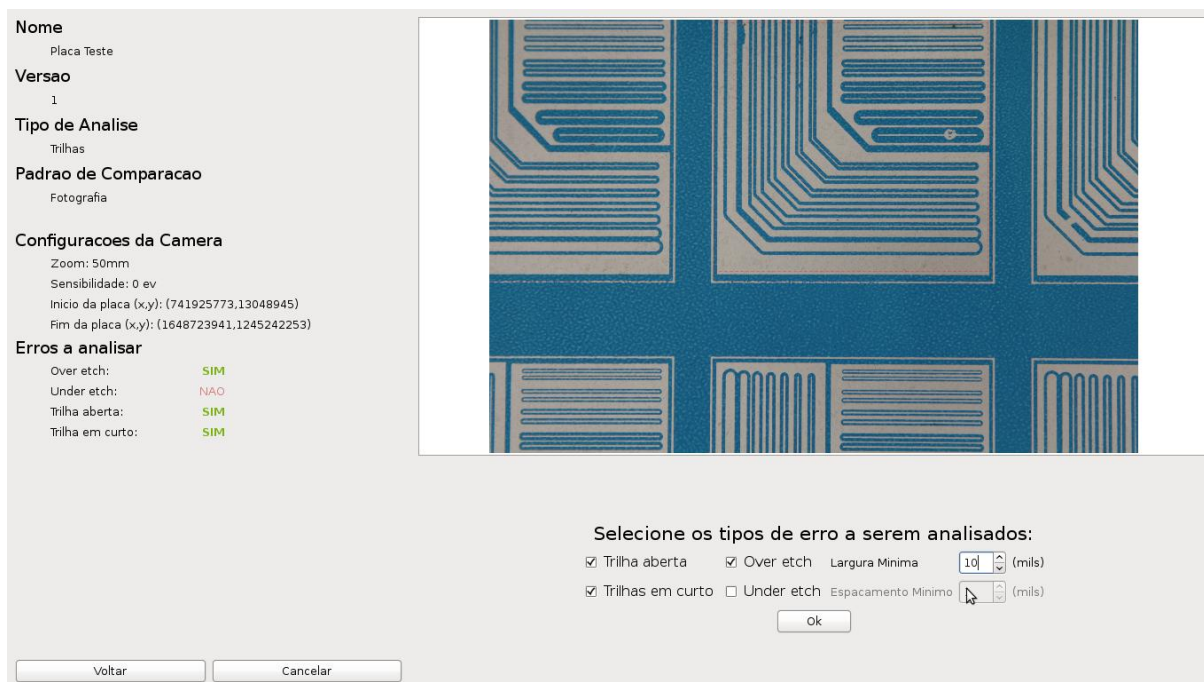


Figura 33 - Screenshot do aplicativo desenvolvido
Fonte: Autoria Própria

3.5. PLATAFORMA EMBARCADA

3.5.1. Estudos Iniciais da Plataforma

Seguindo a escolha de apresentar um equipamento compacto, para uso tanto em laboratórios quanto em ambientes industriais, foi decidido o uso de uma plataforma embarcada para o sistema, provendo assim a solução completa tanto para o software de inspeção quanto para o banco de dados utilizado para guardar os dados das análises. Dessa forma, não é necessário o uso de um computador extra para o funcionamento do equipamento, permitindo otimização de custos e tamanho.

Dentre as várias alternativas de sistemas embarcados existentes no mercado, como os baseados em microcontroladores de alto desempenho, DSPs, processadores x86, foram focadas duas soluções: uso de um PC de baixo custo ou um sistema baseado em um SoC ARM (SoC, do inglês *System on a Chip*). Avaliando os sistemas embarcados pelas suas características de processamento, tamanho e custo, foi possível obter a Tabela 6:

	PC de baixo custo	SoC ARM
Processamento	Consideravelmente maior, mas pode se tornar um recurso em excesso	Semelhante aos processadores x86 de baixo custo, levando em conta os SoC baseados em Cortex A8. Suficiente para a aplicação.
Tamanho	Consideravelmente maior, levando em conta fonte, memórias, placa-mãe.	Reduzido a uma placa, de até 15x15cm e uma fonte de alimentação pequena.
Custo	Pequeno	Pequeno

Tabela 6 - Tabela comparativa de sistemas de processamento
Fonte: Autoria própria

Apesar da capacidade de processamento da plataforma baseada em PC ser consideravelmente maior, os processadores ARM mais recentes são suficientes para a aplicação. Somando-se o tamanho reduzido da plataforma baseada em ARM, e seu custo praticamente equivalente ao da baseada em PC, essa foi a solução escolhida.

3.5.1.1. Escolha da plataforma baseada em SoC ARM

Dentre as várias alternativas de kits de desenvolvimento pesquisadas, a mais interessante encontrada foi a Beagleboard xM (Figura 34), por uma série de motivos:

- Plataforma bastante popular: maior número de desenvolvedores utilizando a plataforma, e conseqüentemente maior a documentação extra-oficial;
- Processador OMAP3: baseado no processador ARM Cortex A8 operando a 1 Ghz;
- DSP integrado ao processador: melhor desempenho ao executar tarefas de processamento digital de imagens;
- Relativamente bem suportada em relação a exemplos, compiladores, etc.
- Melhor suporte a linux embarcado;
- Preço: dos vários kits de desenvolvimento disponíveis, a placa é um dos kits mais baratos.

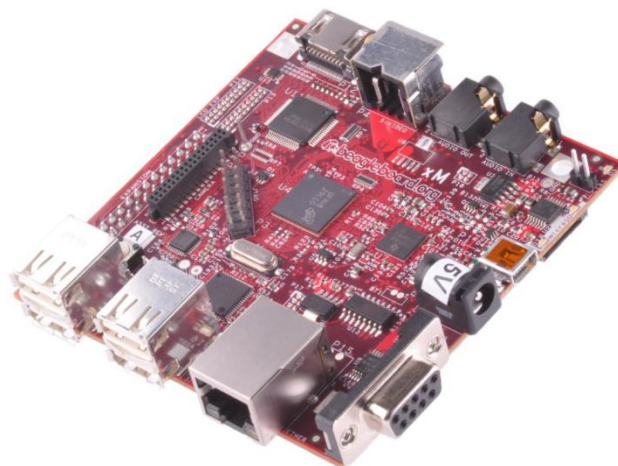


Figura 34 - Beagleboard xM

Fonte: BeagleBoard-xM Rev C System Reference Manual, Revision 1.0

O uso de um sistema operacional foi um dos fatores decisivos para a escolha da placa, já que ele provê acesso as principais interfaces de entrada e saída, como teclado, monitor e para a câmera por meio do USB, como também uma camada de abstração para o desenvolvimento do software em si.

3.5.1.2. SO embarcado

O sistema operacional Linux foi escolhido, na distribuição Angstrom. Foram avaliadas algumas distribuições, como Yocto, Angstrom e Ubuntu em sua versão embarcada, via Linaro (Yocto Project, 2012), (Angstrom Distribution, 2012), (Linaro, 2012).

O projeto Linaro disponibiliza distribuições prontas para uso, bastante robustas, mas com poucas opções de customização. Já as distribuições Yocto e Angstrom permitem a construção de distribuições totalmente customizadas pelo usuário. O Yocto é um projeto que provê ferramentas, modelos e métodos para permitir o usuário criar e compilar a sua própria distribuição Linux, independente da arquitetura utilizada. Já o projeto Angstrom é uma distribuição de fato, que suporta a BeagleBoard nativamente, entre outros hardwares. A grande vantagem da utilização do Angstrom é o seu “online-builder” (Figura 35), uma ferramenta que permite compilar e baixar de uma imagem customizada via uma interface web no site da distribuição. Em outras distribuições, só é possível baixar imagens padronizadas; para obter uma imagem customizada é preciso compilá-la por completo.

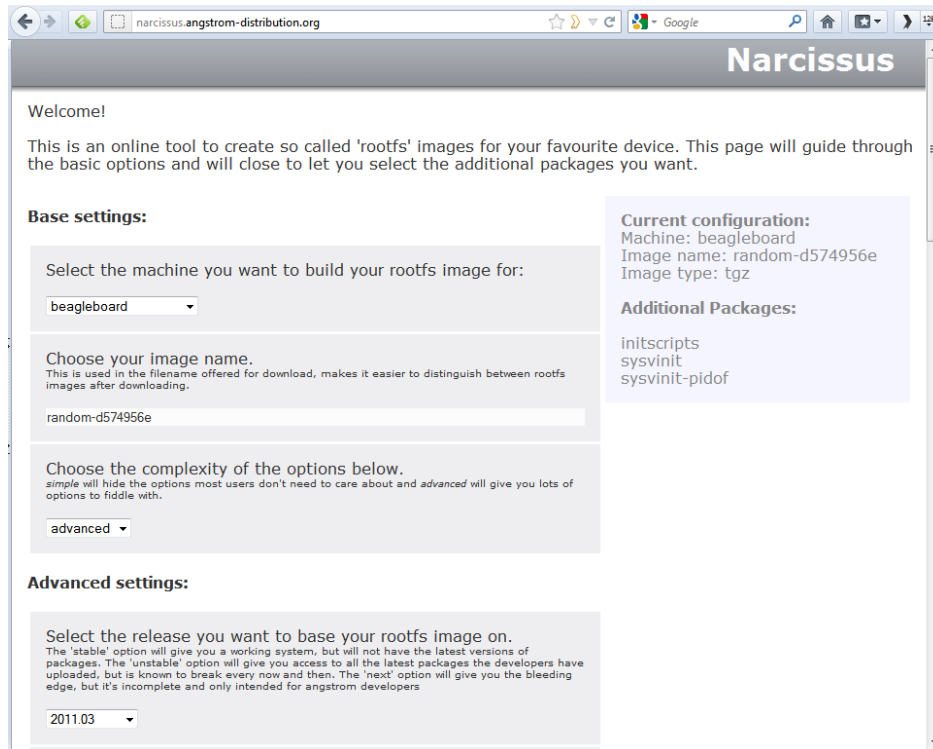


Figura 35 - Screenshot do site Narcissus
Fonte: narcissus.angstrom-distribution.org

Com o “online-builder”, é possível também baixar a toolchain – conjunto de ferramentas necessárias para compilar programas para a distribuição.

3.5.2. Implementação da Toolchain

Para compilar o software de interface que rodará na plataforma embarcada, existem duas opções: compilar diretamente no dispositivo, ou compilar em um computador e carregá-lo no dispositivo.

Ao compilar diretamente no dispositivo, além do processo de compilação ser mais demorado pela menor capacidade de processamento, ainda há o problema de limitação de memória RAM. Essa menor disponibilidade de memória impede a compilação de programas que utilizem grandes bibliotecas – exatamente o caso do Qt. Por isso, foi necessário compilar o software de inspeção separadamente em um computador.

Para gerar programas para arquitetura ARM a partir de outra arquitetura de processador, é preciso utilizar compiladores cruzados. Compiladores cruzados são tipos de compiladores que geram códigos para arquiteturas diferentes daquela em que o compilador está sendo executado.

Algumas distribuições para sistemas embarcados proveem conjuntos de ferramentas para compilar os programas em arquitetura x86 – as chamadas *toolchains*. Incluídos nessas ferramentas estão os compiladores cruzados, *linkers* e outras ferramentas para configuração do ambiente.

A escolha do Angstrom como distribuição Linux permitiu a utilização de uma *toolchain* pré-compilada, cujo download pode ser feito juntamente com a imagem customizada a ser utilizada no sistema embarcado. Dessa forma, há um conjunto de imagem e *toolchain* que precisa ser usado em conjunto para que o programa seja gerado e rode corretamente no dispositivo.

Existem, porém, muitos problemas na utilização dessa configuração, e alguns foram encontrados durante a implementação do software, o que acabou gerando alguns atrasos no cronograma estabelecido.

Um grande problema da plataforma escolhida foi a organização da documentação da plataforma. Como a BeagleBoard é uma placa mantida pela comunidade, toda a sua documentação é realizada por voluntários interessados no projeto, e portanto não existe uma documentação oficial do fabricante. Apesar da existência de fóruns oficiais, as fontes de documentação não são padronizadas e nem sempre estão atualizadas conforme a versão mais atual da placa. Existem ainda muitos sites que disponibilizam documentações relacionadas a plataforma, porém é difícil distinguir qual provê a informação correta ou a mais atual.

Além disso, é preciso uma atenção especial na utilização da *toolchain*, que deve ser devidamente configurada. Como a *toolchain* usa um variante do compilador GCC, que também é disponibilizado com grande parte das distribuições Linux – inclusive a que está sendo usada para gerar o software de interface –, é bastante fácil confundir as fontes de bibliotecas dos compiladores, gerando conflitos ao compilar o software.

Em contrapartida, a utilização da *toolchain* facilitou o uso de bibliotecas *OpenSource*, que foram de grande ajuda para o projeto. As bibliotecas Gphoto, para a comunicação com a câmera, e a Gerbv, para leitura de arquivos gerber, não são disponibilizadas com a *toolchain*, e precisaram ser incluídas.

4. GESTÃO

4.1.ESCOPO

Escopo original, conforme proposta:

“A análise de falhas de fabricação de placas de circuito impresso será realizada com base em fotos obtidas com uma câmera digital de alta resolução, que serão transmitidas ao software de reconhecimento de falhas executado sobre um sistema operacional embarcado em uma placa de desenvolvimento.

O software embarcado terá uma interface gráfica que permitirá ao usuário configurar os tipos de falha a serem analisadas, alterar os parâmetros de reconhecimento de falha, receber os resultados da análise, e ainda realizar alguns ajustes na câmera, caso seja necessário. Sua interface se dará por meio de monitor, mouse e teclado ligados ao sistema embarcado.

O software de reconhecimento de falhas fará a busca exaustiva das falhas pré-definidas pelo usuário, e será construído sobre algoritmos de processamento de imagens programados para seu reconhecimento. Esse software será capaz de gerar relatórios de erros baseados nas falhas encontradas nas placas atuais, assim como será capaz de gerar estatísticas sobre o processo analisado, como as falhas mais comuns em certo lote de placas ou uma comparação com processos anteriores. As “fotos das placas com erro, com destaque nos erros encontrados, também serão armazenadas.”

Considerando a definição do escopo do projeto apresentado na proposta, não houve alterações. Todos os requisitos do projeto foram atendidos, sem a necessidade de alterações para adequação à forma de implementação escolhida.

4.2.CRONOGRAMA

O cronograma sofreu atrasos consideráveis em relação ao planejado na proposta. Houve atrasos consideráveis na implementação do ambiente de desenvolvimento do software embarcado, na implementação do software de interface e nos testes e implementação do algoritmo de inspeção de erros.

No geral, os atrasos ocorreram por dificuldades não consideradas, como no caso do ambiente de desenvolvimento da plataforma embarcada. A falta de experiência com distribuições Linux em sistemas embarcados, aliada a documentação desorganizada da plataforma levaram um atraso do programado de duas semanas para mais de um mês e meio. Ainda, um tempo maior para o

aprendizado do *framework* para a interface do usuário também levou mais tempo que o previsto.

Porém vale considerar que todos os atrasos foram convertidos em aprendizado, pois foi necessário aprender soluções diferentes das previstas para a resolução dos problemas. Pode-se citar o aprendizado de compiladores, linkers, linux, *framework* de interface com o usuário e algumas técnicas de programação como os conhecimentos extras adquiridos mais relevantes.

Abaixo está apresentado o quadro aproximado do cronograma seguido. Observa-se que as datas são aproximadas, já que com os atrasos não houve um acompanhamento mais próximo das datas das tarefas.

Quadro 1 - Cronograma expandido

Núm.	Tarefa	Data Início	Data Término	Responsáveis
1	Análise de viabilidade	02/09/11	30/09/11	André, Marcos, Mariana
2	Hardware	30/09/11	01/12/11	André, Marcos, Mariana
2.1	Escolha do hardware	30/09/11	06/10/11	André, Marcos, Mariana
2.2	Compra da placa de desenvolvimento	07/10/11	08/12/11	André, Marcos, Mariana
3	Projeto da aplicação	07/10/11	26/01/12	André, Marcos, Mariana
3.1	Listar os possíveis erros	07/10/11	17/10/11	André, Marcos, Mariana
3.2	Revisão bibliográfica	18/10/11	26/12/11	Mariana
3.3	Teste e seleção dos algoritmos que serão utilizados.	07/12/11	26/01/12	Mariana
4	Implementação do algoritmo de inspeção	27/01/12	27/04/12	Mariana
4.1	Implementação do algoritmo de inspeção no MatLab	06/01/12	22/03/12	Mariana
4.2	Implementação do algoritmo de inspeção no OpenCV	23/03/12	27/04/12	Mariana
5	Implementação da interface	26/03/12	10/05/12	André, Marcos
5.1	Implementação da interface em um ambiente Linux	26/03/12	02/05/12	André, Marcos
5.2	Preparação e implementação do da plataforma embarcada	20/01/12	26/03/12	Marcos
5.3	Implementação da interface no sistema embarcado	26/03/12	10/05/12	André, Marcos
6	Testes	01/06/12	07/06/12	André, Marcos, Mariana
6.1	Teste da aplicação	01/06/12	07/06/12	André, Marcos, Mariana
6.2	Teste da interface	01/06/12	07/06/12	André, Marcos, Mariana

7	Correções e otimizações	08/06/12	15/06/12	André, Marcos, Mariana
7.1	Correções e otimização da aplicação	08/06/12	15/06/12	André, Marcos, Mariana
7.2	Correções e otimização da interface	08/06/12	15/06/12	André, Marcos, Mariana
8	Redação da proposta de projeto	21/10/11	16/11/11	André, Marcos, Mariana
8.1	Redação da proposta inicial de projeto	21/10/11	09/11/11	Marcos, Mariana, André
8.2	Revisão e entrega da proposta	10/11/11	16/11/11	André, Marcos, Mariana
9	Redação do trabalho de conclusão	02/05/11	06/06/12	Marcos, Mariana, André
9.1	Redação do trabalho de conclusão	02/05/11	30/05/12	Marcos, André, Mariana
9.2	Revisão do trabalho de conclusão	01/06/12	06/06/12	André, Mariana, Marcos

Fonte: autoria própria

4.3.CUSTOS

Os custos referentes aos materiais utilizados no projeto foram divididos em três categorias, referentes a cada parte do projeto: hardware, custos relacionados a eletrônica no projeto; estrutura, referente ao caixote de madeira montado; e iluminação, referente aos custos. O custo de cada material e o total gasto no projeto estão relacionados na tabela abaixo:

Descrição	Custo
BeagleBoard xM	R\$ 369,00
Frete	R\$ 40,00
Teclado	R\$ 39,90
Adaptador HDMI	R\$ 19,00
Total hardware	R\$ 467,90
Madeira	R\$ 15,00
Cola de madeira	R\$ 3,00
Parafuso	R\$ 6,00
Cola	R\$ 6,00
Sapatas	R\$ 5,72
Fita	R\$ 2,34
Papel contact	R\$ 4,00
Borracha	R\$ 27,00
Lixas	R\$ 4,00
Total Estrutura	R\$ 73,06
Lâmpadas, reator e suporte	R\$ 25,00
Interruptor	R\$ 2,10
Plug de tomada	R\$ 2,00
Fios	R\$ 18,00

Fita e vareta	R\$ 3,30
Fita dupla face	R\$ 6,90
Total Iluminação	R\$ 57,30

TOTAL GERAL	R\$ 598,26
--------------------	-------------------

Tabela 7 - Custos do projeto
Fonte: autoria própria

4.4. RISCOS

Tomando o quadro de análise de riscos enviada junto a proposta do projeto, pode-se afirmar que os riscos foram corretamente avaliados, já que não ocorreram problemas além dos relatados. Dentre os riscos relatados, só foram verificados atraso no desenvolvimento e um não domínio parcial nas ferramentas de desenvolvimento.

Quadro 2 - Riscos do projeto

Grau ¹	Descrição	Efeito	Impacto	Probabilidade	Ação
3,6 (Alto)	Problemas com importação de componentes	Atraso para embarcar o sistema operacional	8	0,4	Mitigar: Desenvolver os <i>softwares</i> e utilizar plataforma PC para testes.
3,6 (Alto)	Atraso no desenvolvimento	Atraso e alteração no cronograma	9	0,4	Eliminar: Deslocar membro da equipe para terminar a tarefa
1,8 (Médio)	Algoritmo escolhido não atende os requisitos	Atraso na conclusão do <i>software</i> de reconhecimento de falhas, e alteração no cronograma	9	0,2	Eliminar: Deslocar membro da equipe para o redesign do <i>software</i> de reconhecimento.
1,4 (Médio)	Não dominar as ferramentas de desenvolvimento	Atraso e alteração no cronograma	7	0,2	Eliminar: Deslocar membro da equipe para ajudar na tarefa
0,7 (Baixo)	Orçamento não ser aceito	Atraso no cronograma	7	0,1	Conviver: Realizar orçamentos de componentes mais baratos
0,6 (Baixo)	Evasão de membro de equipe	Sobrecarga nos outros membros e atraso no cronograma	6	0,1	Conviver: Não tomar nenhuma ação

¹ Grau = Impacto x Probabilidade.

Fonte: autoria própria

4.5. CONSIDERAÇÕES

Apesar do projeto não seguir corretamente o cronograma estipulado devido aos atrasos descritos, foi possível manter a entrega no semestre previsto. Os gastos do projeto não foram além do previsto, e não houve risco não avaliado no quadro de análise de riscos.

5. PLANO DE NEGÓCIOS

5.1.SUMÁRIO EXECUTIVO

A empresa Valar, objeto desse plano de negócios, será uma empresa atuante no segmento de produção de equipamentos de inspeção industrial. Seu primeiro produto, um inspetor óptico automatizado de falhas de produção em placas de circuito impressos, já está em fase final de desenvolvimento e já há inclusive um protótipo funcional.

A Valar, cuja sede será Curitiba, será formada por três Engenheiros Eletricistas formados da Universidade Tecnológica Federal do Paraná, com experiência na área de foco de desenvolvimento da empresa: visão computacional.

O Abutre, o primeiro produto da Valar, será capaz de detectar as falhas mais comuns no processo de fabricação de placas de circuito impresso. Apesar do ramo de fabricação dessas placas se encontrar em declínio, o fato de ser composto em sua maioria por pequena e médias empresas permitirá um contato com o gestor ou diretor dessas empresas, dos quais pretende-se ganhar a confiança.

A estratégia de mercado da Valar será adquirir confiança do mercado através de seu primeiro produto, que, apesar de não ser capaz de manter a empresa financeiramente viável, será de vital importância para alavancar as vendas dos próximos produtos que já estão sendo concebidos, também na área de inspeção óptica automatizada de indústrias.

5.2. DEFINIÇÃO DO NEGÓCIO



**Figura 36- Logomarca da empresa Valar.
Fonte: Autoria Própria**

Este plano de negócios analisa a proposta de criação da empresa Valar, no segmento de produção de equipamentos de inspeção industrial. O primeiro produto da empresa, o Abutre, se encontra no estágio final de desenvolvimento – já foi produzido um protótipo funcional - e se trata de um inspetor óptico automatizado de falhas de produção em placas de circuito impressos.

A Valar terá sede em Curitiba, PR. Os três sócios fundadores da empresa, uma mulher e dois homens, são Engenheiros Eletricistas formados em uma mesma turma pela Universidade Tecnológica Federal do Paraná, e possuem conjuntamente as seguintes experiências na área de visão computacional:

- Uma sócia trabalhou com pesquisas na área de visão com múltiplas câmeras no *CEA – Commissariat à l'énergie atomique et aux énergies alternatives*, em Paris, França.
- Um sócio trabalhou em um laboratório de pesquisas em óptica aplicada à biomedicina em Berlin, Alemanha.
- Um sócio trabalhou no desenvolvimento de um microscópio de tunelamento também em Berlin, Alemanha.

A empresa será fundada baseada nos seguintes princípios:

Visão: ser aquele que tudo vê.

Criar e inventar métodos, produtos e soluções que permitam supervisionar processos industriais, permitindo ao gestor localizar e reconhecer falhas, evitando desperdícios, prejuízos e indisposições com o cliente final.

Missão: encontrar maneiras inovadoras de prospectar e interpretar dados em ambiente industrial, diminuindo o desperdício, localizando falhas e prevenindo erros.

Valores: curiosidade, persistência e vontade de ir além do óbvio.

O negócio principal da empresa será prover um canal confiável entre a gestão e o que acontece de fato no negócio gerido, e, sob esse ponto de vista, o produto inicial da Valar seria apenas um meio inicial de se inserir no mercado, ganhar experiência e fazer um nome. A escolha do segmento de PCIs foi consciente e se deu por dois motivos principais: como se trata de um negócio em declínio no Brasil, composto de empresas geralmente de menor porte, é mais fácil entrar em contato direto com os gestores e donos das indústrias; e para conseguir sobreviver à concorrência do mercado estrangeiro, esse é um ramo que precisa, urgentemente, agregar valor aos produtos que fabrica.

5.3.OBJETIVOS

- Vender 25 soluções de inspeção nos primeiros 2 anos (*market share* de 50%) – um produto por mês até Julho de 2014;
- Desenvolver um novo produto na área de inspeção óptica até o final de 2013;

5.4.PRODUTOS E SERVIÇOS

5.4.1. O Abutre

O Abutre, como o nome já implica, é capaz de, por meio de visão, reconhecer placas de circuito impressos incapazes de serem comercializadas. Os defeitos reconhecidos pelo Abutre são: curto entre trilhas, trilha aberta, trilha pouco corroída (*over etch*), trilha muito corroída (*under etch*).


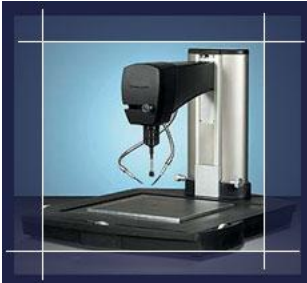
O produto será fisicamente constituído por uma caixa metálica com iluminação especial interna, aonde será colocada a PCI a ser testada. Nessa caixa será afixada uma câmera fotográfica, que pode ser a princípio qualquer câmera comercial da Canon com comunicação USB, que será responsável por adquirir as imagens da PCI. Integrado à caixa haverá um *single-board computer*, com saída para monitor, mouse e teclado, que será responsável pela parte lógica de análise da fotografia adquirida e pela interface com o usuário.

A interface com o aparelho se dará através de um programa que permitirá cadastrar informações de placas a serem analisadas, supervisionar o processo de análise da placa, e verificar os resultados de todas as análises passadas.

5.4.1.1. Personalização do algoritmo

Caso o cliente tenha necessidades especiais de verificação, poderá entrar em contato com a empresa que o caso específico será analisado pela equipe de engenharia da Valar.

5.4.2. Análise Comparativa

Fabricante	Imagem do produto	Características principais
Omron Industrial Automation <u>VT-RNSII</u>		Pequeno, ideal para pequena produção Capaz de fazer a leitura após corrosão, após placement, após reflow Capacidade de identificar problemas de solda.
Orpro Vision <u>INSITE B BENCH TOP AOI</u>		Pequeno. Capaz de fazer verificação apenas post-placement.
Nordson Yestech <u>BX AOI</u>		Pequeno. Capaz de fazer a leitura após corrosão, após placement, após reflow. Capacidade de identificar problemas de solda.
EasyBraid Co. <u>VPI1000</u>		Pequeno Construção simples -> possivelmente mais barato Aparentemente funciona com após placement ou após reflow Aparentemente não é AOI, mas sim uma ferramenta para visualização de solda BGA usando uma “camera” de fibra óptica

Fonte: Autoria própria

O quadro comparativo acima lista as principais características dos produtos concorrentes presentes no mercado.

A pesquisa revelou que, apesar de já existirem no mercado empresas que produzem produtos semelhantes ao da Valar, os possíveis concorrentes encontrados são empresas com sede e fábrica no exterior, o que torna o Abutre vantajoso financeiramente no mercado nacional por conta da carga tributária elevada de importação de maquinário industrial, e pelo apoio técnico personalizado

que a Valar pode dar aos clientes. Isso é uma grande vantagem competitiva principalmente para as pequenas e médias empresas brasileiras.

5.4.3. Tecnologias

As tecnologias empregadas no produto possuem pouco risco de obsolescência no curto e médio prazo, devido ao fato de o principal valor tecnológico presente no Abutre ser o algoritmo desenvolvido pela Valar.

Para que o produto funcione, o necessário é apenas uma plataforma pequena que rode um sistema GNU/Linux. Existem diversas plataformas desse tipo no mercado, e há a possibilidade de fabricação própria dessa placa. Isso diminui o poder de barganha dos fornecedores e aumenta a possibilidade de adaptação do Abutre às necessidades de diversos clientes.

Outra vantagem grande de ter como principal diferencial tecnológico um algoritmo, é que se pode fazer *upgrades* de produtos já vendidos, aumentando a receita e a satisfação do cliente com o produto ao longo do tempo.

Por se tratar de um produto industrial, o tempo de vida do Abutre também foi planejado para girar em torno de uma década.

5.4.4. Produtos e Serviços Futuros

O Abutre é um produto pensado somente para obter um mercado inicial fácil, de maneira rápida, e assim criar um nome e reputação para a Valar, além de servir para adquirir experiência no ramo antes de se partir para uma empreitada mais ambiciosa e cara.

Aplicações futuras da visão computacional seriam inicialmente focadas ainda no ramo de PCIs, e a previsão para expandir o negócio para outras indústrias é de cinco anos.

O primeiro novo produto seria uma melhoria no Abutre, para que ele seja capaz de reconhecer falhas de montagens de componentes em placas. Isso abrangeria um mercado muito maior que o de produção das PCIs, já que a

montagem de placas no Brasil, talvez como um reflexo da Lei da Informática, movimenta muito mais dinheiro que sua fabricação propriamente dita.

Também se pensa em desenvolver uma plataforma de *software* capaz de analisar mais profundamente os dados obtidos pelos aparelhos da Valar, o que adicionaria um valor imenso aos dados brutos prospectados no ambiente industrial. Além disso, isso daria uma experiência grande à Valar na área de otimização e gestão de qualidade industrial, o que abriria o mercado de consultoria para a empresa em um horizonte de 10 a 15 anos.

Produtos futuros da Valar podem incluir:

- Mapeamento 3D de peças e produtos industriais por meio de visão;
- Inspeção visual de segurança de funcionários em ambiente industrial (p. ex. inspeção de uso de Equipamentos de Proteção Individual, controle de acesso a ambientes somente se o funcionário portar os EPIs obrigatórios);
- Inspeção de acabamento e aparência de produtos finais embalados;

5.5. ANÁLISE DO MERCADO RESUMIDA

5.5.1. Resumo do Macro Ambiente

A perspectiva econômica para a indústria no Brasil é, de maneira geral, muito boa. O país tem uma crescente classe C, que passa a fazer parte do mercado consumidor de produtos da linha branca, da linha marrom, e até de automóveis. O fortalecimento do mercado interno é ótima notícia para uma indústria que historicamente é afetada fortemente por crises financeiras internacionais e pela volatilidade cambial. A figura abaixo mostra essa evolução entre 2005 e 2011.

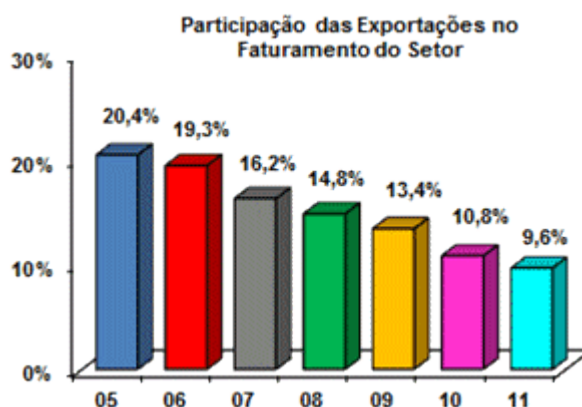


Figura 37 - Participação das exportações no faturamento das Indústrias de Eletroeletrônicos
Fonte: (ABINEE, 2012)

Ainda assim, ela ainda é fortemente afetada pelo cenário externo, como pode ser observado nos gráficos abaixo.

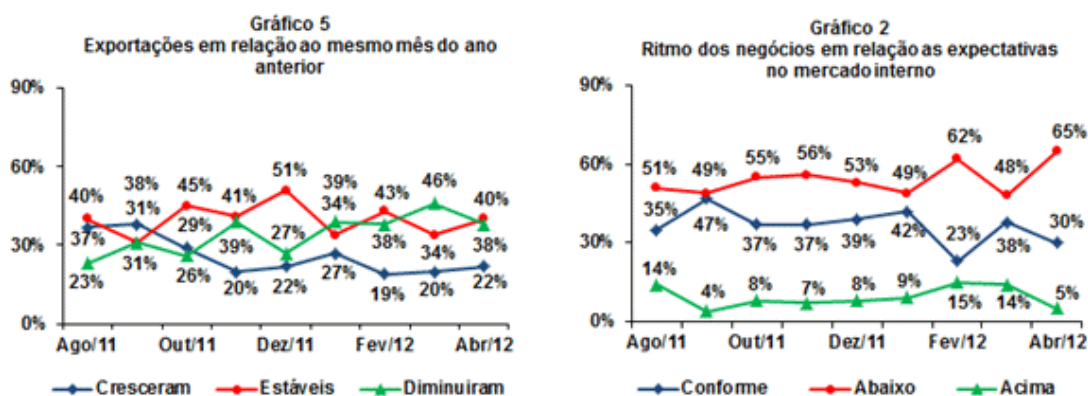


Figura 38 - Expectativas e exportações da indústria de eletroeletrônicos
Fonte: (ABINEE, 2012)

No curto período de tempo coberto pelos gráficos acima, pode-se perceber certa correlação entre a expectativa de negócios no mercado interno e as exportações do setor. No caso ilustrado, é visível a tendência de diminuição de exportações a partir de Ago/2011, e o crescimento dos negócios cujo ritmo é considerado abaixo do esperado, com destaque para Fev/2012 e Abr/2012.

Apesar dessa aparente vulnerabilidade da indústria a fatores externos, o setor de eletroeletrônicos é um setor crescente no Brasil, e a perspectiva em longo prazo é que cresça ainda mais. Programas do governo brasileiro como o Padis (Programa de Apoio ao Desenvolvimento Tecnológico da Indústria de Semicondutores), sancionado em 31 de maio de 2007 (Brasil, 2007), ajudam a ampliar ainda mais o setor, promovendo a abertura de indústrias que hoje não existem no país, injetando capital e aumentando a importância do setor na economia.

Os aspectos socioculturais também são favoráveis para a indústria de eletroeletrônicos. Os aparelhos eletrônicos já permeiam a cultura do brasileiro, e o número de produtos eletrônicos consumidos por grande parte da população aumentou significativamente na década de 2000. Isso pode ser exemplificado com o número de clientes de telefonia celular, ilustrado no gráfico abaixo.

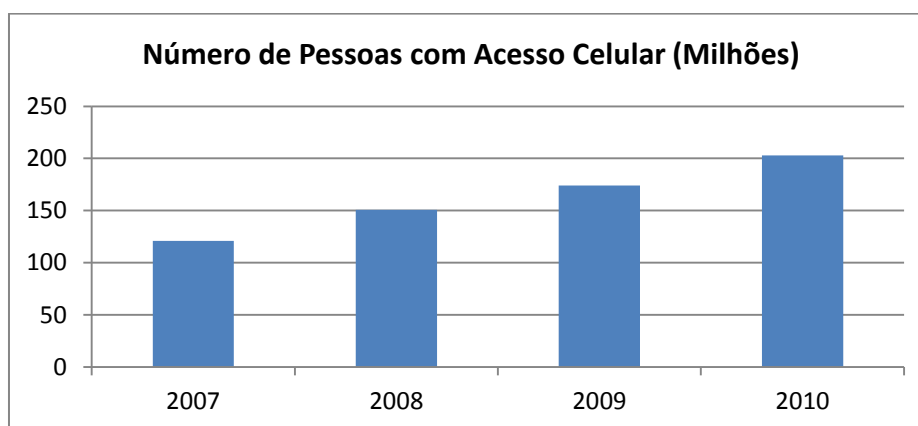


Figura 39 - Uso de telefones celulares de 2007 a 2010
Fonte: Dados: (ANATEL, 2012). Gráfico: Autoria Própria

Uma projeção dos principais indicadores do setor feita pela ABINEE para 2012, tabela 3, mostra que a expectativa é de crescimento.

Indicador	2011	2012	2012 / 2011
Faturamento (R\$ milhões)	134904	152534	13%
Faturamento (US\$ milhões)	80883	82264	2%
Exportações (US\$ milhões)	7880	8275	5%
Importações (US\$ milhões)	40093	46100	15%
Saldo (US\$ milhões)	-32213	-37825	17%
Nº de Empregados (mil)	183	186	2%

Tabela 8 - Projeção de indicadores do setor eletroeletrônico para 2012
Fonte: Dados: (ABINEE, 2011). Acessado em 28/05/2012. Tabela: Autoria Própria

5.5.2. Segmentação do Mercado

A indústria de eletroeletrônicos é subdividida pela ABINEE nos seguintes ramos principais:

- Automação industrial;
- Componentes elétricos e eletrônicos;
- Equipamentos industriais;
- Geração e transmissão de energia;
- Informática;
- Material elétrico de instalação;
- Telecomunicações;
- Utilidades domésticas e eletroeletrônicas;

A tabela abaixo resume o faturamento de cada um dos setores entre 2009 e 2011:

Faturamento Total por Área				
(R\$ milhões a preços correntes)	2009	2010	2011	2011/2010
Automação Industrial	2.943	3.237	3.725	15%
Componentes Elétricos e Eletrônicos	8.263	9.502	9.828	3%
Equipamentos Industriais	15.003	18.754	22.272	19%
Geração e Transmissão de Energia	10.604	12.089	13.097	8%
Informática	35.278	39.864	43.561	9%
Material Elétrico de Instalação	7.954	8.909	9.654	8%
Telecomunicações	18.367	16.714	19.901	19%
Utilidades Domésticas Eletroeletrônicas	13.427	15.307	16.102	5%
Total	111.839	124.376	138.14	11%

Tabela 9 - Faturamento dos setores da Indústria de Eletroeletrônicos
Fonte: Dados: (ABINEE, 2011). Acessado em 28/05/2012. Quadro: Autoria própria

O segmento pretendido pela Valar ao lançar o Abutre é o de Componentes Elétricos e Eletrônicos, ramo ao qual pertencem as indústrias de produção de PCs. O quadro acima mostra, claramente, que esse foi o ramo que menos cresceu de 2010 para 2011, e em pouco tempo deve ser ultrapassado pelo setor de Materiais Elétricos e de Instalação e assumir a posição de ramo com menor faturamento. A partir dessa observação, percebe-se a importância de se aumentar a carteira de produtos da Valar para atender a ramos com uma perspectiva econômica melhor.

5.5.3. Segmento Alvo – Indústria Produtora de Placas de Circuito Impresso

O setor de produção de placas de circuitos impressos no Brasil está em contração, devido principalmente à concorrência internacional. Em 2005 havia 52 empresas filiadas à ABRACI- Associação Brasileira de Fabricantes de Circuitos Impressos. Em 2006, esse número caiu para 50. Ele é composto de empresas por sua maioria consideradas *pequenas* pela ABRACI, utilizando a área de placas produzidas em um mês como critério para a subdivisão entre pequenas, médias e grandes empresas. O gráfico abaixo ilustra o panorama do setor em 2006.

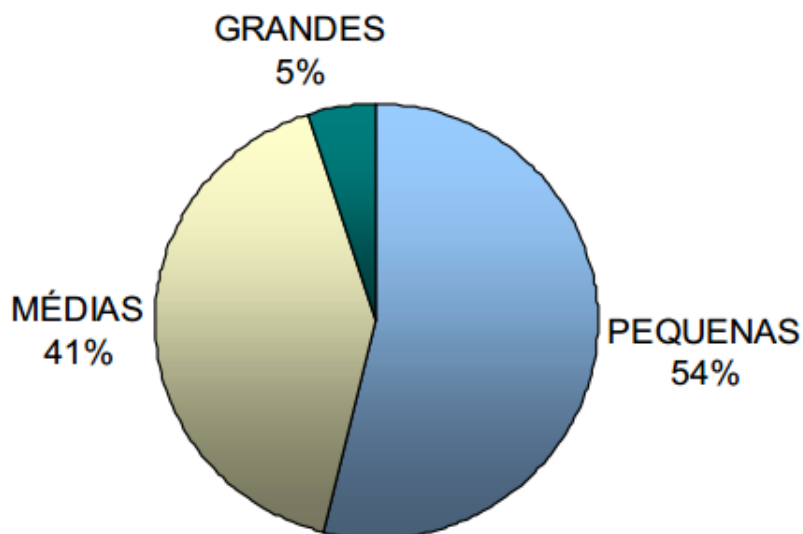


Figura 40 - Divisão das empresas produtoras de PCI de acordo com o tamanho.
Fonte: (ABRACI,2006)

A princípio, o segmento pretendido dentro deste mercado são as **médias empresas**, que já possuem um faturamento adequado para poderem pagar o Abutre, e são de mais fácil abordagem que as de grande porte.

5.5.4. Definição da Proposta de Valor

O Abutre atenderá, numa etapa inicial, às médias empresas produtoras de PCIs. Este é um mercado desprovido de soluções de inspeção óptica automatizada, tendo em vista que as todas as empresas pesquisadas que oferecem esses produtos tem sede no exterior. Isso dificulta o suporte e o atendimento às indústrias, e praticamente inviabiliza a criação de soluções personalizadas para cada caso.

Mais especificamente, será atendido o mercado produtor de placas de camada simples e camada dupla, que, segundo dados da ABRACI, disponíveis na tabela abaixo, representavam a maioria da produção nacional de PCIs em 2006.

ANOS	SIMPLES FACE		DUPLA FACE		MULTICAMADA		TOTAL	
	M ²	US\$ Mil	M ²	US\$ Mil	M ²	US\$ Mil	M ²	US\$ Mil
2003	540.718	12.977	257.303	30.104	57.685	15.142	855.706	58.224
2004	776.060	18.819	273.148	31.958	80.595	18.759	1.129.803	69.537
2005	617.016	14.963	227.564	26.625	89.803	20.903	934.383	62.490
2006 Estimado	437.744	10.615	173.966	20.354	78.928	18.371	690.638	49.341

Tabela 10 - Produção de PCIs no Brasil, subdividida por número de camadas.
Fonte: (ABRACI,2006)

Os clientes serão os gestores de produção e donos dessas indústrias, que precisam se atualizar e modernizar para combater a crescente entrada dos produtores estrangeiros no mercado nacional. O Abutre diminui, além disso, o desperdício do processo de produção e impede que placas defeituosas sejam entregues aos clientes finais.

A figura abaixo resume em um diagrama a proposta de valor do produto proposto.



Figura 41 - Proposta de valor do Abutre.
Fonte: Autoria Própria.

5.6. ESTRATÉGIA

O Abutre, como pode ser concluído da análise de mercado, não é um produto com potencial de mercado suficiente para financiar a estrutura da Valar sozinho. No entanto, é um produto com um segmento de mercado alvo bem definido, acessível e com o qual os sócios já possuem certa experiência. É justamente aí que reside a principal estratégia da empresa quanto à comercialização do produto: ele será a porta de entrada da Valar em um mercado que exige experiência das empresas que contrata.

Nos primeiros meses, um Abutre será emprestado, sem custos, a uma média empresa produtora de PCs de Curitiba. A utilização do produto será acompanhada de perto pela Valar. O estudo de caso dessa empresa será, caso o empresário assim permitir, utilizado como estudo de caso para outros potenciais clientes.

5.6.1. Diferenciais Competitivos

O Abutre é o único produto nacional do gênero, em um mercado repleto de soluções importadas. A proximidade da Valar com os clientes é um diferencial positivo imenso; isso permite a personalização da solução ofertada a um custo mais baixo que a concorrência pode oferecer, se é que tem condições de oferecer.

5.6.2. Vendas

O primeiro produto será emprestado, sem custo, para uma média empresa de Curitiba. O nome de todas as empresas do ramo afiliadas à ABRACI já foi levantado e, em posse dos resultados da utilização do Abutre na empresa curitibana, cada uma delas será visitada pela equipe de engenharia da Valar.

A participação em eventos e feiras industriais também ajuda a promover o produto e a encontrar potenciais clientes.

Após cada venda ser efetuada, a Valar fornecerá treinamento e instrução para os funcionários da empresa, não só sobre a utilização adequada do produto, mas também sobre a criação de uma cultura de qualidade nos processos da empresa, e como o Abutre pode ajudar a empresa nessa questão.

5.6.3. Preço

O preço do produto foi estimado inicialmente em R\$35000,00. Este valor incluiria assistência técnica (e instrução dos funcionários da empresa) por um período de um ano.

Para clientes que necessitem de soluções personalizadas, o orçamento será feito para cada projeto individualmente, e variará de acordo com a complexidade do que for pedido, e com o tempo necessário para implementação da nova solução.

5.6.4. Promoção

A promoção do produto, como o mercado é pequeno, pode ser feita pessoalmente, visitando cada uma das empresas e oferecendo demonstrações práticas do Abutre em ação.

A equipe também pensa em participar de feiras e eventos na área, para se atualizar e, principalmente, entrar em contato com outros empreendedores do setor.

5.6.5. Distribuição

O produto será entregue pessoalmente à empresa que o comprar, e será dado total auxílio à instalação do Abutre no ambiente industrial do cliente.

A princípio os três engenheiros da Valar estarão presentes no momento da entrega, para também, posteriormente, ministrarem o treinamento de utilização do produto.

5.6.6. Futuro

O Abutre não tem potencial de mercado para sustentar e promover o crescimento da Valar. Por conta disso, será iniciado tão cedo quanto possível, após a estrutura física da Valar ser criada, o projeto de um produto capaz de analisar o posicionamento de componentes em uma placa de circuito impresso. Isso ampliaria a carteira de clientes, inicialmente restrita às fabricantes de PCIs, a todas as indústrias que produzirem produtos que contém um circuito eletrônico. Esse produto interessaria tanto às empresas montadoras de placas, quanto aos clientes dessas empresas, que quisessem garantir que a placa comprada veio montada de maneira correta.

Por experiência pessoal de todos os sócios da Valar, a qualidade do serviço ofertado por muitas montadoras de placas é muito baixa, e uma empresa que inserisse no final da produção um processo de inspeção óptica da montagem conseguiria deixar seus clientes muito mais satisfeitos.

5.7. GESTÃO DA EMPRESA

Por se tratar de uma pequena empresa, a princípio, o trabalho dos sócios será dividido entre os responsáveis pelo desenvolvimento e pela gestão da produção e o responsável pela venda e divulgação do produto.

5.7.1. Estrutura Organizacional

Em um primeiro momento a empresa terá uma estrutura informal, já que a equipe é composta apenas pelos sócios e alguns poucos prestadores de serviço. Nessa estrutura haverá apenas duas divisões consideradas importantes para a boa organização da empresa:

- Vendas e contato com os clientes;
- Desenvolvimento e gestão dos fornecedores.

O único prestador de serviço fixo que será contratado pela empresa é o contador.

Os sócios e seus respectivos cargos se encontram relacionados abaixo:

- **André de Castilho**: Diretor de Marketing, Vendas e Financeiro;
- **Marcos Otani**: Diretor de Produção e Qualidade;
- **Mariana Mohr**: Diretora de Desenvolvimento e Projetos Futuros.

5.7.1.1. Vendas e Contato com o Cliente

Como a empresa é uma empresa de pequeno porte voltada ao cliente industrial de um mercado bastante restrito, é possível atingir o público alvo sem a necessidade de grandes campanhas de publicidade. Portanto, um dos sócios será alocado nessa função: a de divulgar o produto e negociar diretamente com os potenciais clientes.

5.7.1.2. Desenvolvimento e Gestão de Fornecedores

Os outros dois sócios da empresa serão responsáveis pelo desenvolvimento de novos produtos e aperfeiçoamento do produto já existente. Eles serão responsáveis também por gerir os fornecedores das partes do produto, uma vez que apenas o software será desenvolvido na própria empresa.

5.7.2. Quadro de Pessoal

A tabela a seguir mostra o quadro de pessoal e os respectivos salários. Nessa tabela foram desconsiderados gastos com prestadores de serviço, como a diarista e o contador, já que a empresa não possui vínculos empregatícios com estes funcionários.

Tabela 11 - Quadro de pessoal

Recurso(s) Humano(s) (Valores em R\$)	2012	2013	2014	2015
Diretor de Marketing; Vendas e Finanças	5.000,00	5.325,00	5.671,13	6.039,75
Diretor de Produção e Qualidade	5.000,00	5.325,00	5.671,13	6.039,75
Diretora de Desenvolvimento e Projetos	5.000,00	5.325,00	5.671,13	6.039,75
TOTAL	15.000,00	15.975,00	17.013,38	18.119,24

Fonte: Autoria própria

5.8. PLANO FINANCEIRO

5.8.1. Projeção de Índices e Taxas

As projeções da tabela a seguir foram feitas com base em resultado anteriores e projeções dos órgãos governamentais responsáveis, sempre considerando o pior dos casos.

Tabela 12 - Projeção de índices e taxas

Semestre	2s2012	1s2013	2s2013	1s2014	2s2014	1s2015
Inflação geral	3,20%	3,20%	3,20%	3,20%	3,20%	3,20%
Financiamentos	5,50%	5,50%	5,50%	5,50%	5,50%	5,50%
Aumento Salarial	0,00%	6,50%	0,00%	6,50%	0,00%	6,50%
Aplicação Financeira	3,92%	3,92%	3,92%	3,92%	3,92%	3,92%

Fonte: Autoria própria

A inflação geral anual estimada pelo COPOM até o ano de 2013 é de 4,5% ao ano, podendo chegar até 6,5% a.a.. Na projeção da Inflação de 2012 até o

início de 2014, foi considerado o pior dos casos, em que a inflação chega a 6,5% a.a. ou 3,20% no semestre. O aumento salarial está diretamente ligado à taxa de inflação e ocorre todo início de ano. A taxa de financiamentos e empréstimos estimada para o ano considerando a maior taxa de juros SELIC em 2012, é de 11,25% a.a. e de 5,5% a.s, enquanto que o retorno de aplicações financeiras é de 8% a.a. ou 3,92% a.s.

Tabela 13 - Encargos sociais

Encargos sociais	Produção	P&D	RH	Finanças	Vendas	Diretoria
INSS	12,00%	12,00%	12,00%	12,00%	12,00%	0,00%
FGTS	8,00%	8,00%	8,00%	8,00%	8,00%	0,00%
Provisões	40,00%	40,00%	40,00%	40,00%	40,00%	0,00%
TOTAL	60,00%	60,00%	60,00%	60,00%	60,00%	0,00%

Fonte: Autoria própria

A partir da tabela acima, verifica-se que o gasto com cada empregado na empresa é de 60% além de seu salário. A diretoria não terá nenhuma contribuição, já que são os próprios donos da empresa.

5.8.2. Orçamento de Produção e de Matérias-Primas

O custo de produção é o custo de produção da placa de circuito impresso e da carcaça, que em um primeiro momento serão compradas montadas do fornecedor, somado aos custos da câmera e do teclado.

Tabela 14 - Orçamento de Produção e de Matérias-Primas

Descrição (Valores em R\$)	Quantidade	Valor Unitario	Valor total
Placa de processamento de dados	1	500,00	500,00
Carcaça	1	1 000,00	1 000,00
Camera	1	600,00	600,00
Teclado	1	32,00	32,00
TOTAL		2 132,00	2 132,00

Fonte: Autoria própria

5.8.3. Orçamento de Mão-de-Obra Direta

Os custos de mão-de-obra na produção correspondem aos salários e encargos dos empregados que se ocupam da produção. Na empresa ABUTRE não há empregados alocados na produção do software, apenas dois sócios se ocuparam dessa atividade. Os encargos foram calculados com base na Tabela 13 e são nulos para sócios.

Tabela 15 - Orçamento de mão-de-obra direta

Mão-de-obra direta (Valores em R\$)	2s2012	1s2013	2s2013	1s2014	2s2014	1s2015
Salários e pró-labore	60 000,00	63 900,00	63 900,00	68 053,50	68 053,50	72 476,98
Encargos Sociais e Provisões	0,00	0,00	0,00	0,00	0,00	0,00
TOTAL	60 000,00	63 900,00	63 900,00	68 053,50	68 053,50	72 476,98

Fonte: Autoria própria

5.8.4. Orçamento de Custos Indiretos de Fabricação

Os custos indiretos de fabricação são custos necessários para a produção, mas genéricos demais para serem apropriados diretamente ao produto. Em um processo de desenvolvimento de software os únicos gastos a serem contabilizados são os gastos com energia elétrica, depreciação e outros custos de pequeno valor.

O cálculo da eletricidade foi baseado no consumo de 2 computadores desktop e 3 lâmpadas fluorescentes, que em conjunto consomem em média 334KWh, o que resulta em uma conta de R\$26,70 mensais, considerando uma tarifa de 0,48R\$/KWh.

Já a depreciação segue uma taxa de 15,3% ao semestre sobre os equipamentos de informática (considerando, como a nova regra da receita define, que um computador tem um tempo de vida de 3 anos) e 5% ao semestre sobre móveis e utensílios. Assim sendo, estima-se uma depreciação de R\$1200 ao semestre sobre o imobilizado utilizado no setor de produção.

Tabela 16 - Orçamento de despesas indiretas de produção

Produção (Valores em R\$)	2s2012	1s2013	2s2013	1s2014	2s2014	1s2015
Eletricidade	160,19	160,19	160,19	160,19	160,19	160,19
Depreciação	795,50	1 591,00	2 386,50	3 182,00	3 977,50	4 773,00
Outros Custos	300,00	309,60	319,51	329,73	340,28	351,17
Total	1 255,69	2 060,79	2 866,19	3 671,92	4 477,97	5 284,36

Fonte: Autoria própria

5.8.5. Orçamento de Despesas Administrativas

A tabela a seguir lista o orçamento de despesas administrativas da empresa. Como um dos sócios será o administrador da empresa e nos primeiros anos não será contratado nenhum funcionário nessa área, os únicos honorários serão da diretoria. A esse custo se soma o aluguel de uma sala, a depreciação dos bens, gastos com energia, telefone e internet, além das despesas gerais.

Tabela 17 - Orçamento de despesas administrativas

Administrativo (Valores em R\$)	2s2012	1s2013	2s2013	1s2014	2s2014	1s2015
Pro-labore	30 000,00	31 950,00	31 950,00	34 026,75	34 026,75	36 238,49
Salario de diarista	1 200,00	1 278,00	1 278,00	1 361,07	1 361,07	1 449,54
Honorarios do contador	1 200,00	1 278,00	1 278,00	1 361,07	1 361,07	1 449,54
Aluguel	3 600,00	3 715,20	3 834,09	3 956,78	4 083,39	4 214,06
Condominio	1 200,00	1 238,40	1 278,03	1 318,93	1 361,13	1 404,69
Depreciação	364,50	729,00	1 093,50	1 458,00	1 822,50	2 187,00
Amortização do projeto	166,67	166,67	166,67	166,67	166,67	166,67
Comunicações	1 500,00	1 548,00	1 597,54	1 648,66	1 701,41	1 755,86
Eletricidade	200,00	206,40	213,00	219,82	226,86	234,11
Agua	480,00	495,36	511,21	527,57	544,45	561,88
IPTU	72,00	74,30	76,68	79,14	81,67	84,28
Despesas de escritório	300,00	309,60	319,51	329,73	340,28	351,17
Despesas de limpeza	300,00	309,60	319,51	329,73	340,28	351,17
TOTAL	40 583,17	43 298,53	43 915,73	46 783,91	47 417,54	50 448,46

Fonte: Autoria própria

5.8.6. Orçamento de Investimentos

Os investimentos em imobilizados utilizados nos cálculos de depreciação são mostrados a seguir.

Tabela 18 - Orçamento de investimentos na produção

Investimentos na produção (Valores em R\$)	2s2012	1s2013	2s2013	1s2014	2s2014	1s2015
Móveis e utensílios						
Mesas de escritório	900,00	0,00	0,00	0,00	0,00	0,00
Cadeiras	510,00	0,00	0,00	0,00	0,00	0,00
Bancada para experimento	500,00	0,00	0,00	0,00	0,00	0,00
Armário simples	200,00	0,00	0,00	0,00	0,00	0,00
Equipamentos de Informática						
Computadores	4600,00	0,00	0,00	0,00	0,00	0,00
TOTAL	6710,00	0,00	0,00	0,00	0,00	0,00

Fonte: Autoria própria

Tabela 19 - Orçamento de investimentos na administração

Investimentos na administração (Valores em R\$)	2s2012	1s2013	2s2013	1s2014	2s2014	1s2015
Móveis e utensílios						
Mesas de escritório	400,00	0,00	0,00	0,00	0,00	0,00
Cadeiras	170,00	0,00	0,00	0,00	0,00	0,00
Equipamentos de Informática						
Notebook	2000,00	0,00	0,00	0,00	0,00	0,00
Impressora	240,00	0,00	0,00	0,00	0,00	0,00
TOTAL	2810,00	0,00	0,00	0,00	0,00	0,00

Fonte: Autoria própria

Os investimentos foram divididos em investimentos na produção e na administração. Como mostram as tabelas acima, o investimento inicial em equipamentos e móveis será de R\$9.520,00. Nos custos da produção estão inclusos dois computadores, três mesas, três cadeiras, uma bancada para experimento e um armário simples. Na administração, por sua vez, estão inclusos uma mesa para escritório, uma cadeira, um notebook e uma impressora.

As tabelas a seguir mostram as movimentações ao longo do tempo dos valores originais do imobilizado.

Tabela 20 - Movimentações do imobilizado na produção

Aquisições da produção (Valores em R\$)	2s2012	1s2013	2s2013	1s2014	2s2014	1s2015
Móveis e utensílios						
Saldo Inicial	0,00	2110,00	2110,00	2110,00	2110,00	2110,00
(+) Novas aquisições	2110,00	0,00	0,00	0,00	0,00	0,00
(-) Vendas	0,00	0,00	0,00	0,00	0,00	0,00
(-) Baixas	0,00	0,00	0,00	0,00	0,00	0,00
(=) Saldo Final	2110,00	2110,00	2110,00	2110,00	2110,00	2110,00
Equipamentos de Informática						
Saldo Inicial	0,00	4600,00	4600,00	4600,00	4600,00	4600,00
(+) Novas aquisições	4600,00	0,00	0,00	0,00	0,00	0,00
(-) Vendas	0,00	0,00	0,00	0,00	0,00	0,00
(-) Baixas	0,00	0,00	0,00	0,00	0,00	0,00
(=) Saldo Final	4600,00	4600,00	4600,00	4600,00	4600,00	4600,00
Resumo Geral						
Saldo Inicial	0,00	6710,00	6710,00	6710,00	6710,00	6710,00
(+) Novas aquisições	6710,00	0,00	0,00	0,00	0,00	0,00
(-) Vendas	0,00	0,00	0,00	0,00	0,00	0,00
(-) Baixas	0,00	0,00	0,00	0,00	0,00	0,00
(=) Saldo Final	6710,00	6710,00	6710,00	6710,00	6710,00	6710,00

Fonte: Autoria própria

Tabela 21 - Movimentações no imobilizado da administração

Aquisições da administração (Valores em R\$)	2s2012	1s2013	2s2013	1s2014	2s2014	1s2015
Móveis e utensílios						
Saldo Inicial	0,00	570,00	570,00	570,00	570,00	570,00
(+) Novas aquisições	570,00	0,00	0,00	0,00	0,00	0,00
(-) Vendas	0,00	0,00	0,00	0,00	0,00	0,00
(-) Baixas	0,00	0,00	0,00	0,00	0,00	0,00
(=) Saldo Final	570,00	570,00	570,00	570,00	570,00	570,00
Equipamentos de Informática						
Saldo Inicial	0,00	2240,00	2240,00	2240,00	2240,00	2240,00
(+) Novas aquisições	2240,00	0,00	0,00	0,00	0,00	0,00
(-) Vendas	0,00	0,00	0,00	0,00	0,00	0,00
(-) Baixas	0,00	0,00	0,00	0,00	0,00	0,00
(=) Saldo Final	2240,00	2240,00	2240,00	2240,00	2240,00	2240,00
Resumo Geral						
Saldo Inicial	0,00	2810,00	2810,00	2810,00	2810,00	2810,00
(+) Novas aquisições	2810,00	0,00	0,00	0,00	0,00	0,00
(-) Vendas	0,00	0,00	0,00	0,00	0,00	0,00
(-) Baixas	0,00	0,00	0,00	0,00	0,00	0,00
(=) Saldo Final	2810,00	2810,00	2810,00	2810,00	2810,00	2810,00

Fonte: Autoria própria

Por estas tabelas é possível perceber que a maior parte do investimento está em equipamentos de informática, já que o foco inicial da empresa é trabalhar com programação.

5.8.7. Depreciação

A depreciação do imobilizado segue a taxa corrente aplicada para móveis e equipamentos de informática. Para móveis é corrente considerar que o tempo de duração é de 10 anos e, portanto, a depreciação é de 10% ao ano ou 5% em um semestre. Já o tempo de duração de equipamentos de informática passou a ser considerado de três anos, então a depreciação desse tipo de imobilizado é de 33,3% ao ano ou 15% no semestre.

Tabela 22 - Depreciação acumulada - Produção

Depreciação acumulada - Produção (Valores em R\$)	2s2012	1s2013	2s2013	1s2014	2s2014	1s2015
Móveis e utensílios						
Saldo Inicial	0,00	105,50	211,00	316,50	422,00	527,50
(+) Depreciações	105,50	105,50	105,50	105,50	105,50	105,50
(-) Baixas	0,00	0,00	0,00	0,00	0,00	0,00
(=) Saldo Final	105,50	211,00	316,50	422,00	527,50	633,00
Equipamentos de Informática						
Saldo Inicial	0,00	690,00	1380,00	2070,00	2760,00	3450,00
(+) Depreciações	690,00	690,00	690,00	690,00	690,00	690,00
(-) Baixas	0,00	0,00	0,00	0,00	0,00	0,00
(=) Saldo Final	690,00	1380,00	2070,00	2760,00	3450,00	4140,00
Resumo Geral						
Saldo Inicial	0,00	795,50	1591,00	2386,50	3182,00	3977,50
(+) Depreciações	795,50	795,50	795,50	795,50	795,50	795,50
(-) Baixas	0,00	0,00	0,00	0,00	0,00	0,00
(=) Saldo Final	795,50	1591,00	2386,50	3182,00	3977,50	4773,00

Fonte: Autoria própria

Tabela 23 - Depreciação acumulada - administração

Depreciação acumulada - Administração (Valores em R\$)	2s2012	1s2013	2s2013	1s2014	2s2014	1s2015
Móveis e utensílios						
Saldo Inicial	0,00	28,50	57,00	85,50	114,00	142,50
(+) Depreciações	28,50	28,50	28,50	28,50	28,50	28,50
(-) Baixas	0,00	0,00	0,00	0,00	0,00	0,00
(=) Saldo Final	28,50	57,00	85,50	114,00	142,50	171,00
Equipamentos de Informática						
Saldo Inicial	0,00	336,00	672,00	1008,00	1344,00	1680,00
(+) Depreciações	336,00	336,00	336,00	336,00	336,00	336,00
(-) Baixas	0,00	0,00	0,00	0,00	0,00	0,00
(=) Saldo Final	336,00	672,00	1008,00	1344,00	1680,00	2016,00
Resumo Geral						
Saldo Inicial	0,00	364,50	729,00	1093,50	1458,00	1822,50
(+) Depreciações	364,50	364,50	364,50	364,50	364,50	364,50
(-) Baixas	0,00	0,00	0,00	0,00	0,00	0,00
(=) Saldo Final	364,50	729,00	1093,50	1458,00	1822,50	2187,00

Fonte: Autoria própria

5.8.8. Orçamento de Aplicações Financeiras e de Financiamento

Os orçamentos de aplicações financeiras e de financiamento só podem ser estimados depois que o fluxo de caixa for feito. Isso porque só então é possível verificar a necessidade e a disponibilidade de dinheiro em caixa. A tabela a seguir mostra os empréstimos tomados para cobrir o *déficit* em caixa dos primeiros semestres.

Tabela 24 - Orçamento de financiamentos

Financiamentos (Valores em R\$)	2s2012	1s2013	2s2013	1s2014	2s2014	1s2015
Taxa de juros (a.s.)	5,50%	5,50%	5,50%	5,50%	5,50%	5,50%
Saldo Inicial		58 670,00	136 460,00	103 565,00	59 356,00	8 966,00
(+) Juros do semestre	0,00	3 226,85	7 505,30	5 696,08	3 264,58	493,13
(-) Amortização	0,00	0,00	32 895,00	44 209,00	50 390,00	8 966,00
(-) Pagamento de Juros	0,00	3 226,85	7 505,30	5 696,08	3 264,58	493,13
(+) Novos empréstimos	58 670,00	77 790,00	0,00	0,00	0,00	0,00
Saldo Final	58 670,00	136 460,00	103 565,00	59 356,00	8 966,00	0,00

Fonte: Autoria própria

A partir do quarto semestre a empresa começa a gerar lucro e pode começar a realizar aplicações. Considerando a taxa de juros da poupança em 2011, de 7,5% ao ano ou 3,68% ao semestre, o resultado das aplicações é mostrado a seguir.

Tabela 25 - Orçamento de Aplicações Financeiras

Aplicações financeiras (Valores em R\$)	2s2012	1s2013	2s2013	1s2014	2s2014	1s2015
Taxa de juros (a.s.)	3,68%	3,68%	3,68%	3,68%	3,68%	3,68%
Saldo Inicial		0,00	0,00	0,00	0,00	0,00
(+) Rendimento	0,00	0,00	0,00	0,00	0,00	0,00
(-) Resgate do principal	0,00	0,00	0,00	0,00	0,00	0,00
(-) Recebimento	0,00	0,00	0,00	0,00	0,00	0,00
(+) Novas aplicações	0,00	0,00	0,00	0,00	0,00	12 355,00
Saldo Final	0,00	0,00	0,00	0,00	0,00	12 355,00

Fonte: Autoria própria

5.8.9. Projeção de Resultados

A seguir são mostradas as estimativas de movimentações financeiras ao longo de três anos utilizadas para levantar a situação econômica e financeira da empresa neste período.

5.8.9.1. Movimentação de contas a receber

Como a única fonte de recebimentos da empresa é a venda de produtos, foram lançados no movimento de contas a receber apenas a previsão de recebimento dos compradores. Foi considerado que o produto será vendido parcelado em 6 vezes sem juros, sendo que os produtos vendidos no primeiro mês do semestre serão pagos totalmente no semestre corrente e os produtos vendidos no fim do semestre serão pagos no semestre seguinte. Fazendo uma média desses pagamentos, foi considerado que metade do pagamento será feito no semestre da venda e metade no semestre seguinte. No final de três anos o saldo final dessa movimentação é positivo porque a segunda metade dos produtos vendidos no primeiro semestre de 2015 será paga apenas no segundo semestre de 2015.

Tabela 26 - Movimentação de contas a receber

Contas a receber (Valores em R\$)	2s2012	1s2013	2s2013	1s2014	2s2014	1s2015
Distribuição do recebimento no tempo						
Faturamento no Semestre						
saldo Inicial		0,00				
2s2012	80500,00	40250,00	40250,00			
1s2013	161000,00	0,00	80500,00	80500,00		
2s2013	201250,00	0,00	0,00	100625,00	100625,00	
1s2014	201250,00	0,00	0,00	0,00	100625,00	100625,00
2s2014	201250,00	0,00	0,00	0,00	0,00	100625,00
1s2015	161000,00	0,00	0,00	0,00	0,00	80500,00
Recebimentos semestrais		40250,00	120750,00	181125,00	201250,00	201250,00
181125,00						
Movimentação contas a receber						
Saldo inicial		0,00	40250,00	80500,00	100625,00	100625,00
(+) Faturamento do semestre		80500,00	161000,00	201250,00	201250,00	201250,00
(-) Recebimento - vendas		40250,00	120750,00	181125,00	201250,00	201250,00
(=) Saldo Final		40250,00	80500,00	100625,00	100625,00	80500,00

Fonte: Autoria própria

5.8.9.2. Movimentação de fornecedores

A tabela abaixo demonstra a movimentação estimada de fornecedores. As movimentações consideradas são relativas à compra do imobilizado e de matéria-prima para a produção estimada em 3 anos, sendo pagas no semestre do recebimento.

Tabela 27 - Movimentação de fornecedores

Movimentação de fornecedores (Valores em R\$)	2s2012	1s2013	2s2013	1s2014	2s2014	1s2015
Saldo Inicial	0,00	62 820,00	0,00	0,00	0,00	0,00
(+) Compra de Imobilizado	9 520,00	0,00	0,00	0,00	0,00	0,00
(+) Compra de Matéria-Prima	53 300,00	0,00	0,00	0,00	0,00	0,00
(-) Pagamento de Imobilizado	0,00	9 520,00	0,00	0,00	0,00	0,00
(-) Pagamento de Matéria-Prima	0,00	53 300,00	0,00	0,00	0,00	0,00
Total	62 820,00	0,00	0,00	0,00	0,00	0,00

Fonte: Autoria própria

5.8.9.3. Movimentação de impostos a recolher

A tabela abaixo mostra as movimentações relativas a impostos a serem recolhidos e pagos.

Tabela 28 - Movimentação de impostos a recolher

Impostos a recolher (Valores em R\$)	2s2012	1s2013	2s2013	1s2014	2s2014	1s2015
Tributos Federais a Recolher						
(+) Tributos Federais sobre vendas	10 500,00	21 000,00	26 250,00	26 250,00	26 250,00	21 000,00
(-) Tributos Federais sobre compras	0,00	0,00	0,00	0,00	0,00	0,00
(=) Saldo do Semestre	10 500,00	21 000,00	26 250,00	26 250,00	26 250,00	21 000,00
(+) Saldo Inicial	0,00	0,00	0,00	0,00	0,00	0,00
(-) Recolhimento do Semestre	10 500,00	21 000,00	26 250,00	26 250,00	26 250,00	21 000,00
(=) Saldo Final	0,00	0,00	0,00	0,00	0,00	0,00
ISS a recolher						
ISS sobre serviços	3 500,00	7 000,00	8 750,00	8 750,00	8 750,00	7 000,00
(=) Saldo do semestre	3 500,00	7 000,00	8 750,00	8 750,00	8 750,00	7 000,00
(+) Saldo Inicial	0,00	0,00	0,00	0,00	0,00	0,00
(-) Recolhimento do semestre	3 500,00	7 000,00	8 750,00	8 750,00	8 750,00	7 000,00
(=) Saldo Final	0,00	0,00	0,00	0,00	0,00	0,00
Provisão para imposto de renda						
Saldo Inicial	0,00	0,00	0,00	0,00	0,00	0,00
(+) IR sobre o lucro	0,00	0,00	0,00	4 985,66	0,00	12 762,15
(-) Recolhimento do semestre	0,00	0,00	0,00	4 985,66	0,00	12 762,15
(=) Saldo Final	0,00	0,00	0,00	0,00	0,00	0,00

Fonte: Autoria própria

Os tributos federais a recolher correspondem à alíquota de 15% (IPI). A tributação incidente sobre o produto é o ISS (5%) (ISS,2011), enquanto que o imposto de renda é pago sobre o lucro (15%) (IRPJ,2011).

5.8.9.4. Movimentação de folha de pagamento, encargos sociais e provisões

A tabela a seguir demonstra os gastos com salários e encargos sociais. Nessa estimativa os encargos são nulos para diarista, contador e diretoria. Só haveria encargo social se houvesse algum vínculo empregatício com algum funcionário, mas esse não é o caso.

Como os salários são pagos no mês seguinte ao início, todos salários são deslocado de um mês no tempo, por isso no primeiro semestre apenas 5 salários são pagos.

Tabela 29 - Movimentação de folha de pagamento e encargos sociais

Cálculo da folha de pagamento (Valores em R\$)	2s2012	1s2013	2s2013	1s2014	2s2014	1s2015
Salário e Honorários brutos						
mão-de-obra direta	0,00	63 900,00	63 900,00	68 053,50	68 053,50	72 476,98
(=) Total da produção	0,00	63 900,00	63 900,00	68 053,50	68 053,50	72 476,98
Administração	2 000,00	2 556,00	2 556,00	2 722,14	2 722,14	2 899,08
(=) Total da Administração	0,00	0,00	0,00	0,00	0,00	0,00
honorários da diretoria	75 000,00	95 850,00	95 850,00	102 080,25	102 080,25	108 715,47
(=) TOTAL GERAL	75 000,00	159 750,00	159 750,00	170 133,75	170 133,75	181 192,44
Desconto de INSS						
Produção	0,00	0,00	0,00	0,00	0,00	0,00
Administração	0,00	0,00	0,00	0,00	0,00	0,00
Soma	0,00	0,00	0,00	0,00	0,00	0,00
Desconto de IR-Fonte						
Produção	0,00	0,00	0,00	0,00	0,00	0,00
Administração	0,00	0,00	0,00	0,00	0,00	0,00
Soma	0,00	0,00	0,00	0,00	0,00	0,00
Salários e Honorários líquidos	75 000,00	159 750,00	159 750,00	170 133,75	170 133,75	181 192,44

Fonte: Autoria própria

5.8.9.5. Movimentação de contas a pagar

A tabela a seguir mostra a movimentação de contas a pagar ao longo dos três primeiros anos da empresa. Despesas como aluguel, condomínio, eletricidade e água são pagos apenas no mês seguinte ao seu uso, por isso essas despesas são deslocadas de um mês na movimentação de contas a pagar.

Tabela 30 - Movimentação de contas a pagar

Contas a pagar (Valores em R\$)	2s2012	1s2013	2s2013	1s2014	2s2014	1s2015
Saldo Inicial	0,00	1202,03	1239,64	1278,46	1318,51	1359,85
(-) Pagamentos	8070,15	10649,04	12070,44	13500,21	14938,61	16385,92
(+) Consumo produção	1255,69	2060,79	2866,19	3671,92	4477,97	5284,36
(+) Consumo administração	8016,50	8625,86	9243,06	9868,35	10501,98	11144,22
(=) Saldo final	1202,03	1239,64	1278,46	1318,51	1359,85	1402,51

Fonte: Autoria própria

5.8.10. Projeção do fluxo de caixa

O fluxo de caixa abaixo agrupa todas as informações colhidas anteriormente para fazer uma previsão do fluxo de caixa ao longo do tempo para a empresa. O saldo inicial corresponde a etapa de pré-projeto.

Tabela 31 - Fluxo de caixa ao longo dos três primeiros anos

Projeção do Fluxo de Caixa (Valores em R\$)	2s2012	1s2013	2s2013	1s2014	2s2014	1s2015
ATIVIDADES DE OPERAÇÕES						
RECEBIMENTOS						
Vendas de Produtos	40 250,00	120 750,00	181 125,00	201 250,00	201 250,00	181 125,00
Total	40 250,00	120 750,00	181 125,00	201 250,00	201 250,00	181 125,00
PAGAMENTOS						
Fornecedores	0,00	53 300,00	0,00	0,00	0,00	0,00
ISS+Tributos Federais (líquidos)	14 000,00	28 000,00	35 000,00	35 000,00	35 000,00	28 000,00
Salários e Honorários da diretoria	75 000,00	94 875,00	95 850,00	101 041,88	102 080,25	107 609,60
Contribuições previdenciárias	0,00	0,00	0,00	0,00	0,00	0,00
Imposto de Renda - Fonte						
Férias, 13 salário e rescisões	0,00	0,00	0,00	0,00	0,00	0,00
Salário de diarista	1 000,00	1 265,00	1 278,00	1 347,23	1 361,07	1 434,79
Honorários do contador	1 000,00	1 265,00	1 278,00	1 347,23	1 361,07	1 434,79
Aluguel	3 000,00	3 696,00	3 814,27	3 936,33	4 062,29	4 192,28
Comunicações	1 250,00	1 540,00	1 589,28	1 640,14	1 692,62	1 746,79
Eletricidade	300,15	365,52	372,09	378,87	385,87	393,09
Água	400,00	492,80	508,57	524,84	541,64	558,97
IPTU	60,00	73,92	76,29	78,73	81,25	83,85
Despesas de escritório	300,00	309,60	319,51	329,73	340,28	351,17
Despesas de limpeza	300,00	309,60	319,51	329,73	340,28	351,17
Outros custos	300,00	309,60	319,51	329,73	340,28	351,17
TOTAL	96 910,15	185 802,04	140 725,02	146 284,43	147 586,90	146 507,68
SUPERÁVIT ANTES DOS JUROS E IR	-56 660,15	-65 052,04	40 399,98	54 965,57	53 663,10	34 617,32
(+) Recebimentos de juros	0,00	0,00	0,00	0,00	0,00	0,00
(-) Pagamentos de encargos financ.	0,00	3 226,85	7 505,30	5 696,08	3 264,58	493,13
(-) Imposto de Renda sobre o lucro	0,00	0,00	0,00	4 985,66	0,00	12 762,15
(=) SUPERÁVIT APÓS JUROS E IR	-56 660,15	-68 278,89	32 894,68	44 283,84	50 398,52	21 362,04
Atividades de Investimentos						
(-) Investimento	0,00	0,00	0,00	0,00	0,00	12 355,00
(-) Fornecedores de Imobilizado	0,00	9 520,00	0,00	0,00	0,00	0,00
(+) Resgate (aplicações financeiras)	0,00	0,00	0,00	0,00	0,00	0,00
(=) SUPERÁVIT (DÉFICIT)	0,00	-9 520,00	0,00	0,00	0,00	-12 355,00
Atividades de financiamentos						
Empréstimos em R\$ - amortização	58 670,00	77 790,00	-32 895,00	-44 209,00	-50 390,00	-8 966,00
(=) SUPERÁVIT (DÉFICIT)	58 670,00	77 790,00	-32 895,00	-44 209,00	-50 390,00	-8 966,00
Caixa						
superávit (déficit) geral	2 009,85	-8,89	-0,32	74,84	8,52	41,04
(+) Saldo inicial	-1 000,00	1 009,85	1 000,96	1 000,64	1 075,47	1 083,99
(-) Saldo mínimo (final)	1 000,00	1 000,00	1 000,00	1 000,00	1 000,00	1 000,00
(=) Aplicação (Captação) em R\$	9,85	0,96	0,64	75,47	83,99	125,03

Fonte: Autoria própria

5.8.11. Demonstração do Resultado em Exercício (DRE)

No quadro abaixo foi feita uma previsão de demonstração de resultados para os próximos três anos, tendo como base todas as análises feitas ao decorrer do plano financeiro.

Tabela 32 - Demonstração do resultado em exercício

Demonstração de resultados (Valores em R\$)	2s2012	1s2013	2s2013	1s2014	2s2014	1s2015
RECEITA OPERACIONAL BRUTA						
Vendas de produtos/serviços	80 500,00	161 000,00	201 250,00	201 250,00	201 250,00	161 000,00
(-) Deduções						
PIS, Cofins	10 500,00	21 000,00	26 250,00	26 250,00	26 250,00	21 000,00
ISS	3 500,00	7 000,00	8 750,00	8 750,00	8 750,00	7 000,00
(=) RECEITA OPERACIONAL LÍQUIDA	66 500,00	133 000,00	166 250,00	166 250,00	166 250,00	133 000,00
(-) CUSTO DOS PRODUTOS VENDIDOS	61 255,69	65 960,79	66 766,19	71 725,42	72 531,47	77 761,33
(=) LUCRO (PREJUÍZO) BRUTO	5 244,31	67 039,21	99 483,81	94 524,58	93 718,53	55 238,67
(-) DESPESAS OPERACIONAIS						
Despesas Administrativas	40 583,17	43 298,53	43 915,73	46 783,91	47 417,54	50 448,46
(=) LUCRO (PREJUÍZO) ANTES DOS ENCARGOS FINANC.	-35 338,85	23 740,68	55 568,08	47 740,68	46 300,99	4 790,21
DESPESAS FINANCEIRAS LÍQUIDAS						
(+) Receita de juros	0,00	0,00	0,00	0,00	0,00	0,00
(-) Despesas de juros	0,00	3 226,85	7 505,30	5 696,08	3 264,58	493,13
(=) LUCRO (PREJUÍZO) ANTES DO IMPOSTO DE RENDA	-35 338,85	20 513,83	48 062,78	42 044,60	43 036,41	4 297,08
(-) IMPOSTO DE RENDA (15%)	0,00	0,00	0,00	4 985,66	0,00	12 762,15
(=) LUCRO (PREJUÍZO) LÍQUIDO	-35 338,85	20 513,83	48 062,78	37 058,94	43 036,41	-8 465,08

Fonte: Autoria própria

Essa tabela mostra que a empresa operará com prejuízo no primeiro e no último semestre, o que mostra que é necessário desenvolver um novo produto para que a empresa continue viável depois dos três primeiros anos.

5.8.12. Payback

A partir do fluxo de caixa da empresa demonstrado na Tabela 31 é possível perceber que o retorno dos investimentos iniciais ocorrerá no final do segundo semestre do terceiro ano de operação, então o *payback* desses investimentos é de 2 anos e 8 meses.

5.8.13. Análise de VPL (Valor Presente Líquido)

O cálculo do VPL é feito normalmente para avaliar se o investimento é vantajoso ou não. Para isso é feita uma comparação do rendimento obtido com a empresa ao final de um período definido (no caso de três anos) com o rendimento que seria obtido com uma aplicação mais segura, levando em conta também os riscos do negócio.

No caso mais otimista, essa taxa de juros é composta de:

- Taxa de juros do governo americano 0,25%
- Taxa risco Brasil 2,40%

- Taxa risco setor 5%
- Taxa risco empresa 25%

No total, a taxa mínima de atratividade é de 32,65% ao ano. Então, para que um investimento seja considerado economicamente viável, a sua expectativa de rendimento deve ser de ao menos 32,65% a.a. ou 15,17% a.s.

Tabela 33 - Calculo do valor presente liquido dos superavits (déficits) futuros

VPL (Valores em R\$)	2s2012	1s2013	2s2013	1s2014	2s2014	1s2015	TOTAL
Juros futuros	0,00%	15,17%	30,34%	45,51%	60,68%	75,85%	
Entrada - saída	-56 660,15	-68 278,89	32 894,68	44 283,84	50 398,52	21 362,04	
VPL	-56 660,15	-59 285,31	25 237,60	30 433,53	31 365,77	12 147,88	-16 760,69

Fonte: Autoria própria

O VPL negativo indica que o negócio não é viável economicamente, ou seja, que o retorno previsto não compensa o lucro perdido em um investimento seguro somado ao risco desse tipo de negócio.

5.8.14. TIR (Taxa Interna de Retorno)

A TIR é a taxa que, aplicada aos valores futuros, resultaria em um VPL de zero. Essa taxa indica a taxa de retorno do projeto e deve ao menos compensar o custo do capital do projeto. A tabela a seguir demonstra que o TIR para que o VPL seja nulo é de 7,07% ao semestre ou 14,63% ao ano, que é menos do que a taxa de financiamento de capital. Isso mostra, mais uma vez, que o negócio não é viável economicamente.

Tabela 34 - Estimativo da TIR

VPL (Valores em R\$)	2s2012	1s2013	2s2013	1s2014	2s2014	1s2015	TOTAL
Juros futuros	0,00%	7,07%	14,14%	21,20%	28,27%	35,34%	
Entrada - saída	-56 660,15	-68 278,89	32 894,68	44 283,84	50 398,52	21 362,04	
VPL	-56 660,15	-63 771,55	28 820,63	36 536,66	39 290,41	15 784,01	0,00
TIR	7,07%						

Fonte: Autoria própria

7. CONCLUSÃO

7.1. ASPECTOS TÉCNICOS

O desenvolvimento do projeto se subdividiu em três tarefas principais: a concepção do algoritmo de reconhecimento de falhas, a criação de um sistema de aquisição de imagens e o desenvolvimento do *software* na plataforma computacional *BeagleBoard-xM*.

Na primeira parte do desenvolvimento do projeto, tentou-se fazer, a princípio, com que fossem reconhecidas falhas em placas já envernizadas e com a arte final aplicada. Após diversas tentativas de segmentação de imagens de placas envernizadas, verificou-se que seria muito difícil desenvolver um algoritmo confiável para este fim. Os trabalhos, então, se voltaram a segmentação de placas com o cobre aparente, o que se mostrou muito mais simples e preciso. O reconhecimento de falhas foi implementado após extensa pesquisa na literatura, e os trabalhos de Tatibana, et al. se mostraram especialmente úteis. A versão final implementada continha algumas ideias deste trabalho, mas com alterações importantes para que ele funcionasse também para placas sem furação alguma.

O sistema de aquisição de imagens foi crucial para simplificar o algoritmo de segmentação das imagens adquiridas. A concepção do sistema foi um processo longo, incremental, mas que se mostrou proveitoso por causa das inúmeras simplificações no algoritmo que puderam ser feitas por causa do ambiente extremamente controlado que foi construído.

Ao final do processo, pode-se constatar que o sistema de inspeção de placas de circuito impresso desenvolvido foi capaz de reconhecer com sucesso todos os erros presentes nas placas testadas, desde que não houvesse pontos de ferrugem acentuada no cobre que prejudicasse a segmentação. Além disso, o algoritmo implementado foi eficiente do ponto de vista do consumo de memória, utilizando cerca de 17MiB dos 1GiB disponíveis na plataforma de desenvolvimento, quando utilizadas imagens de aproximadamente 1500 x 1500 pixels.

7.2. TRABALHOS FUTUROS

Possíveis trabalhos futuros deste trabalho são o desenvolvimento de algoritmos de detecção de falhas de conformidade nos componentes montados em uma placa de circuito impresso e de falhas de solda. O primeiro tipo de falhas abrange falhas de falta ou excesso de componentes, rotação de componentes e de componentes incorretos (resistor com valor incorreto, por exemplo). Já a detecção de falhas em solda deve detectar problemas como solda fria e exige algoritmos mais complexos e iluminação apropriada.

Para tanto, a estrutura física construída para aquisição de imagens, a plataforma computacional utilizada, e o *software* de interação com o usuário se adequariam perfeitamente, bastando modificar e desenvolver novos algoritmos de visão computacional.

REFERÊNCIAS BIBLIOGRÁFICAS

ABINEE. 2012. Exportações no ramo das Indústrias de Eletro-eletrônicos. [Online] 2012. [Citado em: 28 de Maio de 2012.] <http://www.abinee.org.br/abinee/decon/decon15.htm>.

— **2011.** Indicadores Econômicos. [Online] 2011. [Citado em: 28 de Maio de 2012.] <http://www.abinee.org.br/abinee/decon/decon16.htm>.

— **2012.** Projeções Econômicas. [Online] 2012. [Citado em: 28 de Maio de 2012.] <http://www.abinee.org.br/abinee/decon/decon11.htm>.

Amazon.com, Inc. 2012. X-Large Photo Studio-In-A-Box. *Amazon*. [Online] 2012. [Citado em: 09 de 05 de 2012.] http://www.amazon.com/Dot-Line-DOTLCOSIB200-X-Large-Studio-In-A-Box/dp/B001DCBWK6/ref=sr_1_4?ie=UTF8&qid=1337043118&sr=8-4.

ANATEL. 2012. Estatísticas ANATEL. [Online] 2012. [Citado em: 2012 de Maio de 28.] <http://www.anatel.gov.br/Portal/exibirPortallInternet.do>.

Angstrom Distribution. 2012. *Angstrom Distribution*. [Online] 2012. [Citado em: 20 de 06 de 2012.] <http://www.angstrom-distribution.org/>.

Bay, Herbert, et al. 2008. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*. 2008, Vol. 110.

BeagleBoard.org. 2010. BeagleBoard-xM System Reference Manual. *BeagleBoard Official Website*. [Online] 04 de 04 de 2010. [Citado em: 20 de 06 de 2012.] http://beagleboard.org/static/BBxMSRM_latest.pdf.

Borba, João Francisco. 1994. *Proposta de um sistema computadorizado de inspeção automática de placas de circuito impresso [DISSERTAÇÃO]*. Curitiba : s.n., 1994.

Borgefors, Gunilla. 1986. Distance transformations in digital images. *CVGIP*. 1986, Vol. 34, 3.

Bradski, G. 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*. 2000.

Brasil. 2007. Lei nº 11484, de 31 de maio de 2007. [Online] 2007. [Citado em: 28 de 05 de 2012.] http://www.planalto.gov.br/ccivil_03/_ato2007-2010/2007/lei/l11484.htm.

Canon, Inc. 2012. Canon T2i Instruction Manual. *Canon*. [Online] 01 de 02 de 2012. [Citado em: 22 de 06 de 2012.] <http://gdip01.c-wss.com/gds/9/0300004289/02/eosrt2i-eos550d-im3-c-en.pdf>.

Chen, Zhikun e Wang, Qing. 2010. Research of PCB Image Segmentation Based on Color Features. *ICACTE*. 2010.

Everingham, M., et al. 2010. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*. 2010, Vol. 88, 2.

Gamma, Erich, et al. 1994. *Design Patterns - Elements of Reusable Object-Oriented Software*. s.l. : Addison-Wesley, 1994.

Gokturk, S.B., Akarun, L. e Bozma, H.I. Automated Inspection of PCB's using a Novel Approach. *Stanford*.

Gonzalez, Rafael e Woods, Richard. 2008. *Digital Image Processing*. Upper Saddle River : Prentice Hall, 2008.

Hartley, Richard e Andrew, Zisserman. 2004. *Multiple View Geometry in Computer Vision*. Cambridge : Cambridge University Press, 2004.

Jian, F e Dong, Ming-li. 2010. An Auto Multi-threshold Segmentation Approach of PCB Image. *ICINA*. 2010.

Leta, F. R., Feliciano, F. F. e Martins, F. P. R. 2008. Computer vision System for Printed Board Inspection. *ABCM Symposium Series in Mechatronics*. 2008, Vol. 3.

libgphoto2 Development Team. 2007. libgphoto2 Photo Camera Library Documentation. *gphoto.org*. [Online] 22 de 07 de 2007. [Citado em: 20 de 06 de 2012.] <http://gphoto.org/doc/api/>.

Linaro. 2012. *Linaro*. [Online] 2012. [Citado em: 20 de 06 de 2012.] <http://www.linaro.org/>.

Lowe, David G. 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*. 2004, Vol. 60, 2.

Mathworks Inc. 2011. Documentation on the function `adapthisteq`. *MATLAB Image Processing Toolbox Help*. 2011.

Moganti, M, et al. 1996. Automatic PCB Inspection Algorithms: A Survey. *COMPUTER VISION AND IMAGE UNDERSTANDING*. Março de 1996, Vol. 63, 2, pp. 287-313.

Nethercote, Nicholas, Walsh, Robert e Fitzhardinge, Jeremy. 2006. Building Workload Characterization Tools with Valgrind. *IEEE International Symposium on Workload Characterization*. 2006.

Nokia Corporation. 2012. *Qt*. [Online] 2012. [Citado em: 20 de 05 de 2012.] <http://qt.nokia.com/>.

—. **2008.** Nokia to Acquire Trolltech Press Release. [Online] 28 de 01 de 2008. [Citado em: 20 de 05 de 2012.] <http://qt.nokia.com/about/news/archive/press.2008-01-28.4605718236/>.

—. **2011.** Qt State Machine Framework. *Qt Labs*. [Online] 2011. [Citado em: 20 de 06 de 2012.] <http://doc.qt.nokia.com/4.7-snapshot/statemachine-api.html>.

Otsu, Nobuyuki. 1979. A Threshold Selection Method from Gray-Level Histograms. *Systems, Man and Cybernetics, IEEE Transactions on*. 1979, Vol. 9, 1.

Scuri, Antonio. 2009. ImLab Sourceforge Page. *ImLab 2.3*. [Online] 10 de 06 de 2009. [Citado em: 20 de 06 de 2012.] <http://imlab.sourceforge.net/>.

Suzuki, S. e Abe, K. 1985. Topological Structural Analysis of Digitized Binary Images by Border Following. *CVGIP*. 30, 1985, Vol. 1.

Tatibana, M. A. e Lotufo, R.A. 1997. Novel Automatic PCB Inspection Technique Based on Connectivity. *IEEE*. 1997.

Tecnocroma Imagem e Comércio Ltda. 2012. Iluminação para Fotografia Digital. *Tecnocroma*. [Online] 2012. [Citado em: 09 de 05 de 2012.] <http://www.tecnocroma.com.br/acess%C3%B3rios.para.fotografia.htm>.

Theodoridis, Sergios e Koutroumbas, Konstantinos. 2008. *Pattern Recognition*. 4^a. s.l. : Academic Press, 2008.

Trygve, Reenskaug. 1979. Models - Views - Controllers. [Online] 10 de 12 de 1979. [Citado em: 20 de 06 de 2012.] <http://heim.ifi.uio.no/~trygver/1979/mvc-2/1979-12-MVC.pdf>.

Valgrind-developers. 2011. Valgrind Official Webpage. *Valgrind*. [Online] 2011. [Citado em: 20 de 06 de 2012.] <http://valgrind.org>.

Yocto Project. 2012. Yocto Project. [Online] 2012. [Citado em: 20 de 06 de 2012.] www.yoctoproject.org.