

UNIVERSIDADE FEDERAL DE MINAS GERAIS

ESCOLA DE ENGENHARIA

Programação de Pós-Graduação em Engenharia Elétrica

RECONHECIMENTO DE CARACTERES ESTAMPADOS EM
TARUGOS DE AÇO

Marcos Vinicius Lopes Pereira

Dissertação de mestrado submetida à banca examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do Título de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Walmir Matos Caminhas

Belo Horizonte – MG

Maio de 2006

AGRADECIMENTOS

Expresso a minha imensa gratidão às pessoas e às instituições que contribuíram para a realização desta dissertação:

Agradeço a Deus, primeiramente, por me prover saúde e a oportunidade de estudar e conviver em uma das melhores universidades do país;

Ao Prof. Walmir Matos Caminhas, pela inestimável orientação, pelos valiosos comentários e ensinamentos, pela atenção e amizade que sempre demonstrou e principalmente pela paciência.

Ao Marcelo e ao Prof. Antônio de Pádua Braga pelas discussões relevantes e soluções desenvolvidas;

Ao Douglas, Turíbio e demais funcionários da Açominas que auxiliaram enormemente o desenvolvimento do projeto;

À minha mãe Tereza pelo suporte e amor e ao meu pai Pedro cujos ensinamentos honrosos ainda guardo comigo.

À Açominas pelo apoio financeiro.

RESUMO

Esta dissertação descreve a implementação e análise de ferramentas propostas para reconhecimento de caracteres. A mesma promove ainda discussões sobre a sua viabilidade de aplicação em um problema real na Gerdau Açominas. Trata-se do problema de identificação de números de Ordem de Produção (OP) em peças de aço (tarugos). O reconhecimento visa a automatização completa da LIT (linha de inspeção de tarugos). Com a possibilidade de identificação da OP, o sistema de reconhecimento deverá ser capaz de modificar as variáveis paramétricas da linha de inspeção de maneira a separar os tarugos para cada cliente, evitando-se problemas como mistura de peças e inspeções com parâmetros errados. Propõe-se um condicionamento e componentes (hardware) necessários para que seja possível realizar uma coleta de informações (imagens) adequadamente. A mesma é feita de acordo com as limitações impostas pelo processo e sugerem-se melhorias de acordo com o contexto. Analisam-se algoritmos de segmentação, extração de características e classificação sugeridos para o processo. Diante de novos desafios que se apresentam, são propostas abordagens alternativas e sugestões para ajustes visando um desempenho otimizado das estratégias. Uma discussão é feita sobre a razão de compromisso entre eficiência e tempo de processamento já que técnicas de processamento de imagem costumam ter alto custo computacional. São analisadas diversas imagens coletadas do processo real e o desempenho de cada etapa de processamento. Resultados das metodologias mostraram-se insatisfatórios, porém é aberta uma discussão a respeito da qualidade da imagem e o formato da estampagem dos caracteres no tarugo. São apontados itens críticos para mudança no processo e sugeridas modificações nas estratégias de maneira que se adequem melhor ao problema.

ABSTRACT

This dissertation presents the implementation and analysis of tools that aims to achieve characters recognition. This tool promotes discussions on its viability application of a real problem at Gerdau Açominas. The problem consists in the identification of the Production Order number (PO) stamped on steel objects (billets). The recognition goal is the complete automation of the BIL (Billets Inspection Line). Achieving the identification of the PO, the recognition system must be capable of modifying the parametric variables of the inspection line in order to separate the billets for each customer, preventing problems as, for example, mixture of parts and inspections with wrong parameters. The study begins considering a necessary conditioning and components (hardware) to make possible the gathering of information (images) in a proper manner. The collection of images is made inside the limitations imposed by the process and viable improvements in accordance with the context are suggested. Segmentation, feature extraction and classification algorithms are analyzed. In the face of new challenges that it presents, alternative approaches and adjustments aiming an optimized performance of the strategies are suggested. The reason of commitment between efficiency and time of processing are discussed, since image processing methods tend to have high computational cost. Diverse collected images of the real process and the performance of each stage of processing are analyzed. Results of the methodologies turned out unsatisfactory; however a discussion is carried through regarding the quality of the image and the format of the characters that are stamping the billets. Critical points intending the change in the process are mentioned and modifications in the strategies are suggested in order to better fit the problem.

SUMÁRIO

LISTA DE FIGURAS.....	VII
LISTA DE TABELAS.....	X
1 – INTRODUÇÃO	1
1.1 – MOTIVAÇÃO DO PROJETO	1
1.2 – ESPECIFICAÇÕES DO PROJETO.....	2
1.3 – CONTRIBUIÇÃO DA DISSERTAÇÃO	4
1.4 – ORGANIZAÇÃO DO TRABALHO.....	5
2 – PROCESSAMENTO DIGITAL DE IMAGENS	6
2.1 – MÉTODOS DE LIMARIZAÇÃO (BINARIZAÇÃO).....	6
2.1.1 – <i>Otsu</i>	6
2.1.2 – <i>Niblack</i>	7
2.2 – DETECÇÃO DE BORDAS (SOBEL)	7
2.3 – CORRELAÇÃO CRUZADA NORMALIZADA BIDIMENSIONAL (NCC).....	8
2.4 – SEGMENTAÇÃO NUMÉRICA	13
2.4.1 – <i>Projeção dos pixels</i>	14
2.4.2 – <i>Análise de perfil de variância</i>	15
2.4.3 – <i>Transformada de Hough</i>	16
2.5 – ABORDAGENS ALTERNATIVAS EXISTENTES NA LITERATURA	19
2.6 – CONCLUSÕES	19
3 – DESCRIÇÃO DA LINHA DE INSPEÇÃO DE TARUGOS DA AÇOMINAS	20
3.1 – CONCLUSÕES	24
4 – SISTEMA DE VISÃO COMPUTACIONAL.....	25
4.1 – DESCRIÇÃO DOS COMPONENTES DO <i>HARDWARE</i> UTILIZADOS.....	25
4.2 – DESCRIÇÃO DO <i>SOFTWARE</i> E METODOLOGIAS PROPOSTAS	31
4.2.1 – <i>Configuração da aplicação</i>	32
4.2.2 – <i>Armazenamento da lista contendo as ordens de produção (OP)</i>	32
4.2.3 – <i>Modo de espera por eventos</i>	32
4.2.4 – <i>Captura da imagem</i>	33
4.2.5 – <i>Correção da rotação conhecida</i>	33
4.2.6 – <i>Segmentação do tarugo ou localização do tarugo na cena</i>	34
4.2.6.1 – <i>Estratégia inicial implantada</i>	35
4.2.6.2 – <i>Alternativa utilizando perfil de variância</i>	37
4.2.7 – <i>Rotação</i>	42
4.2.8 – <i>Redução do espaço de busca</i>	43
4.2.9 – <i>Correlação cruzada normalizada feita com padrões desejados</i>	46
4.2.10 – <i>Binarização</i>	48
4.2.11 – <i>Comparação com o banco de padrões</i>	51
4.2.12 – <i>Montagem da seqüência numérica reconhecida</i>	52
4.2.13 – <i>Planejamento do software de identificação</i>	55
4.3 – CONCLUSÕES	57
5 – RESULTADOS E DISCUSSÕES.....	58

5.1 – CORREÇÃO DA ROTAÇÃO CONHECIDA	62
5.2 – SEGMENTAÇÃO DO TARUGO NA CENA	62
5.3 – ROTAÇÃO	68
5.4 – REDUÇÃO DO ESPAÇO DE BUSCA	69
5.5 – CORRELAÇÃO CRUZADA NORMALIZADA APÓS BINARIZAÇÃO	71
6 – CONCLUSÕES.....	73
PERSPECTIVAS FUTURAS	75
7 – REFERÊNCIAS	79

LISTA DE FIGURAS

Figura 1.1 – Exemplo de tarugo estampado	3
Figura 2.1 – Limiarização (T_h) automática proposta por OTSU	7
Figura 2.2 – Operação de correlação normalizada	9
Figura 2.3 – Exemplo de cálculo de correlação.....	10
Figura 2.4 – Padrão escolhido	11
Figura 2.5 – Imagem a ser analisada	11
Figura 2.6 – Operação de correlação cruzada normalizada (região azul-clara indicando valores ignorados).....	12
Figura 2.7 – Exemplo Seqüência Numérica	14
Figura 2.8 – Somatório dos pixels de cada linha da imagem binária da figura 2.7.....	14
Figura 2.9 – Somatório dos pixels de cada coluna da imagem binária da figura 2.7	14
Figura 2.10 – Perfil de variância horizontal da figura 2.7.....	15
Figura 2.11 – Perfil de variância vertical da figura 2.7.....	15
Figura 2.12 – Inserção de uma região com rugosidade em imagem lisa.....	16
Figura 2.13 – Perfil de variância horizontal da figura 2.12.....	16
Figura 2.14 – Perfil de variância vertical da figura 2.12.....	16
Figura 2.15 – Em (A) Pontos candidatos. Em (B) e (C) possíveis retas que cruzam os pontos.....	17
Figura 2.16 – Parametrização polar.....	17
Figura 2.17 – Em (A): Exemplo de detecção de bordas utilizando operador SOBEL. Em (B): Exemplo de detecção de linhas utilizando a transformada de HOUGH (número de retas desejadas = 10).....	18
Figura 2.18 – Resultados de r e θ evidenciando o resultado da transformada de Hough selecionando os 10 picos mais evidentes.....	18
Figura 2.19 – Combinação de algoritmos de contorno e transformada de Hough	19
Figura 3.1 – Processo de produção de tarugos	20
Figura 3.2 – Leito de resfriamento dos tarugos	21
Figura 3.3 – Bobinas de fio-máquina	21
Figura 3.4 – Visão do tarugo já inspecionado	23
Figura 4.1 – Iluminação para realçar detalhes no plano do tarugo. Em (A): vista da câmera. Em (B): vista lateral do tarugo. FONTE: [1]	26
Figura 4.2 – Comparação do efeito da iluminação estruturada nos dois possíveis planos de trabalho. FONTE: [1].....	27
Figura 4.3 – Diagrama das instalações onde será localizado o sistema desenvolvido ...	28
Figura 4.4 – Visão do conjunto câmera e iluminação	29
Figura 4.5 – Posicionamento da câmera focando a mesa de rolos	29
Figura 4.6 – Componentes da câmera PELCO PTZ.....	30
Figura 4.7 – Conexão da sala de automação com a câmera	31

Figura 4.8 – Em (C) a imagem (A) após rotação estimada. Em (D) a imagem (B) após rotação estimada	34
Figura 4.9 – Em (A): subtração da componente vermelha na azul. Em (B): resultado da limiarização global	35
Figura 4.10 – Operação no registro de aproximações sucessivas (SAR) (exemplo para um conversor de 4 bits)	36
Figura 4.11 – Iterações do algoritmo de segmentação do tarugo, resultado final em vermelho	37
Figura 4.12 – Binarização utilizada no item anterior	38
Figura 4.13 – Detecção de bordas na FIG.-4-C usando operador de Sobel [12] e limiar automático (eixos coordenados conforme convenção ditada pela varredura de vídeo) .	39
Figura 4.14 – Em (A): perfil de variância das linhas da imagem (sem normalização). Em (B): perfil de variância das colunas da imagem (sem normalização).....	39
Figura 4.15 – Em (C) o diagnóstico nebuloso para os pontos normalizados selecionados em (A). Em (D) o diagnóstico nebuloso para os pontos normalizados selecionados em (B).....	41
Figura 4.16 – Resultado da segmentação baseada em regras nebulosas	42
Figura 4.17 – Destaque da parte lisa, com a estampa numérica, e rugosa do tarugo.	42
Figura 4.18 – Possíveis orientações do tarugo	43
Figura 4.19 – Parâmetros do decisor baseado em blocos	43
Figura 4.20 – Primeira redução do espaço de busca.....	44
Figura 4.21 – Detecção de bordas na figura 4.20 utilizando o operador de Sobel e limiar automático	44
Figura 4.22 – Em (C) o diagnóstico nebuloso para os pontos normalizados selecionados em (A). Em (D) o diagnóstico nebuloso para os pontos normalizados selecionados em (B).....	45
Figura 4.23 – Redução do espaço de busca utilizando análise de perfil de variâncias e lógica nebulosa	46
Figura 4.24 – Mapa de cores JET utilizado	46
Figura 4.25 – Padrão coletado da figura 4.21, escolhido para exemplificar o mapeamento dos valores de correlação em tons de cores.	47
Figura 4.26 – Em (A): imagem original onde se deseja procurar algum padrão de interesse. Em (B): Matriz de correlação cruzada resultante do processamento com o padrão da figura 4.25 e em seguida mapeada em cores.....	48
Figura 4.27 – Estratégia de seleção de picos de correlação superiores a um limiar previamente configurado e que possuam uma distância $2r$ mínima entre eles.....	48
Figura 4.28 – Padrão da figura 4.25 binarizado com o método de OTSU.	49
Figura 4.29 – Em (A): imagem resultante da binarização global utilizando o método proposto por Otsu [10]. Em (B): imagem resultante da binarização local proposta por Niblack [11] com parâmetros ($N = 31$ e $\text{weight} = -0.8$).	49
Figura 4.30 – Em (A): imagem resultante da correlação cruzada normalizada e mapeada em cores do padrão da figura 4.28 com a imagem da figura 4.29-A. Em (B): imagem resultante da correlação cruzada normalizada e mapeada em cores do padrão da figura 4.28 com a imagem da figura 4.29-B.	50
Figura 4.31 – Imagem resultante da binarização local proposta por Niblack [11] com parâmetros ($N = 200$ e $\text{weight} = -0.8$).	50
Figura 4.32 – Imagem resultante da binarização local proposta por Niblack [11] com parâmetros ($N = 200$ e $\text{weight} = -0.8$) após redução no espaço de busca.....	51
Figura 4.33 – Comparativo do desempenho da média geométrica versus média aritmética.	53

Figura 4.34 – Exemplo de reconhecimento e cálculo da confiança global do sistema para a ordem de produção identificada.....	54
Figura 4.35 – Diagrama (resumido) de seqüência UML do software planejado.....	57
Figura 5.1 – Em (A): Enquadramento da cena. Em (B): Enquadramento da face do tarugo. Em (C): Enquadramento da região numérica. OBS: procedimentos feitos manualmente.....	59
Figura 5.2 – Em (A): Enquadramento da cena. Em (B): Enquadramento da face do tarugo. Em (C): Enquadramento da região numérica. OBS: procedimentos feitos manualmente e peça pintada como “B”(base).....	60
Figura 5.3 – Tarugo da ordem de produção 0741626 com pintura amarela em sua face	61
Figura 5.4 – Exemplos de imagens com rotação corrigida (rotação com interpolação bilinear).....	62
Figura 5.5 – Recorte resultante da metodologia com erro desprezível (ACERTO).....	63
Figura 5.6 – Recorte resultante da metodologia com área	63
Figura 5.7 – Recorte resultante da metodologia com área	64
Figura 5.8 – Detecção de bordas usando operador de Sobel e limiar automático (ACERTO).....	64
Figura 5.9 – Perfil de variância das linhas horizontais da imagem	65
Figura 5.10 – Diagnóstico nebuloso indicando limites	65
Figura 5.11 – Detecção de bordas usando operador de Sobel [12] e limiar automático (ERRO).....	65
Figura 5.12 – Perfil de variância das linhas horizontais da imagem	66
Figura 5.13 – Diagnóstico nebuloso indicando limites	66
Figura 5.14 – Detecção de bordas usando operador de Sobel [12] e limiar automático (ERRO).....	66
Figura 5.15 – Perfil de variância das linhas verticais da imagem	67
Figura 5.16 – Diagnóstico nebuloso indicando limites	67
Figura 5.17 – Detecção de bordas na FIG.-4-C usando operador de Sobel [12] e limiar automático (eixos coordenados conforme convenção ditada pela varredura de vídeo) .	67
Figura 5.18 – Perfil de variância das linhas verticais da imagem	68
Figura 5.19 – Diagnóstico nebuloso indicando limites	68
Figura 5.20 – Imagem já devidamente rotacionada e recortada.....	69
Figura 5.21 – Perfil de variância das linhas horizontais.....	69
Figura 5.22 – Diagnóstico nebuloso indicando limites da região numérica.....	69
Figura 5.23 – Imagem já devidamente rotacionada e recortada.....	70
Figura 5.24 – Perfil de variância das linhas horizontais.....	70
Figura 5.25 – Diagnóstico nebuloso indicando limites da região numérica.....	70
Figura 6.1 – Resultado de etapas consecutivas com confiança 0,95. Gráfico representa $0,95^n$, onde n é o número de etapas	73
Figura 6.2 – Padrões coletados na LIT (imagens com zoom manual da câmera) e sua imagem resultante da binarização local proposta por Niblack [11] com parâmetros (N = 31 e weight = -0.8).....	74
Figura 6.3 – Imagem coletada com a presença de ruído intenso.....	76
Figura 6.4 – Tarugo posicionado atrás de outro, impedindo a sua visualização	76
Figura 6.5 – Ocorrência de estampagem em local indevido.....	76
Figura 6.6 – Detecção de bordas com SOBEL e limiarização automática.....	77
Figura 6.7 – Linhas detectadas por Hough (tamanho mínimo das linhas escolhidas = 20 pixels)	78

LISTA DE TABELAS

Tabela 2.1 – Sensibilidade da NCC a alterações de escala sem inserção de ruído	12
Tabela 3.1 – Características técnicas dos tarugos.....	22
Tabela 3.2 – Tamanhos possíveis de bitolas dos tarugos	23
Tabela 4.1 – Evolução do tempo de processamento com o aumento do número de padrões por dígito	52
Tabela 5.1 – Acertos da metodologia para cada grupo de imagens	63
Tabela 5.2 – Acertos da metodologia para cada grupo de imagens	68
Tabela 5.3 – Acertos da metodologia para cada grupo de imagens	68
Tabela 5.4 – Acertos da metodologia para cada grupo de imagens	70
Tabela 5.5 – Número de imagens úteis por OP	71
Tabela 5.6 – Acertos da metodologia para cada grupo de imagens	72
Tabela 6.1 – Correlação entre padrões reais coletados.....	75

1 – INTRODUÇÃO

Existe um desejo constante do ser humano em conseguir mimetizar tarefas triviais do seu cotidiano, em um conjunto de técnicas ou regras que possam ser implementadas em ambiente computacional. Uma tarefa interessante de ser estudada é a capacidade visual desenvolvida não somente pela raça humana, mas também por uma grande quantidade de seres vivos. Trazer tal tarefa para ambiente computacional não é simples, pois equivale a tentar definir regras ou procedimentos detalhados do complexo processo de visão.

As técnicas que se relacionam de alguma maneira com o interesse pela extração de informação proveniente de imagens, utilizando o computador, podem ser agrupadas em uma categoria do conhecimento denominada *processamento digital de imagens*.

O presente trabalho dá continuidade a um projeto de pesquisa e desenvolvimento realizado em parceria entre a UFMG e a GERDAU AÇOMINAS S.A. Tal projeto visa identificar caracteres alfa-numéricos estampados em baixo relevo em blocos metálicos denominados “tarugos”. De acordo com especificações feitas em [2], o sistema pode utilizar técnicas de processamento de imagens em conjunto com uma interação entre dispositivos de vídeo e sistemas de automação, visando facilitar a inspeção dos tarugos e detectar possíveis misturas de materiais de composições diferentes. Estudos de técnicas e propostas de abordagens para o problema já foram feitas em ambiente MATLAB por Ramalho [1]. Foram estudadas novas técnicas para melhorar o processo de reconhecimento, além de uma transposição de ambiente das rotinas do MATLAB para um protótipo desenvolvido em C++, utilizando a MFC (*Microsoft Foundation Classes*).

1.1 – Motivação do projeto

Na literatura existem vários exemplos de aplicações para análise de imagem cujo foco é a identificação de padrões em diversas situações, como o reconhecimento numérico em placas de carros. O reconhecimento numérico para o caso industrial do sistema de inspeção da AÇOMINAS pode ser considerado singular, pois envolve diversas dificuldades, tais como qualidade da imagem, qualidade da estampagem, tempo reduzido desejado para o processamento, etc. O projeto tem vital importância, pois além de fornecer uma inspeção de tarugos mais determinística poderá agilizar o processo produtivo.

1.2 – Especificações do projeto

O sistema de identificação de tarugos fará parte de uma nova linha de inspeção de material. Esse sistema será utilizado para identificar, por meio de imagem coletada por câmera, a *ordem de produção* (OP) estampada nos tarugos que estão entrando na *linha de inspeção de tarugos* (LIT). Essa etapa é de vital importância no processo, pois, através da leitura dos caracteres alfa-numéricos estampados no tarugo, será feito todo o ajuste das máquinas da linha de inspeção. Além disso, é de total importância que materiais de ordens de produção diferentes não venham a ser misturados. O sistema também garantirá tal condição, impedindo que tarugos de ordens de produção diferentes daquela configurada no momento pelo sistema de automação venham a prosseguir pela linha e sejam empacotados juntos.

1.2.1 – Dificuldades

As características que tornam o processo interessante podem ser resumidas nos seguintes itens:

- Posicionamento dos tarugos em frente à câmera:
 - Variações de distância do tarugo em relação à aproximação frontal da câmera (influenciam no ajuste de foco);
 - Variações de posicionamento do tarugo no eixo lateral (podendo não parar no ponto exato do ângulo de visão da câmera);
 - Proximidade dos tarugos – os tarugos podem vir lado a lado, sem espaço entre eles, o que pode dificultar o processo de detecção do número da OP e sua identificação;
 - Rotação dos tarugos, podendo deixar os números da OP de ponta cabeça, de lado, ou na posição correta;

- Identificação dos caracteres estampados nos tarugos:
 - Rugosidade da superfície onde os números da OP são estampados;
 - Baixo contraste entre os números estampados e o fundo da imagem;
 - Estampagem nem sempre uniforme e nítida;
 - Alta similaridade entre os caracteres;
 - Tamanho dos caracteres impressos no tarugo (4 a 7 mm);
 - Tamanho da área da seção quadrada do tarugo (mínimo: 76x76mm / máximo: 160x160mm);
 - Exemplo de face de tarugo presente na figura 1.1.

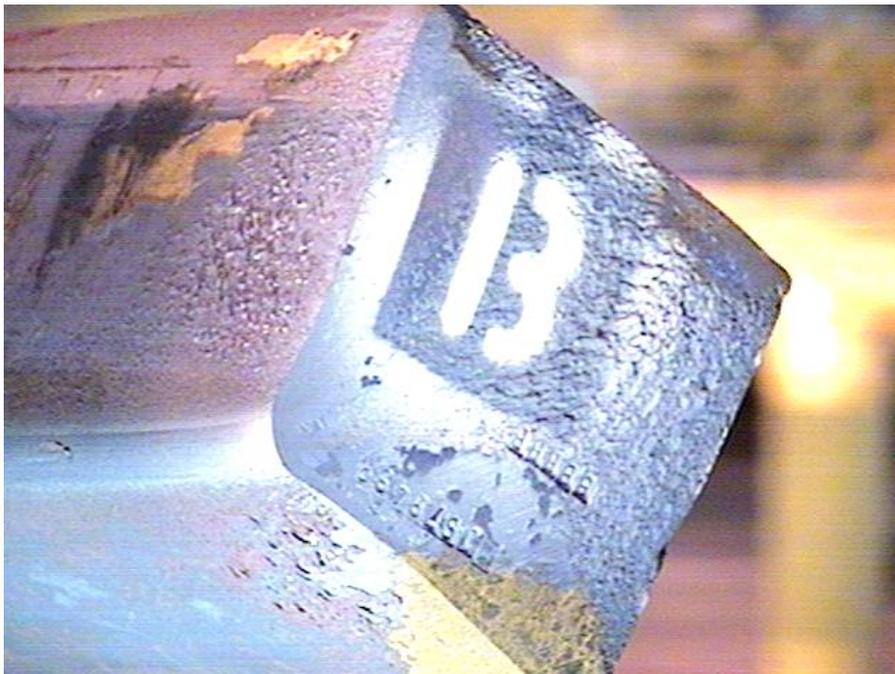


Figura 1.1 – Exemplo de tarugo estampado

- Ambiente agressivo:
 - Presença de vibração devido ao deslocamento dos tarugos;
 - Presença de poeira;
 - Possível interferência eletromagnética nos equipamentos de vídeo.
- Índice de acerto esperado:
 - Não pode haver mistura de tarugos de OP's diferentes, e nem confusão ao se informar o número da OP do primeiro tarugo de uma OP (pois as configurações da linha são dependentes dele). Portanto, o índice de acerto deve ser de 100%.

Uma descrição mais completa do sistema de estampagem numérica e inspeção será apresentada posteriormente no capítulo 3.

A GERDAU AÇOMINAS contactou diversas empresas a fim de tentar resolver o problema da identificação de caracteres para o presente caso, porém o trabalho foi rejeitado por todas em função das dificuldades impostas pelo processo. As metodologias e discussões desenvolvidas no presente trabalho, juntamente com as metodologias propostas por Ramalho [1] são resultantes da parceria com a UFMG, que foi a única instituição a aceitar o desafio.

1.3 – Contribuição da dissertação

Dentre as contribuições do presente trabalho, pode-se destacar:

- A continuidade de metodologias já estudadas anteriormente por Ramalho [1], de maneira a convertê-las para que sejam utilizáveis não somente em ambiente MATLAB, mas também em um programa que futuramente se comunicará com o nível de automação.
- Propostas de melhorias a respeito das instalações e equipamentos necessários para um bom desempenho do sistema de software. Propostas viáveis para o processo de maneira que no ponto de coleta de imagem exista um condicionamento tal que o desempenho do sistema seja determinístico.
- Metodologias de organização de código como a orientação a objetos torna o sistema reutilizável, além de dar suporte a um desenvolvimento incremental sem trazer grande complexidade para os futuros desenvolvedores ou mantenedores.
- Desenvolvimento do sistema de modo que todos os processamentos computacionalmente complexos pudessem ser realizados no menor tempo possível.
- Realizar um estudo aprofundado de técnicas de processamento de imagens, sendo que este material (know how) será disponibilizado para a GERDAU AÇOMINAS utilizar também em outros trabalhos.
- Realizar um estudo visando indicar direções para melhorias no processo, possibilitando o uso do sistema de identificação dos números estampados e apontando os locais onde se devem implantar melhorias:
 - Estampadeira de tarugos – evitando a estampagem falha e verificando regularmente a qualidade dos tarugos;
 - Tipo de caracteres alfa-numéricos - substituindo os existentes por caracteres com baixa similaridade entre si;
 - Continuar a explorar as técnicas de processamento de imagens para realçar mais a parte que é interessante do tarugo;
 - Colocações de sensores para fazer com que as peças parem sempre a uma mesma distância da câmera;
 - Definir um condicionamento de iluminação da cena adequado à identificação;
 - Estudar com a operação do processo possibilidades viáveis de modificações no processo, como a pintura de áreas restritas da face do tarugo.

Foi executado o desenvolvimento do software, o protótipo do programa analisador de imagens. Entretanto foi necessário estudar e implementar novas soluções ou técnicas de processamento de imagem. Tal medida foi tomada a fim de resolver novos problemas não estudados anteriormente. O critério tempo de processamento também pode ser considerado como fator essencial, e foi feito um largo estudo de

maneira a ser possível executar todos os processamentos computacionalmente complexos no menor tempo possível.

1.4 – Organização do trabalho

O Capítulo 1 descreve de maneiras gerais os objetivos, dificuldades e contribuições da tese.

No Capítulo 2 são descritas as técnicas de processamento de imagens utilizadas durante o desenvolvimento do sistema. Durante a análise são expostos os prós e contras de cada abordagem estudada.

No Capítulo 3, está presente a descrição do processo de inspeção de tarugos da AÇOMINAS, para o qual o sistema desenvolvido será aplicado.

No Capítulo 4, é descrito o sistema de visão computacional, contemplando tanto a parte de *hardware* quanto de *software*. Na parte relacionada ao *hardware*, são destacados cada componente necessário ao sistema, seja ele efetivamente utilizado ou meramente especificado para melhorias futuras. Na parte relacionada ao *software*, são demonstradas as metodologias convertidas para o protótipo on-line e, além disso, são propostas e especificadas novas metodologias em MATLAB para auxiliar o reconhecimento.

O Capítulo 5 contém os resultados off-line das metodologias para imagens coletadas no ambiente industrial.

Finalmente, o Capítulo 6 contém as conclusões e propostas de continuidade para o presente projeto.

2 – PROCESSAMENTO DIGITAL DE IMAGENS

Neste capítulo será feita uma breve revisão dos conceitos selecionados em [1] e utilizados no desenvolvimento do protótipo, além da adição de novas propostas para auxiliar o reconhecimento. O capítulo inicia com a apresentação dos métodos mais comuns de casamentos de padrões, técnicas para segmentação numérica em imagens e, além disso, inclui comentários a respeito da codificação e organização de um software utilizando programação orientada a objetos.

2.1 – Métodos de Limiarização (binarização)

2.1.1 – Otsu

O método de Otsu é uma forma de limiarização baseada na análise do histograma da imagem, que é uma representação gráfica da distribuição da ocorrência de cada valor de pixel. Uma das vantagens é apresentar característica não paramétrica [10]. O histograma da imagem é utilizado para estimar a função probabilidade dos valores de intensidade dos pixels. Em um problema de duas classes, caracterizado por um histograma bimodal, o objetivo é achar o limiar que minimize a variância intraclasse, resultando em regiões homogêneas [10].

Dada a imagem I de tamanho $M \times N$, a imagem binária Im_{bin} será dada por:

$$Im_{bin}(i, j) = \begin{cases} 0, & \text{se } I(i, j) \leq T_h \\ 255, & \text{se } I(i, j) > T_h \end{cases}$$

Onde:

$i = 1 \dots M$;

$j = 1 \dots N$;

$T_h \rightarrow$ limiar calculado;

O método de Otsu sempre irá produzir dois clusters, e em seguida um teste é efetuado para se saber se a região é constituída de uma ou duas classes. Este teste é efetuado calculando-se a diferença entre as médias dos dois clusters. Se o módulo dessa diferença (distância) for superior ao valor θ então a região consistirá de duas classes. Caso contrário, a região consiste de apenas uma classe.

Na figura 2.1 pode ser verificado um histograma representando a distribuição dos pixels em uma determinada imagem. O método de OTSU visa encontrar o melhor valor de separação das classes a fim de promover uma binarização posterior.

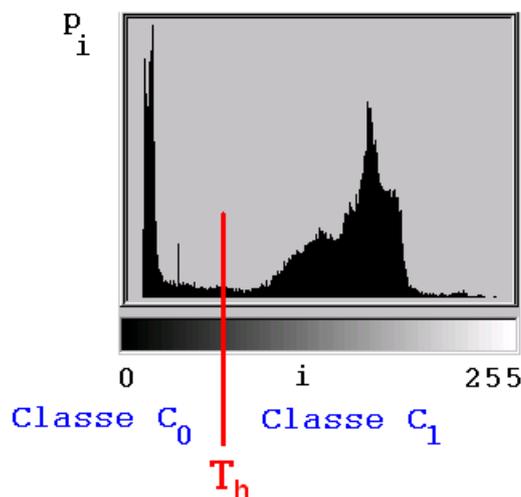


Figura 2.1 – Limiarização (T_h) automática proposta por OTSU

2.1.2 – Niblack

Este método foi proposto por Niblack em 1986 [11] e é largamente utilizado em algoritmos de limiarização local [1,14,16]. Consiste em deslocar uma janela sobre a imagem original, calculando os limiares localmente. Considerando uma janela de dimensão $N \times N$ pixels e uma imagem I , o limiar é calculado segundo a equação:

$$L = \mu(i, j) + k \cdot \sigma(i, j) \quad (1)$$

Onde:

$i = 1, \dots, N$;

$j = 1, \dots, N$;

$\mu(i, j)$ – média da janela espacial com $i = 1, \dots, N$ e $j = 1, \dots, N$.

$\sigma(i, j)$ – desvio padrão da janela espacial com $i = 1, \dots, N$ e $j = 1, \dots, N$.

k – fator de peso, parâmetro do método.

Os parâmetros para essa técnica são o tamanho N , em pixels, da janela e o peso k . O tamanho da janela deve ser uma solução de compromisso, uma vez que pequenas dimensões preservam detalhes, enquanto grandes reduzem ruídos. O valor do peso k atua principalmente nas bordas dos objetos, nas quais há pixels de transição, contendo valores intermediários entre o objeto e o fundo da imagem. O peso define quanto da borda do objeto será agregada à imagem limiarizada [1].

2.2 – Detecção de bordas (SOBEL)

A borda de um objeto pode ser denominada como o conjunto de pixels que delimitam duas regiões com distintos valores de luminância [14]. Existem duas abordagens básicas para a detecção de bordas [14]: detecção por diferenciação e

detecção baseada em modelos. A maioria dos métodos propostos e utilizados na literatura são compostos por operadores de diferenciação [14]. Por esse motivo, o estudo conduzido nesta seção demonstrará os princípios e ferramentas dessa abordagem.

A aplicação dos métodos de diferenciação é realizada pelo gradiente para derivadas primeiras e pelo laplaciano para derivadas segundas [14]. Contudo, o cálculo do gradiente fornece melhor desempenho, especialmente em imagens contendo ruído [14]. Por esse motivo, são propostos na literatura diversos métodos para a aproximação do gradiente discreto, através do cálculo das diferenças entre pixels adjacentes [14]. As técnicas definem janelas espaciais que são convoluídas com a imagem original, fornecendo uma aproximação do cálculo do gradiente para cada pixel.

Com relação à direção do cálculo do gradiente, as técnicas podem ser: ortogonais (realizam o cálculo nas direções verticais e horizontais) ou multi-direcionais.

As técnicas multi-direcionais calculam o gradiente em várias direções, escolhendo, em cada pixel, o maior valor. Supondo duas janelas (G_x e G_y), uma para cada direção, e uma imagem I , o gradiente G é estimado [14]:

$$\mathbf{G}_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * \mathbf{I} \quad \text{and} \quad \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{I} \quad (2-1-a)$$

Em cada ponto na imagem, o resultado da aproximação do gradiente na imagem pode ser combinado para obter a magnitude do gradiente:

$$\mathbf{G} = \sqrt{\mathbf{G}_x^2 + \mathbf{G}_y^2} \quad (2-1-b)$$

O operador de Sobel é uma técnica ortogonal e apresenta maior sensibilidade a contornos diagonais do que a contornos verticais e horizontais [12].

2.3 – Correlação Cruzada Normalizada Bidimensional (NCC)

Uma medida de similaridade muito utilizada em processamento de imagens é a correlação cruzada normalizada [1]. Nesse método, o padrão é deslocado pela imagem, e, em cada posição, é multiplicado pelos valores da imagem nessa região. O somatório dessa multiplicação é armazenado para cada posição de deslocamento. O deslocamento do padrão sobre a imagem inclui as posições em que ele se encontra parcialmente imerso na imagem, que para efeitos práticos não gera resultados válidos para a análise, sendo que a informação só deve ser analisada na região onde o padrão encontra-se completamente imerso na imagem.

Considerando uma imagem I de dimensão $M \times N$ pixels e um padrão P de dimensão $K \times J$ pixels, a correlação cruzada é definida como:

$$R_{IP}(m,n) = \sum_k \sum_j I(i,j)P(k-m,j-n)$$

$$m = 0,1,\dots,M-1$$

$$n = 0,1,\dots,N-1$$

$$j = 0,1,\dots,J$$

$$k = 0,1,\dots,K$$
(2-2)

A matriz resultante da correlação cruzada possuirá então o tamanho $(M+K-1) \times (N+J-1)$. Na matriz resultante, valores altos podem significar similaridade com o padrão. Entretanto, para facilitar a análise e evitar valores altos em regiões de alto brilho, é necessário realizar uma normalização.

Uma possível normalização é dividir o resultado pela média geométrica entre a autocorrelação do padrão e a autocorrelação da região da imagem I que está sendo analisada no momento. Essa forma de normalização pode ser definida como:

$$N(m,n) = \frac{\sum_k \sum_j I(k,j)P(k-m,j-n)}{\sqrt{\sum_k \sum_j I^2(k,j) \sum_k \sum_j P^2(k-m,j-n)}}$$
(2-3)

A normalização proposta por Russ [14], evidenciada na EQ. 2-3, gera valores no intervalo de 0 a 1, sendo que correlação 0 indica a correlação do “complemento” da imagem e correlação 1 indica correlação direta máxima. Na figura 2.2 pode ser verificado um exemplo desse processamento, onde as regiões da matriz onde não foram inseridos valores indicam elementos que podem ser ignorados, pois são resultados de operações quando o padrão ainda não se encontra completamente imerso na imagem.

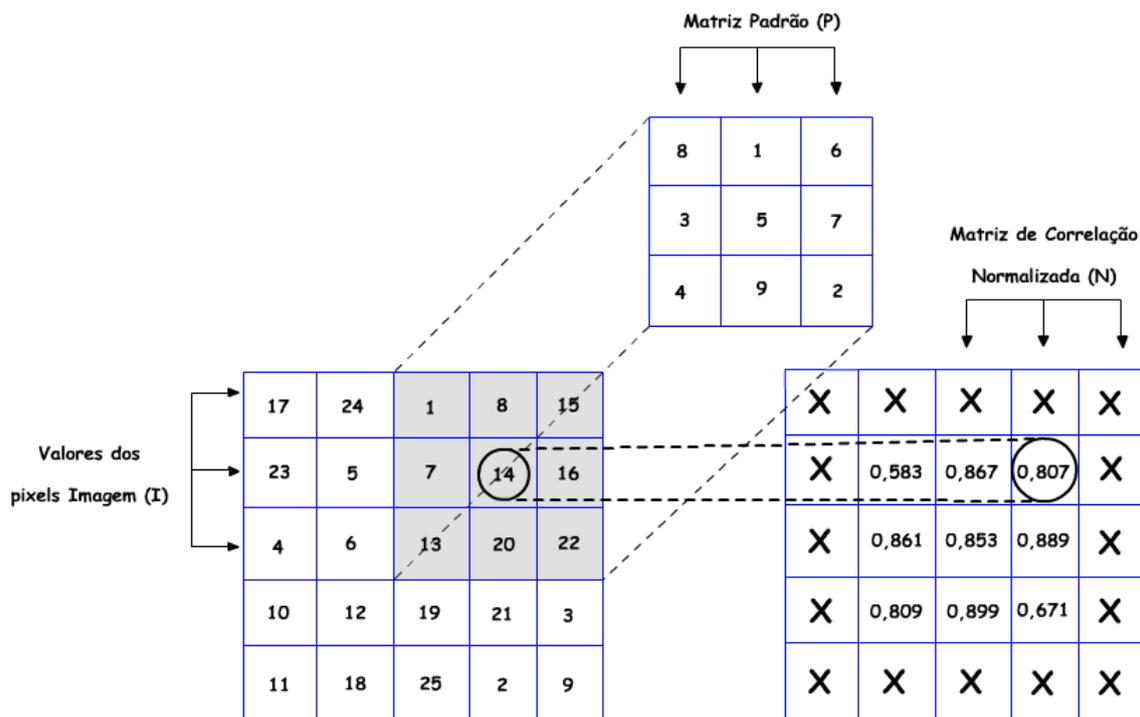


Figura 2.2 – Operação de correlação normalizada

Na figura 2.3 pode ser visto um exemplo de cálculo da correlação normalizada, os valores são referentes à figura 2.2.

autocorr_Ip_2D	=	1*1+8*8+15*15+7*7+14*14+16*16+13*13+20*20+22*22	=	1844
autocorr_P_2D	=	8*8+1*1+6*6+3*3+5*5+7*7+4*4+9*9+2*2	=	285
xcorr_2D	=	1*8+8*1+15*6+7*3+14*5+16*7+13*4+20*9+22*2	=	585
N(2,4) = xcorr_2D / sqrt(autocorr_Ip_2D*autocorr_P_2D)				
N(2,4) = 585/sqrt(1844*285) = 0,8070				

Figura 2.3 – Exemplo de cálculo de correlação

A normalização indicada pela EQ. 2-3 é sensível a variações de média ao longo da imagem: se for alterada, por exemplo, a média do padrão P ou da imagem I, o resultado da correlação é modificado, mesmo em um caso em que anteriormente a correlação máxima fosse 1. Diante desse fato, Gonzalez [22] propõe uma modificação no cálculo para tornar o resultado mais robusto, retirando a média local da imagem I e também do padrão antes de se realizar a correlação:

$$N(m,n) = \frac{\sum_k \sum_j [I(k,j) - \overline{I(k,j)}] \cdot [P(k-m, j-n) - \overline{P}]}{\sqrt{\sum_k \sum_j [I(k,j) - \overline{I(k,j)}]^2 \sum_k \sum_j [P(k-m, j-n) - \overline{P}]^2}} \quad (2-4)$$

A correlação cruzada normalizada proposta por [22] e visualizada na equação 2-4, gera resultados no intervalo de -1 a 1, foi sugerida por Ramalho [1] para o problema do presente estudo, corresponde à metodologia de correlação que foi implementada. O cálculo da correlação cruzada normalizada tem o custo computacional proporcional ao quadrado do número de pontos ($O(n^2)$). Implementações atuais utilizam artifícios baseados no domínio de Fourier, onde os somatórios de correlação são reduzidos a um simples produto matricial, fazendo com que tanto o tempo de processamento quanto o número de operações sejam reduzidos ($O(n \cdot \log(n))$). Atualmente, existem bibliotecas específicas para o cálculo eficiente da FFT (Fast Fourier Transform), no caso da FFTW [6] é da ordem de $O(n \cdot \log(n/C))$ onde C é o tamanho ideal da memória cache no computador utilizado ($C < n$). É necessário observar que, se o padrão possuir uma dimensão muito “pequena” (só é possível determinar essa dimensão com análises comparativas de desempenho), o cálculo da correlação no domínio temporal torna-se mais eficiente do que no domínio da frequência.

Para exemplificar, considere o padrão mostrado na figura 2.4 e a imagem da figura 2.5. A figura 2.6 apresenta o resultado da correlação cruzada normalizada entre o padrão e a imagem. Tal operação retorna uma matriz de coeficientes que foram mapeados em cores para facilitar sua análise. A região com tonalidades avermelhadas indica a localização da estrutura na imagem mais parecida com o padrão procurado.



Figura 2.4 – Padrão escolhido

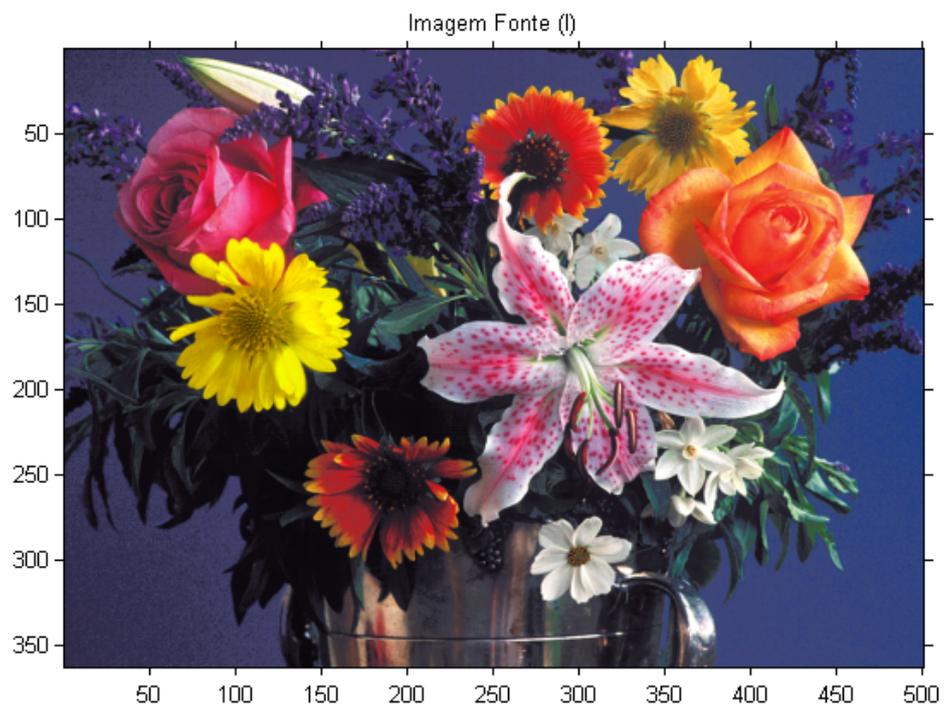


Figura 2.5 – Imagem a ser analisada

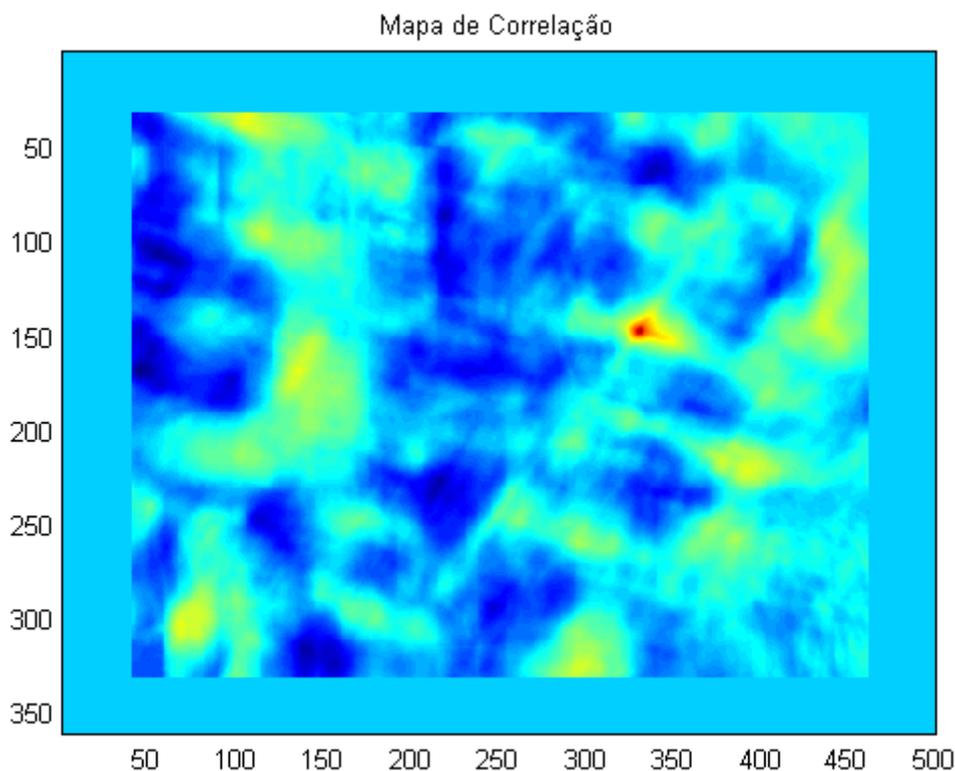


Figura 2.6 – Operação de correlação cruzada normalizada (região azul-clara indicando valores ignorados)

Uma deficiência da NCC é a sua sensibilidade à escala, ou seja, um padrão, se escalonado por um fator de crescimento ou redução de suas dimensões, pode afetar significativamente o reconhecimento na imagem. Tal problema relacionado à escala pode forçar a utilização de um banco de padrões com escalas diferentes para se evitar reconhecimentos indevidos.

Na Tabela 2.1 são apresentados valores da NCC em função da alteração das dimensões do padrão, sofrendo um redimensionamento utilizando interpolação bicúbica. A escala do redimensionamento, o valor do máximo da NCC, e a coordenada desse máximo podem ser analisados, observando que, para pequenas alterações de escala (faixa de +- 5%), o valor máximo da correlação não se altera significativamente a ponto de comprometer o reconhecimento.

Tabela 2.1 – Sensibilidade da NCC a alterações de escala sem inserção de ruído

Escala de P	Máximo da NCC	Coordenadas do máximo (x,y)
1,500000	0,413978	(323,136)
1,475000	0,427990	(324,137)
1,450000	0,446917	(324,137)
1,425000	0,457892	(325,138)
1,400000	0,474670	(325,139)
1,375000	0,488619	(325,140)
1,350000	0,495221	(325,140)

1,325000	0,508723	(325,141)
1,300000	0,525059	(326,141)
1,275000	0,538671	(326,142)
1,250000	0,553639	(326,143)
1,225000	0,570162	(327,143)
1,200000	0,579622	(327,143)
1,175000	0,598851	(328,144)
1,150000	0,631664	(328,145)
1,125000	0,651640	(329,146)
1,100000	0,700195	(331,147)
1,075000	0,779538	(331,147)
1,050000	0,854329	(331,147)
1,025000	0,939699	(330,147)
1,000000	1,000000	(330,148)
0,975000	0,897987	(330,148)
0,950000	0,782111	(330,149)
0,925000	0,724352	(332,149)
0,900000	0,672681	(334,147)
0,875000	0,643784	(335,147)
0,850000	0,621969	(336,147)
0,825000	0,596446	(334,152)
0,800000	0,563013	(334,153)
0,775000	0,547428	(335,152)
0,750000	0,548546	(342,143)
0,725000	0,548954	(343,143)
0,700000	0,548408	(344,143)
0,675000	0,548611	(345,143)
0,650000	0,549322	(345,143)
0,625000	0,552172	(346,143)
0,600000	0,551936	(347,142)
0,575000	0,551755	(86,296)
0,550000	0,564903	(87,297)
0,525000	0,569591	(350,142)
0,500000	0,573129	(88,297)

2.4 – Segmentação numérica

O objetivo da segmentação é particionar imagens em regiões. O problema da segmentação numérica, em processamento de imagens, pode ser definido como qualquer tarefa em que se deseja extrair regiões numéricas de uma imagem qualquer [15]. Também pode ser encarada como segmentação numérica a separação de dígitos de uma seqüência numérica já encontrada.

Muitas metodologias para resolver o problema de segmentação numérica podem ser encontradas na literatura focando a aplicação em reconhecimentos de placas de veículos como pode ser visto em [16]. A seguir são descritas algumas metodologias que de alguma forma podem fornecer soluções para o presente trabalho.

2.4.1 – Projeção dos pixels

Uma estratégia bastante utilizada em problemas de separação de caracteres em uma seqüência numérica é a metodologia de projeção ou de somatório dos pixels, tanto na direção horizontal como na vertical [16]. Em uma imagem binária é arbitrado que os pixels pretos possuem valor 1 e os brancos possuem valor 0. O somatório delimita regiões onde o número e os dígitos fossem localizados. A figura 2.7 mostra uma seqüência numérica tomada como exemplo em um problema de detecção de regiões numéricas em placas de carros [16].

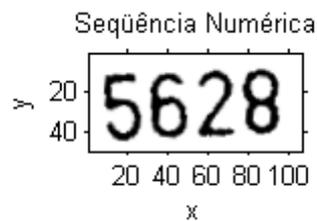


Figura 2.7 – Exemplo Seqüência Numérica

O perfil horizontal pode ser definido como um vetor contendo os somatórios de cada linha da imagem. O perfil vertical, analogamente, pode ser definido como o vetor contendo o somatório dos pixels em cada coluna.

A figura 2.8 contém informações de onde a seqüência numérica se encontra, aproximadamente entre o pixel 8 e o 45 no eixo Y. A figura 2.9 indica a região no eixo X onde se encontra cada dígito, sendo evidenciados 4 dígitos aproximadamente igualmente espaçados. O resultado do processamento, que apesar de parecer simples, pode ser muito eficiente em casos sem muito ruído na imagem binária. Casos com muito ruído podem dificultar a determinação exata da separação entre os dígitos.

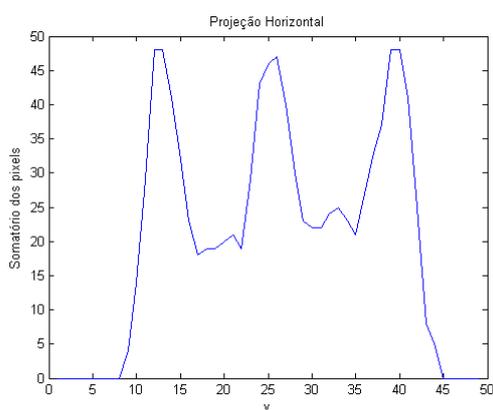


Figura 2.8 – Somatório dos pixels de cada linha da imagem binária da figura 2.7

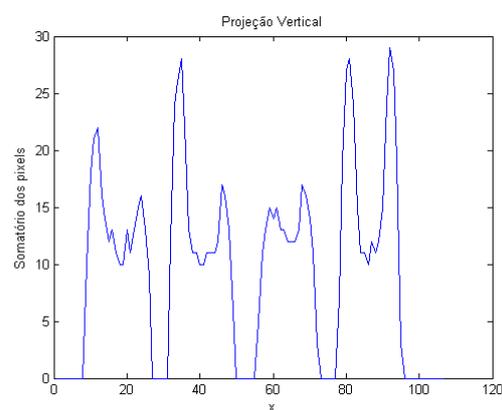


Figura 2.9 – Somatório dos pixels de cada coluna da imagem binária da figura 2.7

2.4.2 – Análise de perfil de variância

A análise pode ser feita tanto para identificar dígitos, como na análise de projeção de pixels, como também para identificar diferença de rugosidades entre regiões em uma imagem.

Um método comum para estimar a variância é:

$$Var = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 \quad (2-5)$$

Onde :

n – número de amostras no vetor;

\bar{x} – valor médio das amostras.

As Figuras 2.10 e 2.11 apresentam resultado semelhante ao da análise de projeção de pixels para o exemplo da separação de dígitos.

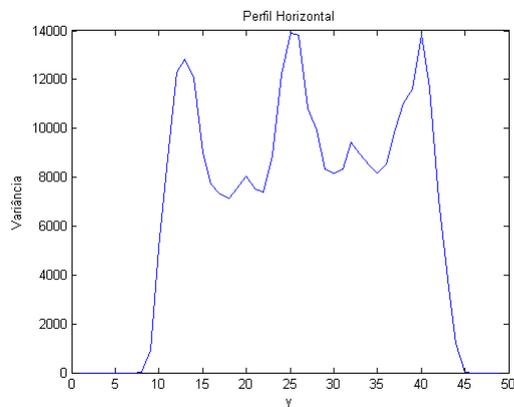


Figura 2.10 – Perfil de variância horizontal da figura 2.7

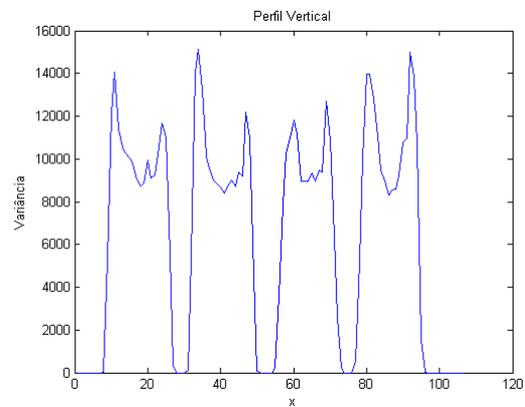


Figura 2.11 – Perfil de variância vertical da figura 2.7

A figura 2.12 trata de um exemplo de aplicação onde é inserida uma região com ruído aleatório para simular rugosidade em uma imagem predominantemente lisa.

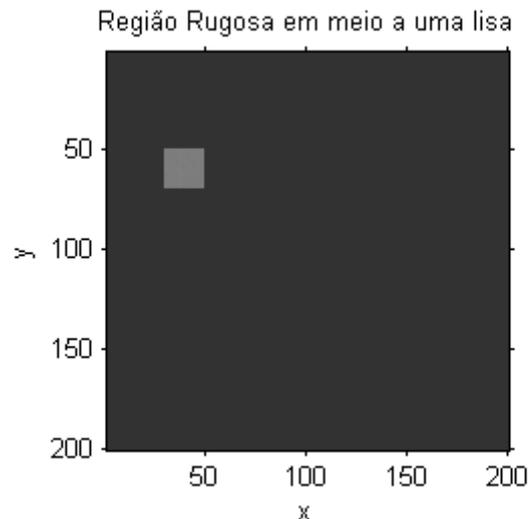


Figura 2.12 – Inserção de uma região com rugosidade em imagem lisa

Segundo as Figuras 2.13 e 2.14, pode-se perceber que a análise da variância indica a localização da região rugosa. É interessante notar que a análise de variância tende a identificar regiões rugosas, mesmo que tenham valores de tonalidades muito próximos, o que não ocorre quando se utiliza a técnica da projeção de pixels.

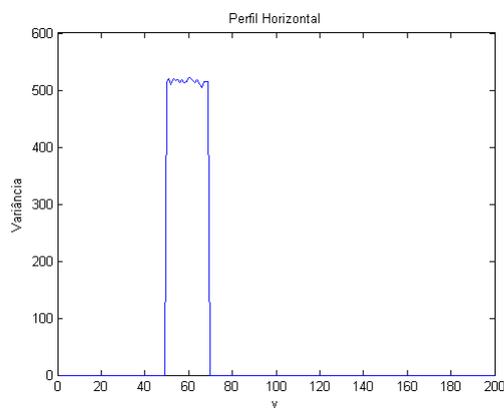


Figura 2.13 – Perfil de variância horizontal da figura 2.12

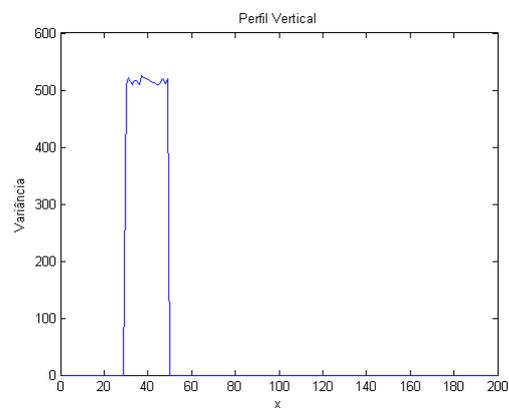


Figura 2.14 – Perfil de variância vertical da figura 2.12

2.4.3 – Transformada de Hough

A transformada de Hough é particularmente útil no processamento de uma descrição global de uma característica (onde o número de classes soluções precisa ser conhecido *a priori*), dadas algumas medições locais (possivelmente contendo ruído). A motivação por trás da idéia da transformada de Hough para detecção de linhas é que cada valor de entrada (ex: um ponto coordenado) indica sua contribuição para uma solução global consistente (ex: a linha física que passa por aquele ponto).

Como um simples exemplo, considere um problema comum de ajuste de segmentos de reta em um conjunto discreto de pontos em uma imagem (ex: resultado

proveniente de um operador de detecção de bordas). A figura 2.15 demonstra algumas possíveis soluções para este problema. A carência de um conhecimento *a priori* sobre o número de segmentos de reta desejados (e a ambigüidade sobre o que constitui um segmento de reta) coloca o problema sobre várias restrições.

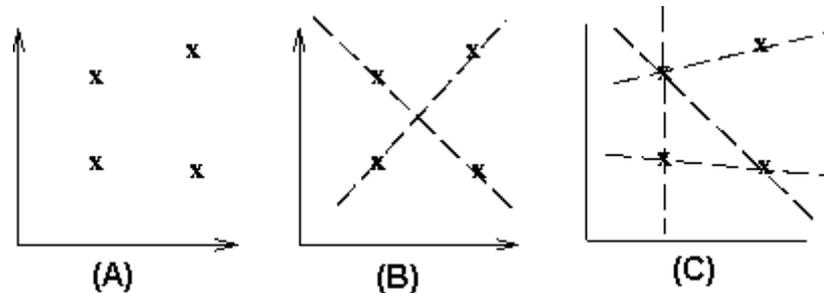


Figura 2.15 – Em (A) Pontos candidatos. Em (B) e (C) possíveis retas que cruzam os pontos

É possível, analiticamente, descrever um segmento de reta de várias maneiras. Entretanto, uma equação conveniente para descrever uma reta utilizando uma noção *paramétrica* ou *normal* (figura 2.16) é:

$$x \cdot \cos \theta + y \cdot \sin \theta = r \quad (2-6)$$

Onde:

r → é o tamanho de um segmento normal que vai da origem (0,0) até o segmento de reta;

θ → é a orientação de r com respeito ao eixo X.

Observação: para qualquer ponto (x,y) presente na linha então r e θ são constantes.

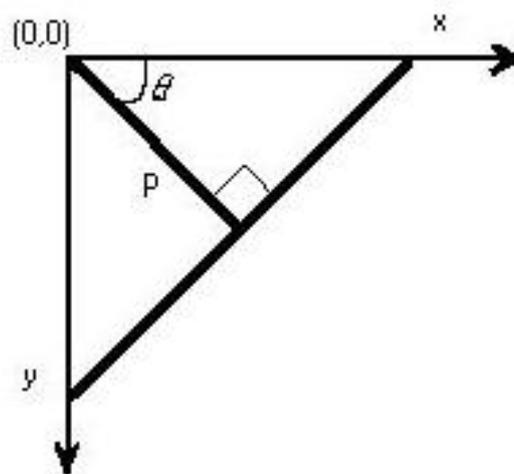


Figura 2.16 – Parametrização polar

Nas Figuras 2.17 e 2.18, podem ser vistos os resultados de um processamento em que se deseja encontrar segmentos de retas em uma imagem resultante de uma operação de detecção de bordas. Fazendo-se uma análise criteriosa, vê-se que foram

identificadas 10 retas, estando muitas delas sobrepostas. Os exemplos anteriores demonstram a utilidade da transformada de Hough e também expõem algumas de suas fragilidades, como a necessidade de um pós-processamento para se fazer a escolha adequada das retas.



(A)

(B)

Figura 2.17 – Em (A): Exemplo de detecção de bordas utilizando operador SOBEL. Em (B): Exemplo de detecção de linhas utilizando a transformada de HOUGH (número de retas desejadas = 10).

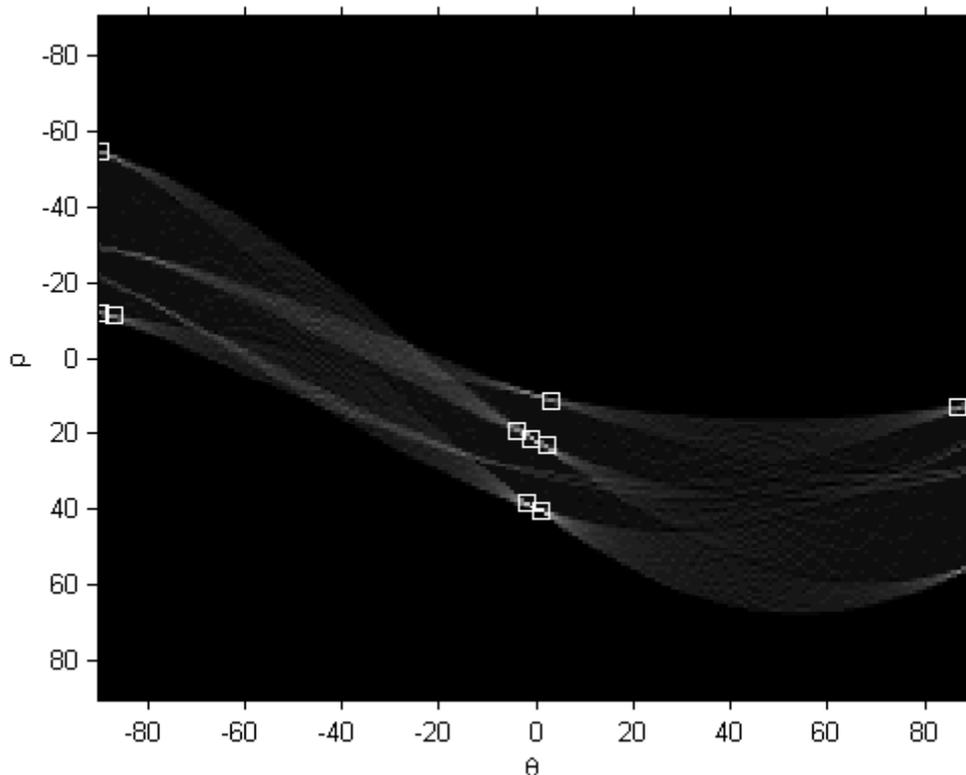


Figura 2.18 – Resultados de ρ e θ evidenciando o resultado da transformada de Hough selecionando os 10 picos mais evidentes.

Um exemplo de aplicação da transformada de Hough pode ser visualizado na figura 2.19, onde os seus resultados são utilizados para escolher regiões candidatas ao reconhecimento de numerações estampadas em placas veiculares. A partir de resultados de processamentos de detecção de bordas é aplicada a transformada visando encontrar

retas paralelas que definam retângulos ou paralelogramos, é então o primeiro passo para redução de espaço de busca obtendo regiões candidatas.

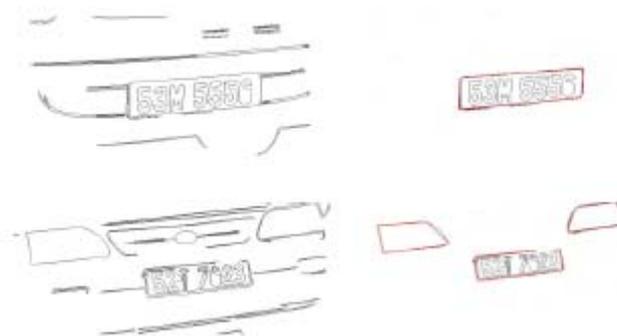


Figura 2.19 – Combinação de algoritmos de contorno e transformada de Hough

2.5 – Abordagens alternativas existentes na literatura

Uma alternativa que melhora o desempenho da identificação de caracteres são os correlacionadores baseados na transformada Mellin-Fourier [24]. Tal transformada possui características de invariância de seu comportamento em função da escala da imagem. Outra abordagem que é invariante à rotação da imagem é a análise do momento de Zernike [29]. A tentativa de desenvolver metodologias insensíveis à oclusão de objetos é outro tema que merece destaque [55].

2.6 – Conclusões

Na literatura, podem ser encontradas metodologias distintas para detecção de objetos em imagens [32,33,34,35,36,37,38,39,40,41,42,43,44,45]. O tema de pesquisa relacionado a reconhecimento de padrões também fornece alternativas para solução deste tipo de problemas [69,70,72], cabe identificar qual mais se identifica com o problema em questão. É interessante verificar que o foco da maioria das pesquisas se intensifica no que tange à busca de metodologias que possuam resposta invariantes em função de características como escala, rotação, iluminação.

Neste capítulo é apresentada uma breve revisão dos métodos de limiarização, detecção de bordas, e correlação cruzada sugeridos por Ramalho [1] para o problema de reconhecimento de caracteres estampados em tarugos de aço. Adicionalmente foram inseridos, pelo presente trabalho, metodologias que serão utilizadas como base para a tarefa de segmentação de objetos e/ou identificação da localização das estampas numéricas nas imagens.

3 – DESCRIÇÃO DA LINHA DE INSPEÇÃO DE TARUGOS DA AÇOMINAS

Os tarugos são de fundamental importância para a GERDAU AÇOMINAS, sendo, inclusive, um forte produto de exportação. Em função disso, a empresa dispõe de um rígido controle de qualidade, atendendo às rigorosas especificações do cliente. Os tarugos, antes de serem encaminhados aos clientes, são inspecionados na LIT (Linha de Inspeção de Tarugos). Nesse setor, são realizados o controle de qualidade e a destinação da produção para o cliente como pode ser resumido pela figura 3.1.

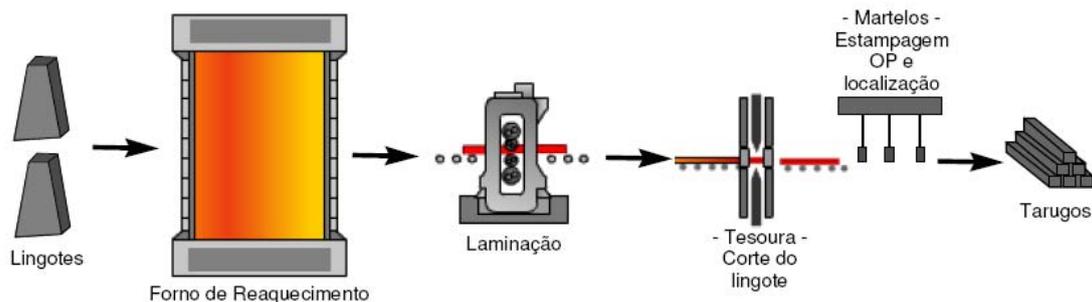


Figura 3.1 – Processo de produção de tarugos

Os tarugos são objetos de aço provenientes do processo de laminação de peças, denominadas “lingotes”, oriundas da Aciaria, com características próprias na composição química (especificadas pelo cliente). Essas peças passam por um processo de conformação mecânica, denominada “laminação”, até atingirem a bitola requerida pelo cliente, apresentando, no final do processo, grandes extensões. Como o comprimento também é uma especificação de produção, esses tarugos são, então, cortados por um equipamento chamado “tesoura pendular”. Os tarugos são as peças no fim deste processo (figura 3.2). Para haver controle da produção, os tarugos cortados são estampados com a ordem de produção (OP), número que designa as características químicas e as dimensões especificadas pelos clientes, gerada a partir da negociação entre o setor de vendas e o cliente. As estampas nos tarugos são marcadas após o corte por equipamentos chamados “martelos”, que estampam no aço quente o número da OP (figura 3.2). O primeiro e o último tarugo cortados recebem, além desse número, uma identificação chamada “localização”. A letra T (Topo) designa o primeiro tarugo e a letra B (Base) o último. As estampas de Topo (letra T) e Base (letra B) são impressas por martelos distintos. Alguns clientes requisitam que a localização, além de ser estampada, seja pintada. Como não é um padrão para todas as peças, a tinta pode ou não

aparecer nos tarugos, não podendo ser utilizada para identificação. Os tarugos intermediários não recebem nenhuma estampa adicional.



Figura 3.2 – Leito de resfriamento dos tarugos

A Gerdau Açominas produz tarugos laminados utilizados, principalmente, para a produção de fio-máquina, barras, e produtos forjados para uso automotivo. Produz também tarugos lingotados utilizados para produção de fio-máquina, e vergalhões. Tarugos lingotados são provenientes do lingotamento contínuo, não passando pela laminação primária. Apenas os tarugos laminados são inspecionados na LIT. Um exemplo de produto final gerado a partir de tarugos é o fio-máquina, que pode ser visualizado na figura 3.3.



Figura 3.3 – Bobinas de fio-máquina

Os tarugos são produzidos conforme padronizações adotadas pela GERDAU AÇOMINAS e podem ser verificadas pela tabela 3.1.

Tabela 3.1 – Características técnicas dos tarugos

Tipo	Dimensões (mm)	Principais Especificações Atendidas
Tarugos Laminados	- Seção: 76 x 76 a 160 x 160	Aços baixo, médio e alto carbono, baixa liga e ligados conforme as seguintes normas: - JIS G 3101, 3112, 3503, 3505, 3506, 3507, 4051 e 4105 - ABNT NBR 7480 - ASTM-A-36, A350, A572 e A588 - SAE-J-403 (1006 ~ 1095) e - SAE-J-404 - DIN EN 10025
	- Seção: 130 x 130 a 160 x 160 - Comprimento: 5.100 a 16.000	- SAE-J-403 (1006 ~ 1095) - JIS-G-3101, 3112, 3505 e 3506 - ASTM-A-36

Depois de produzidos e estampados, os tarugos são encaminhados para a etapa de resfriamento e estocagem, onde permanecem até serem inspecionados e destinados aos clientes. Os tarugos são conduzidos para a LIT, onde será realizado o controle de qualidade e a correta separação e empacotamento dos tarugos. O controle de qualidade é realizado por inspeções internas, por meio de ultra-som, e superficiais utilizando um equipamento que realça os defeitos nas faces do tarugo para detectar presença de defeitos na produção. Alguns tipos de defeitos podem ser reparados ainda na LIT, através de um equipamento chamado “esmerilhadeira” (para correção de defeitos superficiais) ou através de cortes (para correção de defeitos internos). Os tarugos cortados recebem outra estampagem no final do processo. Os tarugos reformados são, então, conduzidos para a empacotadeira, onde se juntam aos tarugos aprovados na LIT, para serem encaminhados para o cliente.

A figura 3.4 mostra uma imagem tirada de um tarugo já inspecionado e pronto para ser empacotado. A partir dessa imagem, pode-se notar uma característica presente em todos os tarugos, que é o fato de possuir uma região lisa e outra rugosa, sempre separadas pela diagonal. Tal característica é decorrente do processo de corte. Outro fato importante é notar que a impressão numérica **sempre** ocorre na parte lisa do tarugo. É importante destacar que os tarugos têm seção aproximadamente quadrada já previamente definida pela sua ordem de produção (OP), tais dimensões possíveis estão definidas na Tabela 3.2. Uma característica do processo, também relacionada ao tamanho pré-definido das seções (bitolas) dos tarugos, é o fato de que o tamanho da seqüência numérica não deve ultrapassar o tamanho mínimo de bitola estipulado pelo processo. O tamanho da seqüência numérica é então limitado a 76 mm, e em todos os

demais tipos de tarugos os caracteres são impressos da mesma maneira, com a mesma fonte e com o mesmo tamanho.



Figura 3.4 – Visão do tarugo já inspecionado

Tabela 3.2 – Tamanhos possíveis de bitolas dos tarugos

Bitola	Dimensão (mm x mm)
1	76x76
2	82x82
3	83x83
4	95x95
5	100x100
6	120x120
7	121x121
8	125x125
9	130x130
10	131x131
11	140x140
12	149x149
13	150x150
14	160x160

Atualmente, o sistema de inspeção funciona com intervenção do operador da linha. O operador tem a função de parametrizar a linha de acordo com o tamanho da bitola a ser inspecionada. Tal informação é obtida no sistema de automação por meio da ordem de produção a ser processada no momento. Com o auxílio de câmeras de vigilância, cabe ao operador também detectar possíveis misturas. Misturas ocorrem

quando um ou mais tarugo(s) estiver(em) presente(s) em meio a uma ordem de produção que não confere com àquela em que ele foi produzido. Trazer as tarefas de monitoramento e parametrização da linha de inspeção para um nível automático são os desafios presentes no projeto no qual se insere o presente estudo.

3.1 – Conclusões

Neste capítulo é apresentada uma descrição breve da AÇOMINAS e do processo de inspeção de tarugos, salientando as necessidades de um sistema de inspeção automático.

4 – SISTEMA DE VISÃO COMPUTACIONAL

O sistema desenvolvido para tratar do reconhecimento de padrões, que no caso em questão são caracteres numéricos impressos em tarugos de aço, pode ser dividido em duas partes distintas:

1 – hardware – equipamentos para captura de imagem, movimentação da câmera e lentes, iluminação para condicionamento do ambiente e meios de comunicação (cabos coaxiais, etc.);

2 – software – programa desenvolvido em Visual C++ 6.0, utilizando a MFC (*Microsoft Foundation Classes*), tendo como base funções equivalentes às rotinas desenvolvidas anteriormente para identificação de caracteres em ambiente MATLAB [1], além das rotinas e modificações necessárias inseridas a partir do presente trabalho.

4.1 – Descrição dos componentes do *hardware* utilizados

Especificações de equipamentos para compor o sistema foram analisadas e diante disso foram sugeridos os seguintes:

1 – Câmera – um exemplar que fosse robusto o suficiente pra assegurar a devida proteção dos seus componentes internos, visto que o meio industrial normalmente é agressivo (poeira, calor, etc.). No caso do presente trabalho, a digitalização é necessária de qualquer forma, pois a análise será feita pelo computador sobre a representação digital da imagem. Tendo em vista tal fato, a melhor opção é a digitalização mais próxima da fonte, ou seja, a própria câmera deve realizar essa tarefa. Uma digitalização distante da fonte pode acarretar a inserção de ruídos de transmissão no sinal analógico de vídeo. Câmeras digitais podem ser encontradas suportando diferentes resoluções de imagem capturada. Em sendo assim, pode-se deduzir que uma câmera de vídeo digital é a escolha mais coerente;

2 – Lentes – devem ser especificadas de acordo com a distância de trabalho em que se deseja operar;

3 – Iluminação – o problema de condicionamento da iluminação é crucial e deve ser feito de maneira a realçar o que se deseja analisar, além de promover uma repetibilidade no padrão da imagem, independente das condições ao redor (luminosidade proporcional à hora do dia, condições climáticas, etc.);

4 – Placa de captura de vídeo – no caso de ser necessária, é interessante que possua compatibilidade com sistemas operacionais mais recentes;

5 – Meios de comunicação – deve existir um meio de comunicação entre os equipamentos e o local onde será feita a monitoração do sistema. É recomendável que o meio escolhido seja robusto a ruídos e às diferentes agressões causadas pelo ambiente. No caso de uma câmera digital, a comunicação é normalmente feita por par trançado usando *firewire* (Protocolo IEEE-1394) para comunicação. Uma característica desse tipo de comunicação é a necessidade do uso de repetidores (HUB) para o envio de informação por distâncias longas. Para envio de sinal de vídeo analógico, o mais comum é o uso de cabo coaxial.

A câmera utilizada atualmente trata-se de um equipamento comumente escolhido para tarefas de vigilância. A forma de envio do sinal do vídeo é analógica, sendo, então, necessário o uso de uma placa de captura de vídeo. A placa de captura que atualmente encontra-se instalada na estação de trabalho aparentemente não possui *drivers* para computadores cujo sistema operacional instalado seja Windows NT ou superior.

O conjunto de lentes presentes na câmera é motorizado e permite ajuste de diversas funções por meio de um *joystick* de comando localizado remotamente.

A iluminação foi intencionalmente escolhida para realçar os relevos da face do tarugo onde se encontram os números. Entretanto, a rugosidade também é destacada. A direção da iluminação proposta pode ser vista na figura 4.1. Como os números são impressos em baixo relevo, ocorre formação de sombra em seu interior. Esse efeito não garante um sombreamento perfeito na estampa se a impressão estiver desgastada. O efeito da iluminação proposta pode ser verificado na figura 4.2.

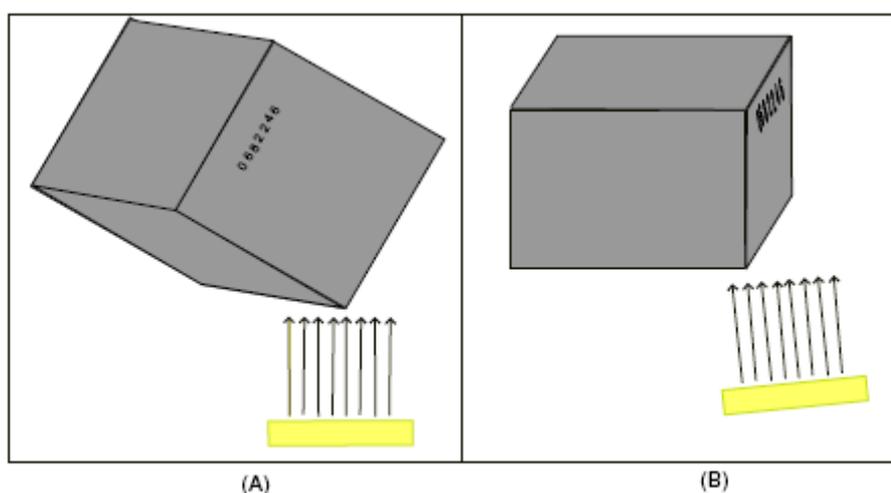


Figura 4.1 – Iluminação para realçar detalhes no plano do tarugo. Em (A): vista da câmera. Em (B): vista lateral do tarugo. FONTE: [1]

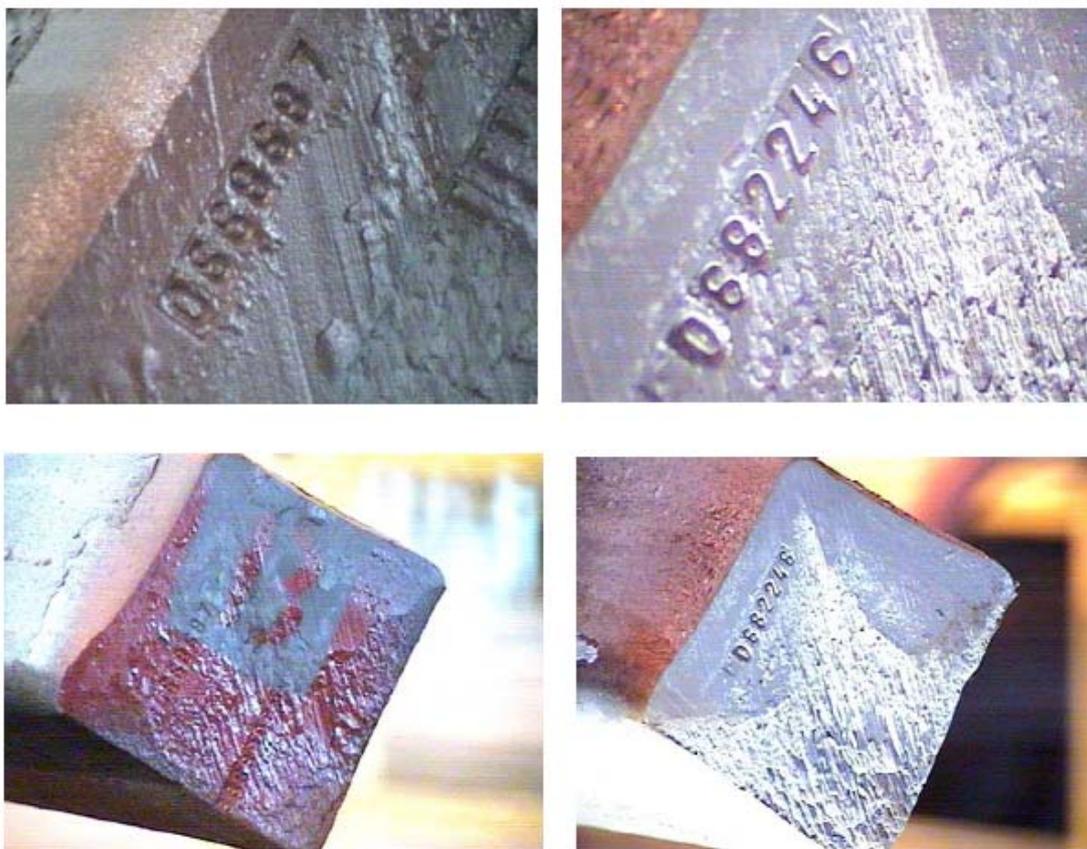


Figura 4.2 – Comparação do efeito da iluminação estruturada nos dois possíveis planos de trabalho. FONTE: [1]

Foi disponibilizada uma placa de captura que transforma o vídeo analógico em imagens de tamanho 640x480 *pixels* a uma taxa de no máximo 30 quadros por segundo. Tal placa *drivers* testados para Windows 98. Os sinais de vídeo captados são do tipo vídeo composto. É fornecido juntamente com a placa um tutorial de desenvolvimento com códigos exemplos e uma biblioteca dinâmica (dll) para facilitar o acesso às funcionalidades da placa [3]. Apesar do manual de desenvolvimento informar que o sistema funciona no Windows NT, foi percebido que, ao utilizar até mesmo um programa exemplo do pacote em alguma plataforma NT, ocorre a realização de uma “operação ilegal” ao tentar capturar uma imagem.

Existem atualmente duas vias de comunicação da câmera com a sala de controle de linha. A primeira delas é referente ao sinal de vídeo analógico, que é transmitido por meio de um cabo coaxial. A segunda conexão tem como objetivo comunicar o sistema de acionamento com os módulos de comando de movimentação da câmera: o cabo utilizado é do tipo 10 Base-T/RJ45 (par trançado). Tal cabo é utilizado normalmente para conectar computadores em rede, mas no presente caso ele é usado pra realizar uma comunicação serial a uma distância razoável.

A Figura 4.3 mostra a disposição dos equipamentos na LIT, pode-se verificar o percurso que os tarugos realizam e em que momento a câmera terá algum deles em seu campo visual. O sensor de presença do tarugo foi colocado com a borda em vermelho na figura indicando que ainda não foi possível sua instalação, mas a sua presença é imprescindível para que o sistema fique operando de maneira automática. O

funcionamento do sensor deve ser de maneira tal que, na presença de um tarugo na mesa de rolos, ocorrerá uma pausa no processo, por alguns segundos, a fim de que o sistema possa realizar a captura e análise da imagem. Quando o tarugo se encontra na mesa de carregamento, não há uma garantia da distância da sua face à câmera. O sensor foi uma alternativa proposta para posicionar a face do tarugo sempre a uma mesma distância da câmera, para tornar as condições das imagens mais similares e minimizar problemas de reconhecimento decorrentes da variação de escala dos números na imagem, já que será utilizada uma estratégia de reconhecimento sensível à escala.

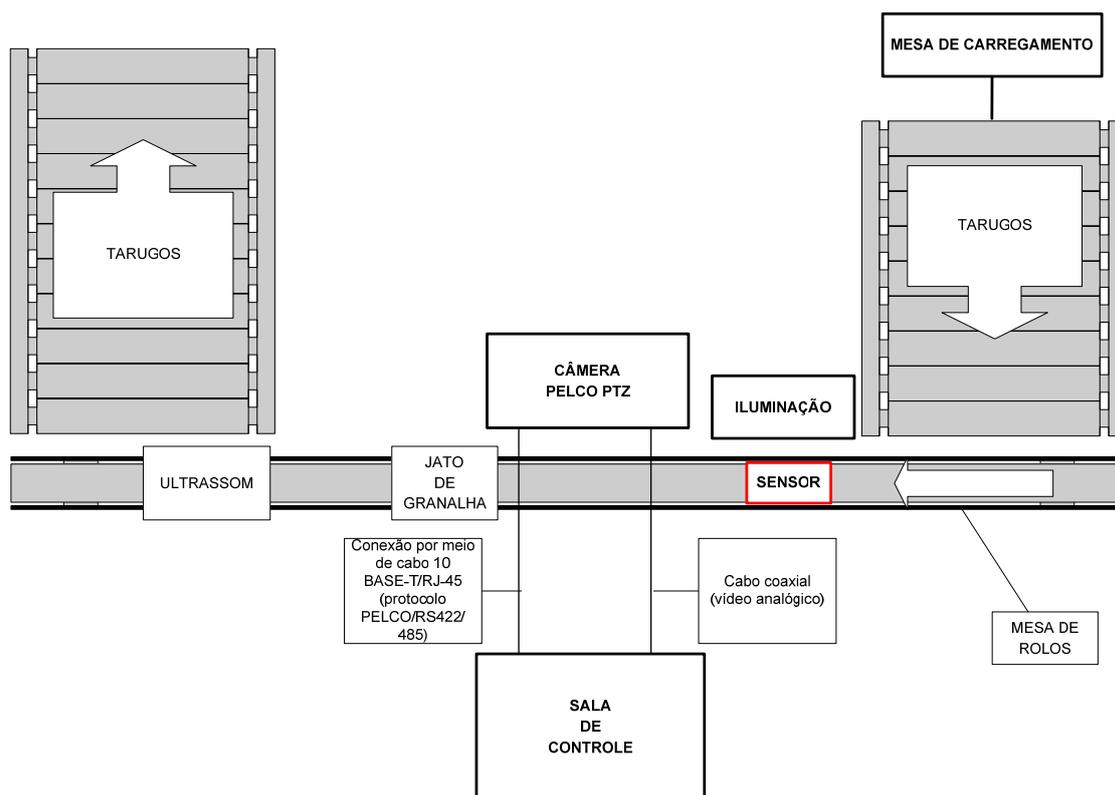


Figura 4.3 – Diagrama das instalações onde será localizado o sistema desenvolvido

A figura 4.4 mostra uma visão lateral da câmera focando seu posicionamento no momento da coleta de imagens na LIT.



Figura 4.4 – Visão do conjunto câmera e iluminação

Por questões operacionais já estava definida a região na LIT onde a câmera de monitoramento está situada; na saída da mesa de carregamento. A abordagem escolhida foi focalizar o primeiro tarugo pronto para entrar na mesa de rolos e ir para o jato de granalha para ser realizada a coleta de imagens (figura 4.5).



Figura 4.5 – Posicionamento da câmera focando a mesa de rolos

A câmera utilizada possui diversos componentes que ditam seu comportamento (figura 4.6). Internamente, existe um sistema de acionamento que comanda os movimentos horizontais (PAN) e verticais (TILT) do corpo da câmera. Existem também

comandos para lentes motorizadas que resultam em modificações no zoom (TELE até WIDE), foco (NEAR até FAR) ou íris (OPEN até CLOSE).

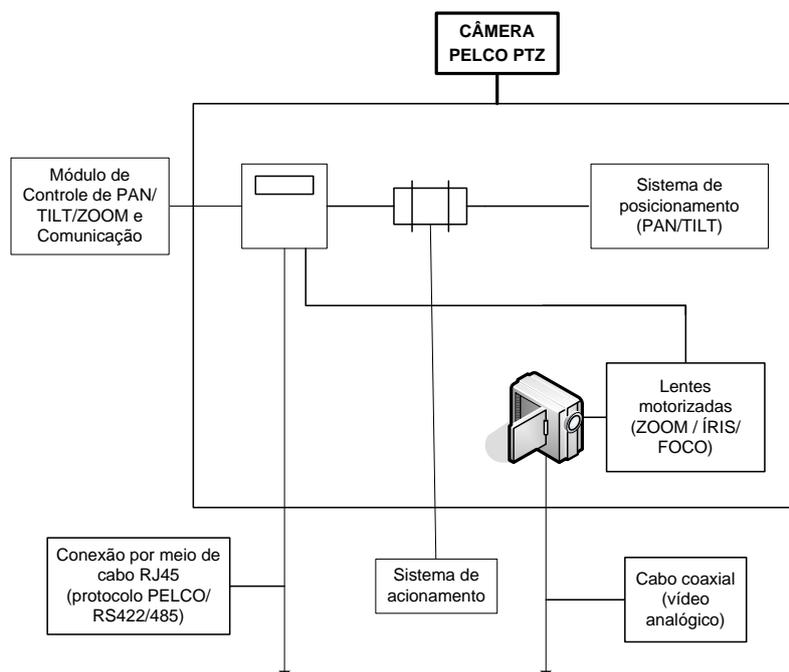


Figura 4.6 – Componentes da câmera PELCO PTZ

O sinal de vídeo proveniente da câmera é enviado tanto aos monitores de vigilância quanto à conexão da placa de captura do computador utilizado para processamento de imagens (figura 4.7). Os diversos comandos fornecidos à câmera são enviados por um *joystick*, que gera um pacote de dados estilo MODBUS, um protocolo serial proprietário da PELCO [4]. Na mensagem de comando está incluída qual câmera se deseja acionar além das informações como intensidade, direção, etc.

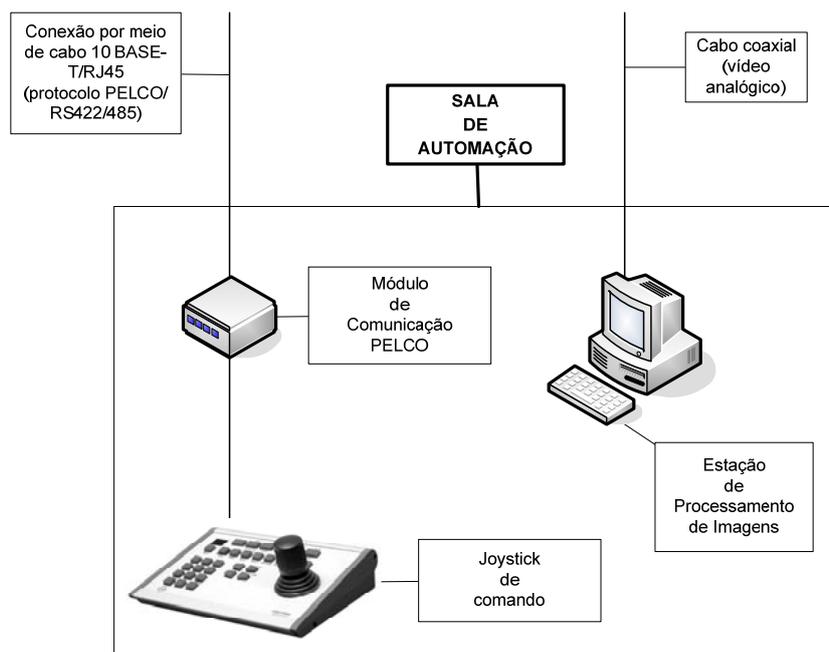


Figura 4.7 – Conexão da sala de automação com a câmera

Para evitar problemas com diferenças entre escalas dos números que aparecem na imagem, foi então escolhida uma posição da câmera que enquadraria a maior bitola de tarugo dentre aqueles tamanhos possíveis de serem inspecionados, que é de 160 mm de aresta. A escolha dos números da estampa é decorrente do tamanho mínimo de bitola, que é 76 mm de aresta. A dimensão da estampagem física possui uma dimensão única e já prevista em função do tamanho da menor bitola de tarugo. Para tentar reproduzir, no ambiente digital, imagens de caracteres que não possuam diferença considerável em sua dimensão foi escolhido manter uma distância fixa do foco da câmera.

4.2 – Descrição do *software* e metodologias propostas

Todas as metodologias propostas serão descritas detalhadamente mais a frente, sendo que algumas delas ainda se encontram codificadas somente para ambiente MATLAB. Pode-se, resumidamente, classificar o processamento ideal sendo feito com as seguintes etapas:

1. Configuração da aplicação;
2. Busca da lista de ordens de produção;
3. Modo de espera não ocupada da aplicação: a saída desse modo ocorre quando há um evento lançado pelo sensor colocado na mesa de rolos;
4. Captura do quadro contido na placa de captura;
5. Rotação da imagem em um ângulo conhecido;
6. Análise, buscando a localização da face do tarugo na cena, ou seja, a segmentação de sua região de interesse na imagem;
7. Análise, buscando descobrir em qual orientação se encontra a região numérica;
8. Segmentação de uma faixa horizontal contendo a região numérica;

9. Comparação dos padrões escolhidos com a região selecionada da imagem;
10. Montagem de uma seqüência numérica reconhecida juntamente com a confiança de cada dígito;
11. Comparação com a lista de ordens de produção programadas;
12. Utilização da informação de tamanho da bitola, contida na lista de ordens de produção, como dado redundante na identificação para diminuir erro no reconhecimento;
13. Configuração dos parâmetros da linha de inspeção para processar uma determinada ordem de produção, ou contar os tarugos de uma ordem já em execução. Ocorrendo o não-reconhecimento ou a identificação de uma ordem de produção diferente da configurada, então o sistema deve informar possível mistura e pedir auxílio ao operador.

4.2.1 – Configuração da aplicação

Entrada de dados por meio da interface dos seguintes parâmetros:

- ***Limiar de reconhecimento*** – valor real de 0 a 1, que estipula o mínimo de confiança para que um caractere seja considerado reconhecido.
- ***Diâmetro mínimo entre picos de reconhecimento*** – quando do reconhecimento de um padrão, a região ao seu redor também possui valores elevados. No entanto, todos esses “máximos” representam o mesmo padrão. Tal parâmetro deve ser configurado para que cada máximo ou região de máximos seja reconhecida unicamente. Mais detalhes são fornecidos na seção 4.2.9;
- ***Relação pixel por milímetro*** – parâmetro necessário para o cálculo da dimensão da bitola (dependente da distância da face do tarugo e do posicionamento do foco da câmera);

4.2.2 – Armazenamento da lista contendo as ordens de produção (OP)

A lista de OP's do dia contém informações relevantes, tais como o número das ordens programadas para o dia, e também a dimensão da seção quadrada de cada uma delas. Essas informações serão importantes para inserir dados redundantes na identificação e melhorar o desempenho global.

O ideal é possuir a requisição dessas informações do próprio sistema de automação. Entretanto, isso ainda não foi possível. Atualmente, a lista de ordens de produção deve ser inserida manualmente em um arquivo de configuração lido na inicialização do programa.

4.2.3 – Modo de espera por eventos

A aplicação, após ser configurada e carregar todos os parâmetros necessários para sua execução, entra em modo de espera por eventos. Após o acontecimento do evento esperado, a aplicação captura a imagem, faz as análises devidas, gera resultados, e, no final, retorna ao modo de espera por eventos.

No planejamento do sistema, foi prevista a necessidade da colocação de um sensor de presença para identificar tarugos sendo inseridos na mesa de rolos. No

momento em que o tarugo fosse posicionado devidamente, o sensor é acionado, interrompendo o fluxo momentaneamente. Então é gerada uma informação para o sistema de automação e, em seguida, a aplicação será acionada.

Atualmente, devido ao caráter de protótipo em que o sistema se encontra, o sensor ainda não foi instalado, e não há previsões para tal, até que a parte de identificação esteja suficientemente confiável. A comunicação com o nível de automação ainda não foi executada por diversos motivos: um deles é que a parte de identificação merece mais atenção, e esta é uma das últimas etapas a ser implantadas. Uma questão interessante de ser levantada é como fazer para que o sistema de automação envie uma informação assíncrona de acionamento do sensor de presença para a aplicação, gerando o evento de processamento de imagem. Esse tipo de problema merece uma atenção considerável, pois soluções como, por exemplo, fazer a aplicação checar de forma cíclica o estado do sensor podem congestionar a rede e causar problemas na linha.

4.2.4 – Captura da imagem

A captura da imagem para análise pode ser feita de 2 maneiras atualmente:

- Arquivo – necessário para *debug* da aplicação: a imagem é carregada de um arquivo bitmap;
- Microsoft DirectX – captura um quadro em um dispositivo compatível com essa API (Application Programming Interfaces). Em sistemas Windows NT ou superiores essa é uma maneira de acessar o dispositivo de captura de vídeo.

Após a captura da imagem, ocorre o envio do quadro para um objeto que irá manipulá-la. Dentre as funcionalidades desse objeto, podem-se destacar, por exemplo, realizar redimensionamento e/ou rotação com interpolações diversas, possibilidade de aplicação de diversos filtros clássicos na imagem, etc. Tais funcionalidades são muito úteis para facilitar as demais etapas.

4.2.5 – Correção da rotação conhecida

Na posição onde foi encontrada a melhor qualidade de imagem possível, tem-se uma rotação física conhecida de aproximadamente 45° do tarugo. Porém, pode ser demonstrado que, devido à visão em perspectiva, a rotação a ser realizada na imagem digital para compensar tal efeito a fim de tornar as laterais do tarugo paralelas aos limites da imagem corresponde a aproximadamente 35° no sentido anti-horário (figura 4.8) [1]. O sistema executa essa tarefa como primeira etapa de processamento e realiza a rotação utilizando interpolação bilinear. Se em algum momento ocorrer estampagens numéricas não paralelas às arestas do tarugo, a identificação posterior pode ser comprometida.

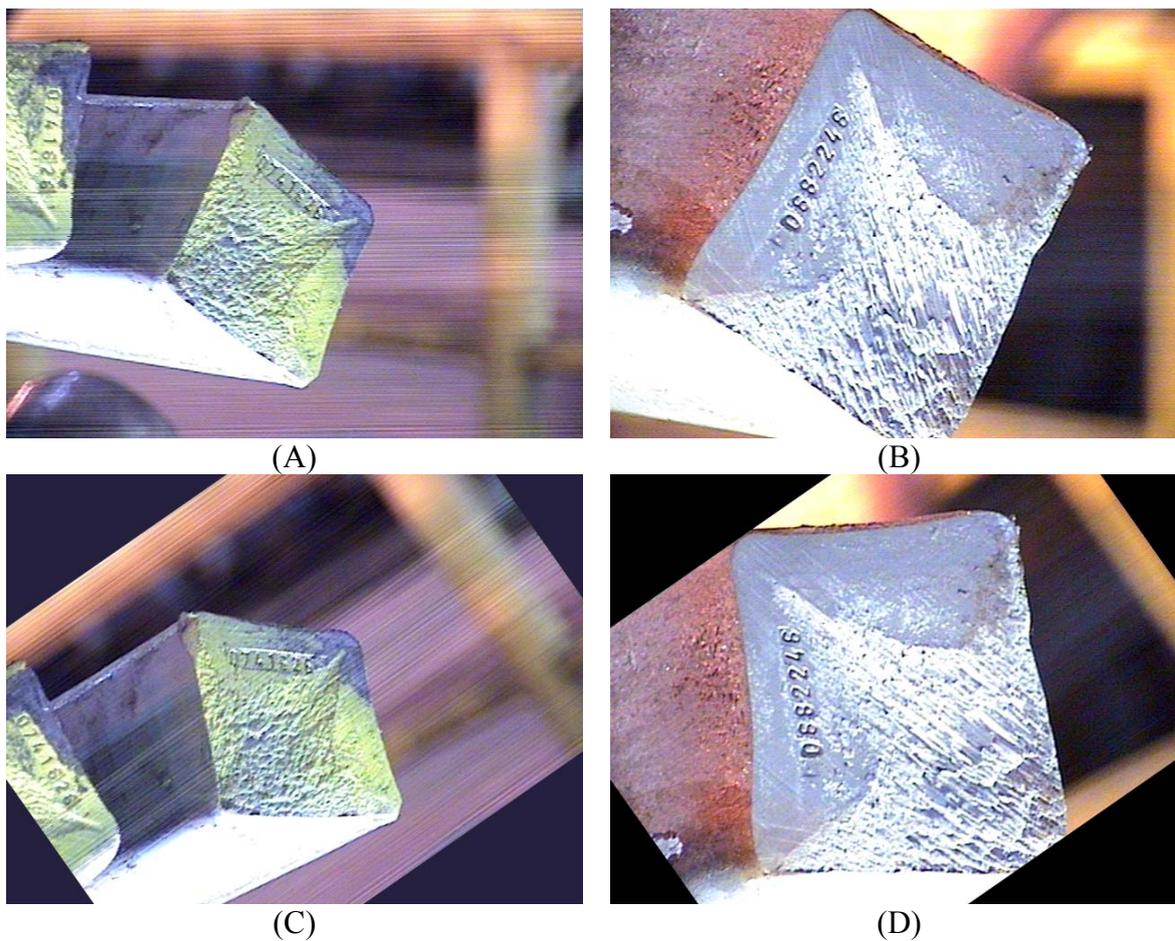


Figura 4.8 – Em (C) a imagem (A) após rotação estimada. Em (D) a imagem (B) após rotação estimada

4.2.6 – Segmentação do tarugo ou localização do tarugo na cena

O passo seguinte após a correção da rotação é encontrar o retângulo que delimita a região da imagem onde se encontra a face do tarugo. É fato que, conforme especificações, a face do tarugo deve possuir seção aproximadamente reta. Encontrar a região que melhor define a face do tarugo é de extrema importância pelo fato de que será utilizada como dado redundante na identificação. Com a calibração da área da imagem em função da área real, pode-se estimar o tamanho da bitola e compará-la com a que deve estar realmente sendo inspecionada no instante.

A abordagem inicial para isolamento do tarugo foi proposta tendo como base uma segmentação utilizando o sistema de cores RGB [1]. A idéia é utilizar-se do efeito proporcionado pela iluminação, característica essa que demonstra uma forte tendência na imagem de enfatizar tons azulados na face e tons avermelhados ao redor. Baseado nisso, foi sugerida uma operação de subtração da componente vermelha na componente azul (B-R) [1] e uma posterior limiarização automática global [10]. O resultado da metodologia pode ser visto na figura 4.9.

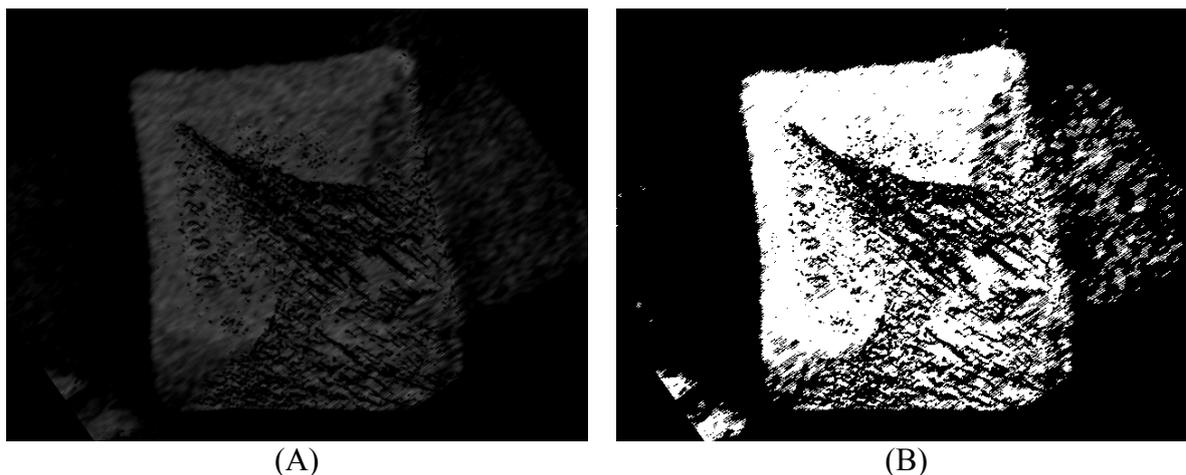


Figura 4.9 – Em (A): subtração da componente vermelha na azul. Em (B): resultado da limiarização global

A metodologia mostrou-se bastante eficiente ao promover um bom destaque da face do tarugo. Sendo assim, foi posteriormente implantada como método de binarização da imagem a fim de possibilitar posteriormente a localização do retângulo que melhor delimita a face do tarugo.

4.2.6.1 – Estratégia inicial implantada

Mesmo em posse de uma estratégia para destacar a face do tarugo na cena, um problema ainda fica em aberto, que é um método para encontrar um retângulo que melhor represente a região demarcada pelo tarugo.

Em alguns tipos de conversores A/D (analogicos→digitais), para se encontrar qual a melhor seqüência de bits que represente um valor de tensão na entrada do conversor, é utilizado um processo de pesquisa binária. Nesse método, é feita uma pesquisa binária da combinação de N bits do conversor que mais se aproxima da tensão de entrada V_{IN} (tensão de entrada). Inicialmente, o registro onde será armazenada a representação em N bits da tensão V_{IN} é deixado zerado e, em seguida, acontece o ensaio do bit mais significativo (MSB). Se a tensão de saída do conversor exceder V_{IN} , então tal bit é deixado em '0'; caso contrário, é atribuído '1'. O bit seguinte é ensaiado e o processo é repetido ao ritmo do sinal de relógio (clock), até que o bit menos significativo seja testado (LSB). Independente de qual tensão se queira representar, são necessários N passos para que ela seja representada pelo conversor.

Um exemplo para um conversor de 4 bits pode ser visto na figura 4.10. O eixo y representa a tensão de saída do conversor D/A (DAC). No exemplo, a primeira comparação mostra que $V_{IN} < V_{DAC}$. Logo, ao bit 3 é atribuído o valor '0'. No conversor D/A é então atribuído o valor 0100_2 e a segunda comparação é realizada. Como $V_{IN} > V_{DAC}$, o bit 2 permanece em '1'. O conversor D/A recebe o valor 0110_2 , então a terceira comparação é realizada. Ao bit 1 é atribuído o valor '0', e o conversor D/A é então modificado para armazenar 0101_2 para sua última comparação. Finalmente, o bit 0 permanece em '1' porque $V_{IN} > V_{DAC}$.

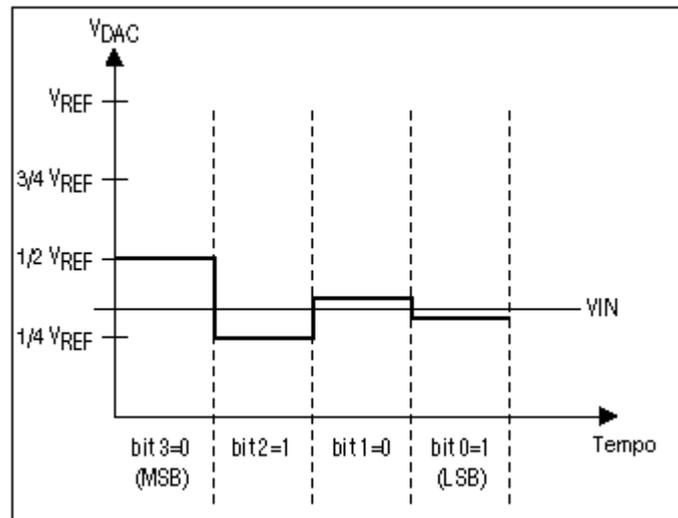


Figura 4.10 – Operação no registro de aproximações sucessivas (SAR) (exemplo para um conversor de 4 bits)

Analogamente ao que é feito no conversor A/D exemplificado acima, será proposta a seguir uma forma de se encontrar o melhor retângulo na imagem. É necessário que seja visível na imagem algum retângulo que possa definir onde se encontra a face do tarugo, como pode ser percebido na figura 4.9-B. O retângulo deve possuir seu perímetro imerso, em sua maior parte, na região definida em cor branca. Além disso, o retângulo deve também estar na região limite, tal que a próxima expansão em sua dimensão resulte em um retângulo com perímetro imerso em uma região predominante de cor preta. Um retângulo é inserido inicialmente de maneira a ter certeza que ele estará totalmente imerso na região em branco, predominantemente. Em seguida, cada lado do retângulo começa independentemente a aumentar ou diminuir suas dimensões proporcionalmente a um fator de escala configurável (no caso da pesquisa binária tal fator é sempre uma potência de 2). A figura 4.11 contém um exemplo de todos os passos feitos pelo algoritmo proposto até chegar na solução final. Cada lado desloca-se no sentido de aumentar o perímetro, quando seu passo anterior ainda continua imerso em região branca. Se o lado cair em uma região predominante preta, então o lado retrocede metade da distância entre o passo atual e o último passo, que ainda se encontrava em região branca.

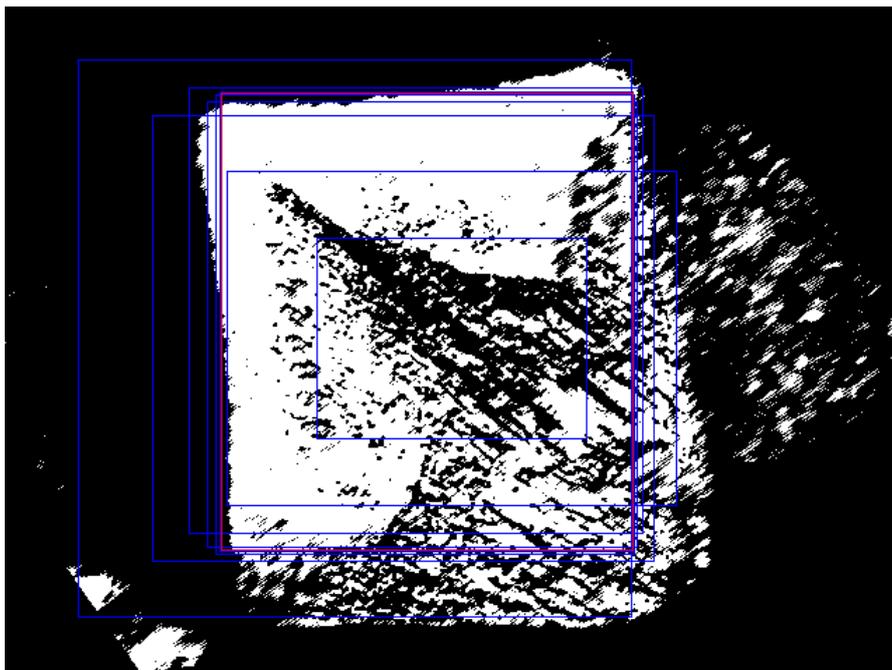


Figura 4.11 – Iterações do algoritmo de segmentação do tarugo, resultado final em vermelho

Em seguida, o retângulo encontrado é recortado da imagem original já rotacionada. Os processamentos seguintes serão feitos no resultado dessa etapa. Em função da calibração pixel por milímetro feita anteriormente, é possível determinar ao final dessa etapa um valor estimado para a bitola do tarugo. A complexidade do método é semelhante ao do método da pesquisa binária ($O(4 \cdot \log_k(n))$), sendo que o algoritmo é executado nos n pontos 1 vez para cada aresta, e a variável k define o passo (no caso da pesquisa binária o passo é igual a 2).

4.2.6.2 – Alternativa utilizando perfil de variância

A alternativa demonstrada no item 4.2.6.1 mostrou um desempenho determinístico e ótimos resultados, os quais serão quantificados mais à frente. Porém, uma característica do processo a tornou inviável. Devido às condições de iluminação fluorescente, qualquer tarugo pode ficar com tonalidade azulada se a sua face mantivesse suas características originais, ou seja, permanecesse acinzentado sem adição de nenhuma pintura. No processo de inspeção, foi identificado que em alguns instantes ocorre a pintura das faces por exigência do cliente ou para evitar misturas, principalmente em se tratando de aços destinados a forjaria. As pinturas utilizadas podem ser de várias tonalidades; dentre as que foram identificadas, podem-se citar o amarelo e vermelho. A figura 4.12 é um exemplo de resultado da metodologia de binarização em casos onde ocorre a pintura da face do tarugo.



Figura 4.12 – Binarização utilizada no item anterior

Como era de se esperar, uma metodologia baseada em binarização utilizando os planos de cores e esperando uma predominância de tons azulados não tem como destacar a face de um tarugo pintado de amarelo. Para contornar o problema, outra estratégia teve que ser adotada.

Como pode ser percebido na figura 4.13, a detecção de bordas pode fornecer resultados promissores [1]. Entretanto, a imagem apresenta uma quantidade considerável de ruído. Para um ser humano, a tarefa subjetiva de encontrar uma face aproximadamente quadrada na figura 4.13 é trivial. Entretanto, transformar isso em algum algoritmo computacional pode ser complexo. Apenas observando os limites da face, pode-se inferir que o tarugo se encontra entre os pontos 150 a 370 no eixo das ordenadas e entre 220 a 420 no eixo das abscissas. Semelhante à análise feita com etiquetas, descrita no item 2.4, pode-se tentar analisar o perfil de variância do resultado da detecção de bordas e verificar se existe algum destaque da região onde se encontra o objeto de interesse. Um fator que influencia negativamente no resultado da próxima análise ocorre quando a rotação não proporciona uma figura com arestas aproximadamente paralelas às laterais da imagem. A consequência disso é um perfil de variância com crescimento suave e não abrupto. Tal fato leva a uma incerteza muito grande da região do tarugo e inviabiliza as próximas etapas.

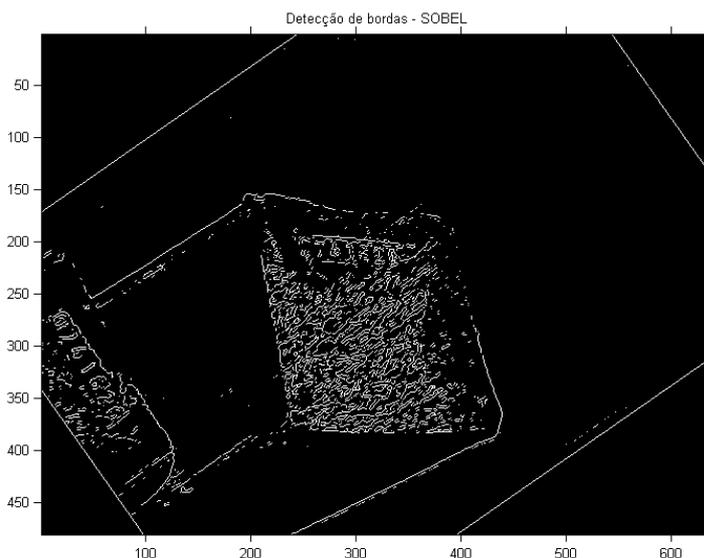


Figura 4.13 – Detecção de bordas na FIG.9-C usando operador de Sobel [12] e limiar automático (eixos coordenados conforme convenção ditada pela varredura de vídeo)

Esperava-se que por meio da análise de perfil da variância fosse possível destacar a região da face do tarugo, e foi o que ocorreu nesse caso. Analisando a figura 4.14, subjetivamente, é possível se estimar os limites da face do tarugo, que é aproximadamente entre os pontos 200 a 380 referentes ao eixo das ordenadas da figura 4.13 e entre 210 a 400 referentes ao eixo das abscissas da figura 4.13. O ruído visto na imagem da figura 4.13 representa outro tarugo. Se a imagem for coletada na mesa de rolos, onde o tarugo é isolado, não ocorre tal tipo de ruído.

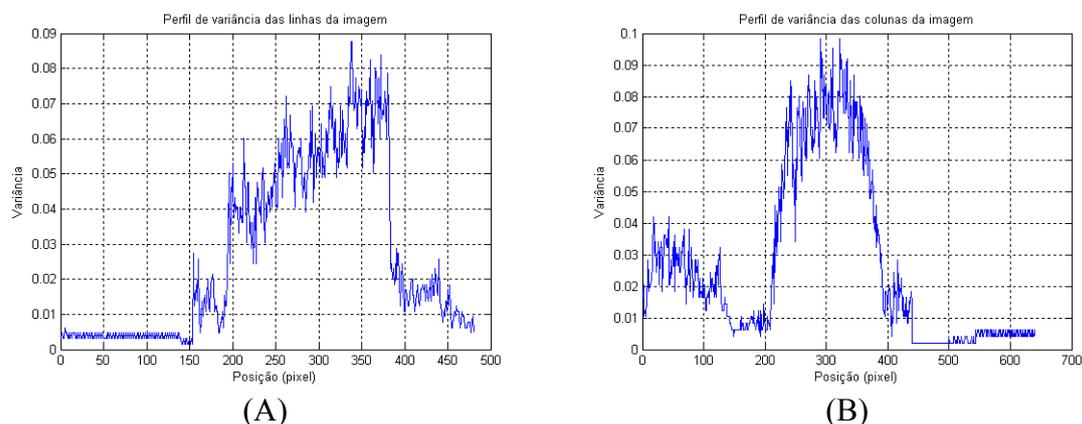


Figura 4.14 – Em (A): perfil de variância das linhas da imagem (sem normalização). Em (B): perfil de variância das colunas da imagem (sem normalização)

Para utilizar adequadamente tal estratégia, ainda restam duas questões:

- Os valores de mínimos e máximos da variância podem diferir muito de imagem para imagem;
- Como gerar um algoritmo a fim de encontrar os limites dos platôs nos vetores de variância;

Para o problema de valores distintos de máximos e mínimos na variância em cada imagem analisada, decidiu-se normalizar os dados antes da análise e suavizar o sinal utilizando um filtro não-causal que não aplica defasagem no sinal. A normalização é feita utilizando os máximos e mínimos do próprio vetor. A normalização baseada somente no vetor que se deseja analisar pode não ser tão confiável, porque uma situação em que não ocorra destaque gráfico da face do tarugo pode ser interpretada incorretamente se a normalização for feita baseada em dados do próprio vetor.

Gerar um algoritmo para encontrar os limites pode não ser uma tarefa trivial porque, primeiramente, o sinal tem ruído que pode interferir nos resultados. Uma filtragem prévia do sinal que não aplique defasagem em baixas frequências, que possua ganho unitário, e que remova as altas frequências facilita bastante a análise. Levando-se em conta que os limites do patamar gerado pela análise têm sua posição e tamanho variáveis, isso torna a análise bem subjetiva. Uma solução para transformar tal subjetividade em algo palpável é o uso de lógica nebulosa.

A partir de simples observação gráfica, pode-se deduzir que os pontos que determinam os limites de interesse estão contidos no conjunto de mínimos locais do gráfico. Também é característica dos limites o fato de estarem bem próximos de regiões com alta variação de valor, ou seja, o início ou o fim de um patamar. O início da fronteira é determinado por um mínimo local que possui alguns locais mínimos posteriores a valores “grandes” e os anteriores a valores “pequenos”. O fim da fronteira é determinado por um mínimo local que possui alguns locais mínimos posteriores a valores “pequenos” e os anteriores a valores “grandes”. Tendo em posse tal conjunto de regras subjetivas, resta então a definição das regras equivalentes em um modelo nebuloso [20].

A figura 4.15 mostra os perfis de variância nas direções horizontal e vertical e o diagnóstico nebuloso correspondente a cada uma delas.

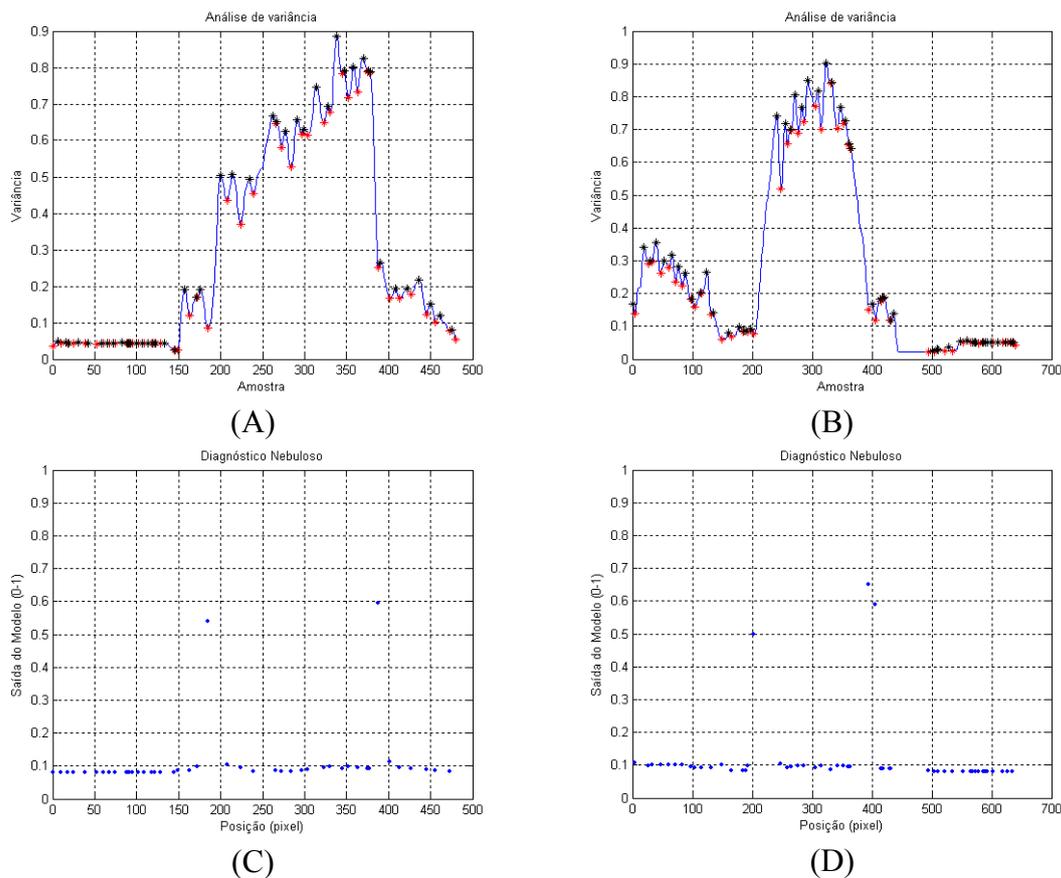


Figura 4.15 – Em (C) o diagnóstico nebuloso para os pontos normalizados selecionados em (A). Em (D) o diagnóstico nebuloso para os pontos normalizados selecionados em (B).

Com base no diagnóstico nebuloso, pode ser tomada a decisão de delimitar a região da face do tarugo, escolhendo dentre os candidatos resultantes os que melhor representam as fronteiras tanto verticais como horizontais. O resultado obtido, que pode ser visualizado na figura 4.16, indica resultados promissores que ainda podem ser melhorados apenas ajustando as regras do modelo Mamdani [20] utilizado.

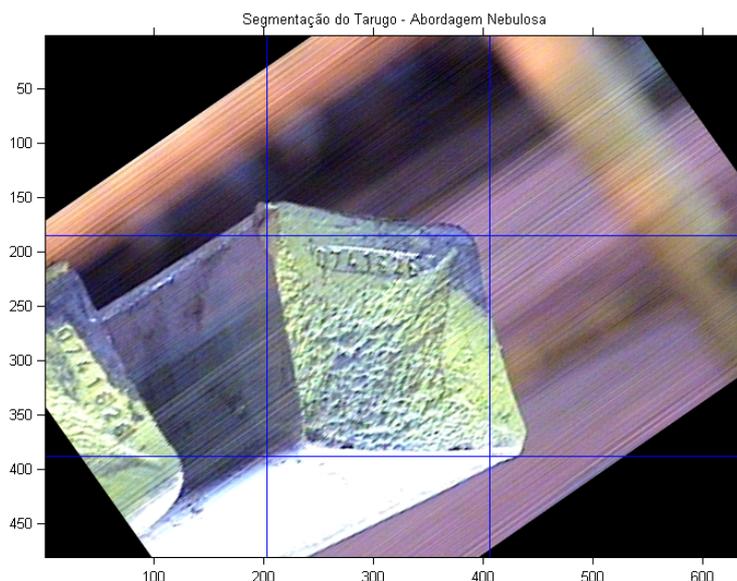


Figura 4.16 – Resultado da segmentação baseada em regras nebulosas

4.2.7 – Rotação

Devido a uma característica do processo, todos os tarugos, sem exceção, possuem uma parte lisa e uma parte rugosa. Tal característica é determinada quando ocorre o corte pela tesoura rotatória. Durante o processo, parte do tarugo é cortada efetivamente e o restante é rasgado da peça original. Isso faz com que todo tarugo possua uma divisão em uma de suas diagonais, definindo duas áreas que são a lisa e a rugosa. A estampa numérica **sempre** ocorre na parte lisa como visualizado na figura 4.17.

Não existe garantia da orientação da estampa numérica na linha de inspeção de tarugos (figura 4.18), fazendo-se necessária a sua prévia identificação para que as próximas etapas sejam factíveis.

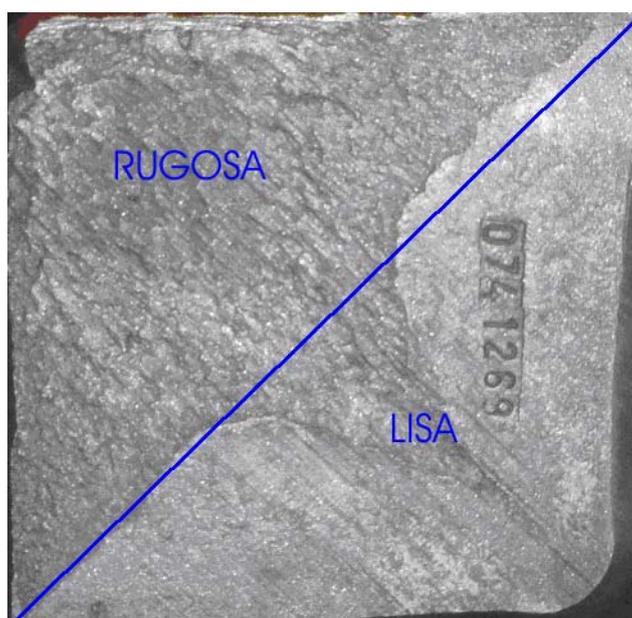


Figura 4.17 – Destaque da parte lisa, com a estampa numérica, e rugosa do tarugo.

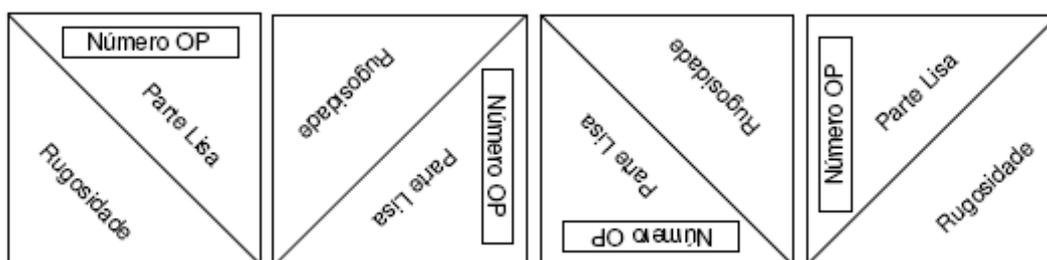


Figura 4.18 – Possíveis orientações do tarugo

Várias metodologias foram testadas e desenvolvidas para a identificação e conseqüente correção da orientação [1]. Essa etapa, apesar de não ter sido desenvolvida no presente trabalho, é de vital importância para o funcionamento normal das seguintes fases de processamento.

O método utilizado e implementado pode ser denominado como decisor baseado em blocos, onde blocos são regiões retangulares na imagem. O método consiste em se confrontar a variância de blocos opostos a fim de determinar qual região é lisa e qual é rugosa. Outro fator é adicionado à análise, que é o fato de que linhas opostas não podem possuir a mesma textura. Os parâmetros do decisor podem ser visualizados na figura 4.19.

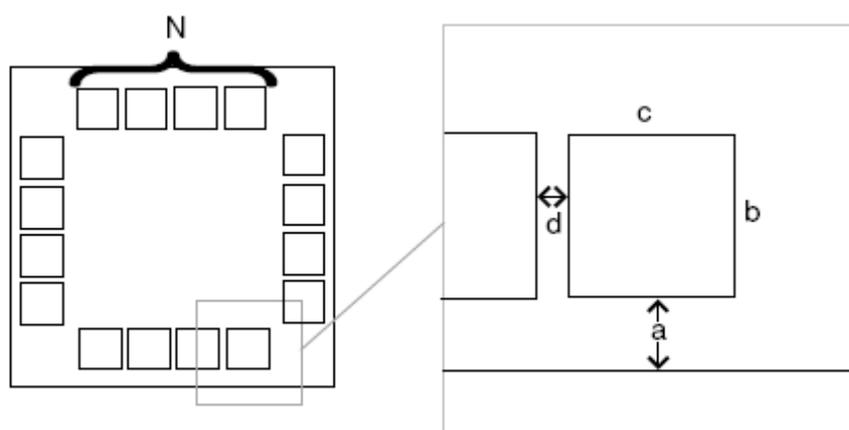


Figura 4.19 – Parâmetros do decisor baseado em blocos

4.2.8 – Redução do espaço de busca

Após a fase de correção de orientação, tem-se que o tarugo está com a estampa numérica voltada para cima. Critérios do processo definem que a estampa numérica obrigatoriamente deve ficar na metade superior do tarugo, quando esse se encontra com orientação correta. Baseando-se nesse fato, pode-se desprezar a metade inferior da face contendo somente região rugosa sem informação útil (figura 4.20). Reduções no espaço de busca são extremamente necessárias visto que os processamentos subseqüentes,

baseados em correlação, têm custo computacional alto e proporcional ao quadrado do tamanho da imagem analisada.



Figura 4.20 – Primeira redução do espaço de busca

Inicialmente, a imagem obtida pela figura 4.20 é utilizada e como variável de entrada para as próximas etapas. Mas, devido ao custo computacional alto percebido, ainda pode ser sugerida mais uma etapa para redução de espaço de busca. A metodologia é praticamente idêntica à utilizada na segmentação da face do tarugo. Vale ressaltar que o parâmetro de entrada da atual etapa é a parte da imagem que é predominantemente lisa e, além disso, contém a estampa numérica. Uma detecção de bordas é realizada, como demonstrado na figura 4.21, e o esperado é que o destaque ocorra na região numérica e em algumas partes rugosas aleatoriamente.

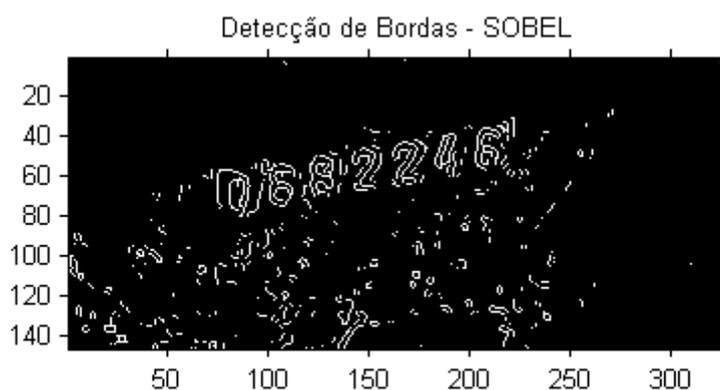


Figura 4.21 – Detecção de bordas na figura 4.20 utilizando o operador de Sobel e limiar automático

A análise de perfil de variâncias das linhas horizontais forneceu resultados promissores, sendo que o fato do destaque da região numérica auxiliou bastante a identificação (figura 4.22-C). Entretanto, a análise de linhas verticais não foi tão bem sucedida (figura 4.22-D). Dentre as diversas oscilações bruscas encontradas no gráfico da figura 4.22-B, muitas são indicativas de fronteiras entre os números. A informação de fronteira entre os números, sendo mais visível e menos sujeita a erros, tornará os processamentos seguintes muito mais eficientes. Porém, trata-se de uma tarefa bastante arriscada para o presente caso, pois uma segmentação de cada dígito somente é recomendável se existir uma maneira de evidenciar as fronteiras entre eles.

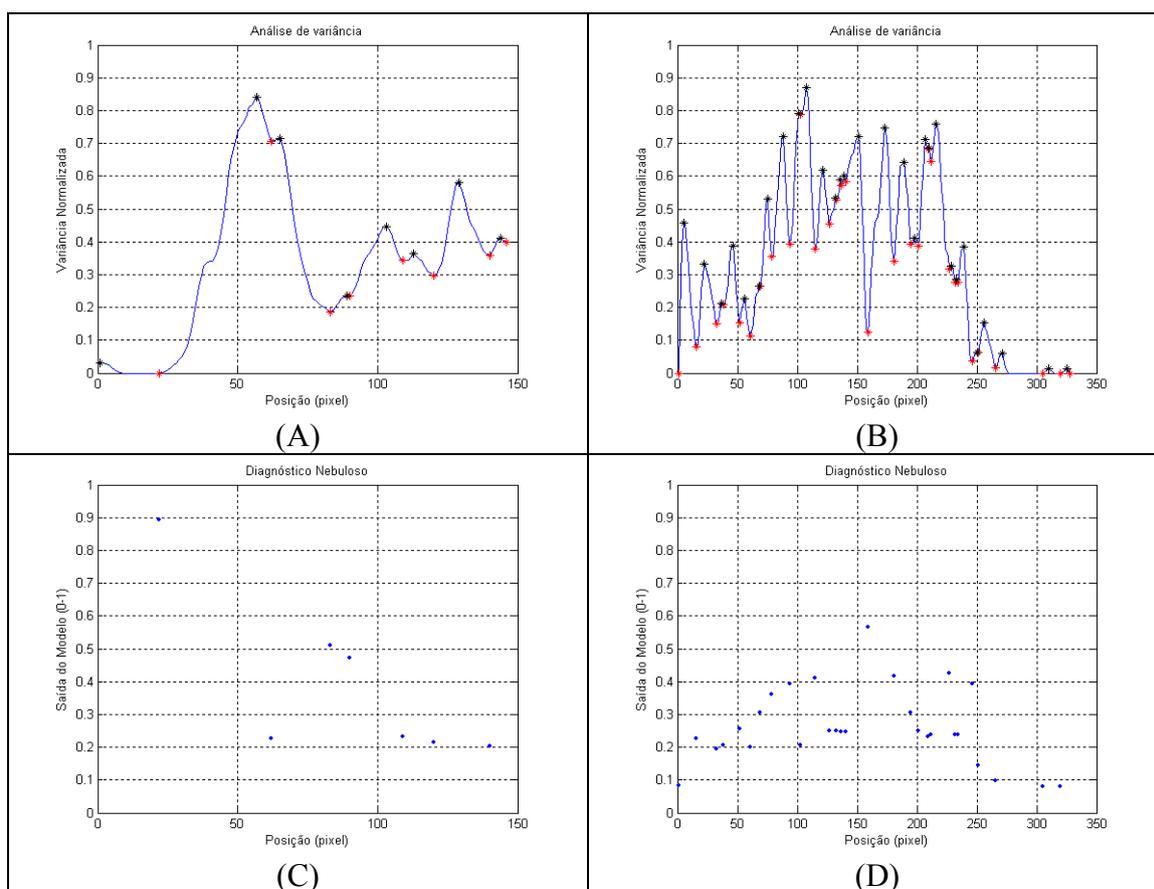


Figura 4.22 – Em (C) o diagnóstico nebuloso para os pontos normalizados selecionados em (A). Em (D) o diagnóstico nebuloso para os pontos normalizados selecionados em (B).

Com a determinação das fronteiras superior e inferior do número, os processamentos seguintes ficam mais eficientes e confiáveis. Na análise atual, consegue-se perceber pela figura 4.23 que ocorreu uma redução do espaço de busca na imagem, e também pode ser percebido o erro decorrente da rotação dita “conhecida” realizada na primeira etapa de processamento. Se o erro na rotação for muito acentuado, como nesse caso, pode haver interferência tanto na análise do perfil de variância, como nas análises seguintes, que não esperam por dígitos inclinados.

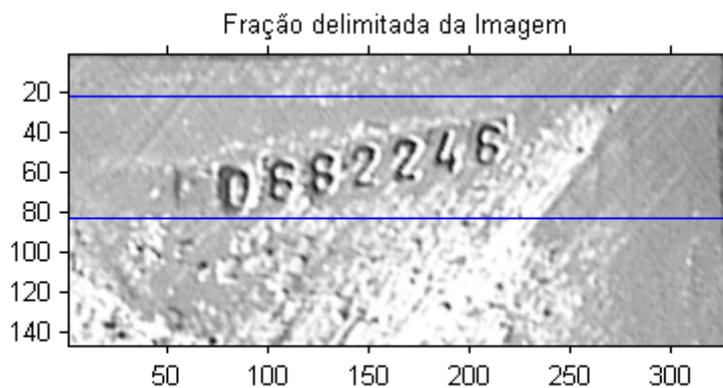


Figura 4.23 – Redução do espaço de busca utilizando análise de perfil de variâncias e lógica nebulosa

4.2.9 – Correlação cruzada normalizada feita com padrões desejados

A partir da imagem obtida anteriormente (figura 4.20), uma análise comparativa de padrões já pode ser feita, objetivando identificar quais padrões aparecem com maiores índices de confiança na imagem. Contudo que a etapa demonstrada na figura 4.23 reduza o espaço de busca sem danificar a região numérica, então o resultado é o mesmo do alcançado com a figura 4.20, sendo que a única alteração é referente ao tempo de processamento. A análise feita a seguir considera que existe um banco de padrões previamente escolhido, e é importante para a robustez do sistema que cada dígito seja representado por vários padrões em escalas diferentes para evitar reconhecimentos equivocados. Entretanto, tal fato leva o sistema a possuir um maior tempo de processamento, sendo que a presente etapa é a mais lenta de todo o processo.

Como dito anteriormente, a correlação cruzada normalizada gera valores no intervalo $[-1 \ 1]$. Para a análise, apenas o valor absoluto do resultado da correlação cruzada é interessante, e, para facilitar a visualização do comportamento do sistema, foi inserido um mapeamento dos valores obtidos na matriz em valores de cores no sistema RGB. O mapa de cores utilizado possui a denominação de “jet” no MATLAB, e é semelhante ao utilizado em sistemas de termografia, para mapear valores de temperatura (figura 4.24).

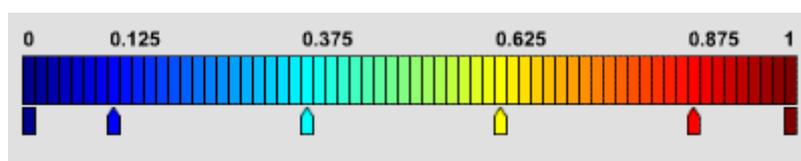


Figura 4.24 – Mapa de cores JET utilizado

Para exemplificar o resultado do processamento do mapeamento sugerido foi coletado na imagem contida na figura 4.20 o padrão representando o número “2”, correspondente ao quinto dígito da seqüência numérica “0682246” contida na estampagem da ordem de produção analisada. Como o padrão está contido na imagem,

então era de se esperar que, após o processamento da correlação cruzada normalizada, um pico aparecesse na região onde o padrão se sobrepõe à imagem original. Tal pico indica o centro de uma região na imagem que mais se aproxima do padrão procurado. A figura 4.25 mostra um padrão para o dígito “2”, de tal maneira que foi escolhido a partir das imagens coletadas para teste.

2

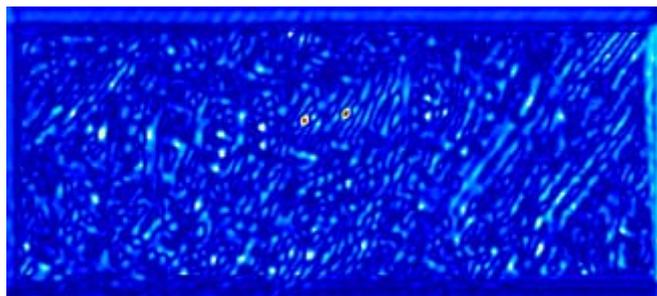
Figura 4.25 – Padrão coletado da figura 4.21, escolhido para exemplificar o mapeamento dos valores de correlação em tons de cores.

Pode-se verificar na figura 4.26-B que o resultado foi uma matriz indicando, em sua maior parte, tons de cores azulados. Conforme mapeamento determinado na figura 4.24, isso indica valores de correlação em torno de zero ou desprezíveis. Regiões amareladas ou com tonalidades que tendem ao verde indicam correlação em torno de 0,5 a 0,6, aproximadamente. Ocorreu o aparecimento de dois picos indicando, primeiramente, que foram encontradas as coordenadas do centro do padrão coletado e que o valor de correlação é igual a 1. Além disso, foi encontrado outro possível candidato na sua vizinhança, que é exatamente o quarto dígito da seqüência numérica “0682246”, que por acaso também é um número 2. É necessário lembrar que valores negativos da correlação normalizada indicam semelhança com o complemento do padrão procurado. Entretanto, quando é feito o mapeamento, toma-se o absoluto dos valores de correlação, significando que os picos encontrados podem tanto representar semelhança direta como semelhança com o complemento do padrão.

Ainda a respeito da figura 4.26-B, pode ser verificado o aparecimento de regiões de alta correlação em torno dos picos de maior significância. Tal fato pode levar a interpretações incorretas de que existem vários padrões em uma única região de pico, quando de fato só existe um único de interesse. Percebendo que esse tipo de situação é freqüente, ocorreu a necessidade de se desprezar valores de correlação muito altos em torno de um mesmo pico. Essa seleção de picos na imagem ocorre primeiramente buscando valores de correlação na matriz de correlação cruzada normalizada acima de um limiar arbitrado (normalmente $\geq 0,6$). Tal limiar define qual valor é razoável para indicar um candidato a padrão reconhecido na imagem. Após a escolha de candidatos, ocorre a separação dos valores de pico e seus vizinhos. Em seguida, são removidos os candidatos que estão dentro de um mesmo raio, deixando somente aquele com maior valor de correlação para aquela região. O raio que determina a região onde um único pico será considerado deve ser previamente configurado pelo usuário, e sua escolha depende de verificações experimentais, visto que ele não pode ser grande a ponto de encobrir os picos apresentados por outros dígitos que se queira reconhecer (figura 4.27).



(A)



(B)

Figura 4.26 – Em (A): imagem original onde se deseja procurar algum padrão de interesse. Em (B): Matriz de correlação cruzada resultante do processamento com o padrão da figura 4.25 e em seguida mapeada em cores.

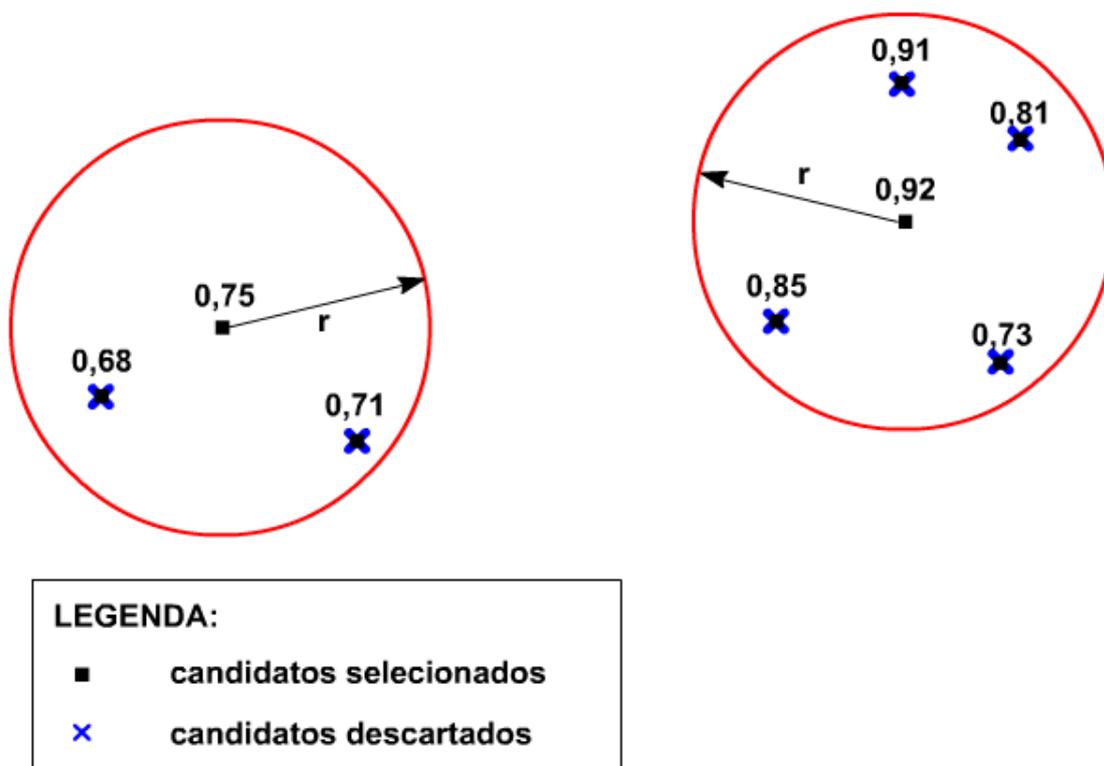


Figura 4.27 – Estratégia de seleção de picos de correlação superiores a um limiar previamente configurado e que possuam uma distância $2r$ mínima entre eles.

4.2.10 – Binarização

A simples correlação em tons de cinza pode tornar o sistema muito sensível a variações ambientais. Então, foi escolhido um pré-processamento muito utilizado na literatura antes da etapa de comparação entre padrões, que é a binarização da imagem [1]. Entretanto, no presente caso, uma binarização em uma imagem que possua um contraste insuficiente pode ocultar a região de interesse. Métodos de binarização

baseados em determinação de limiares globais tendem a ocultar detalhes que em certas situações podem ser de interesse.

Na seqüência, serão apresentados como exemplo dois resultados distintos do mesmo procedimento aplicado na figura 4.26-B, sendo que o primeiro sofre um pré-processamento de binarização global sugerido por Otsu [10], e o segundo sofre um pré-processamento de binarização local proposto por Niblack [11] e sugerido por Ramalho [1] para o presente caso.

A figura 4.28 mostra o padrão coletado para o dígito “2” e exemplificado na figura 4.25 após um processo de binarização proposto por OTSU. As figuras 4.28-A e 4.28-B mostram resultados de binarização utilizando o método de Otsu e Niblack, respectivamente, ambos resultados são referentes à imagem da figura 4.26-A.

2

Figura 4.28 – Padrão da figura 4.25 binarizado com o método de OTSU.



(A)



(B)

Figura 4.29 – Em (A): imagem resultante da binarização global utilizando o método proposto por Otsu [10]. Em (B): imagem resultante da binarização local proposta por Niblack [11] com parâmetros ($N = 31$ e $\text{weight} = -0.8$).

Pode ser verificado visualmente que tanto para os resultados da figura 4.30-A como para a figura 4.30-B houve um decréscimo nos valores de correlação, percebido em função da redução dos tons avermelhados na imagem. E tal fato pode ser confirmado analisando-se o máximo do valor de correlação obtido em cada imagem. Na imagem figura 4.30-A, que é resultante da correlação da imagem binarizada utilizando o método de Otsu [10], o máximo ocorreu com valor de aproximadamente 0,5445. Na imagem figura 4.30-B, que é resultante da correlação da imagem binarizada utilizando o método de Niblack [11], o máximo ocorreu com valor de aproximadamente 0,7206.

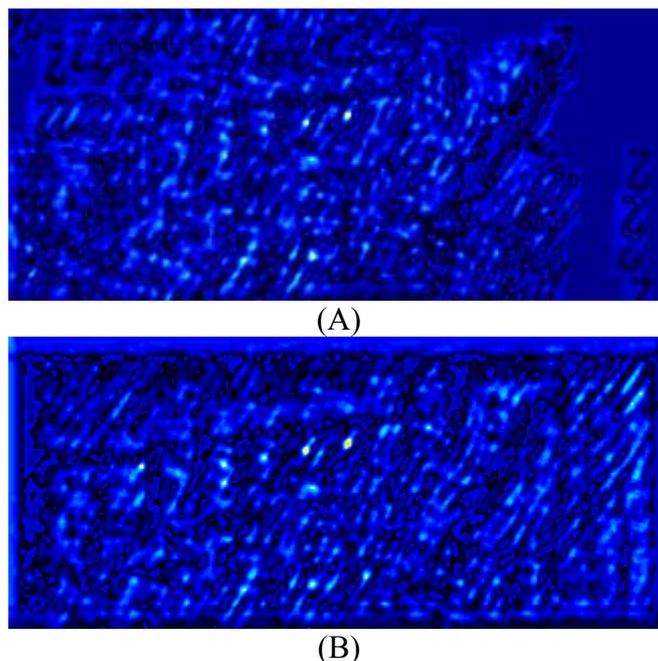


Figura 4.30 – Em (A): imagem resultante da correlação cruzada normalizada e mapeada em cores do padrão da figura 4.28 com a imagem da figura 4.29-A. Em (B): imagem resultante da correlação cruzada normalizada e mapeada em cores do padrão da figura 4.28 com a imagem da figura 4.29-B.

Apesar de reduzir naturalmente os valores de correlação máximos, a binarização feita de maneira a não descaracterizar a imagem pode tornar o sistema mais robusto, pois a correlação em tons de cinza é considerada pela literatura [14,22] como mais sensível a mudanças de iluminação ou mesmo contraste na imagem. No presente caso, a binarização local usando o método de Niblack [11] promoveu uma menor degradação nos dígitos na imagem. Uma das desvantagens do método de Niblack é a escolha dos parâmetros para realizar a binarização, que no método de Otsu é obtida automaticamente. A binarização feita em figura 4.29-B pode ter resultados melhores, promovendo o destaque apenas da região numérica com um aumento significativo do parâmetro N como demonstrado na figura 4.31. Porém, isso deve ser analisado em função de testes em diversas imagens, a fim de que o parâmetro possa ser mais bem ajustado à situação média de operação.

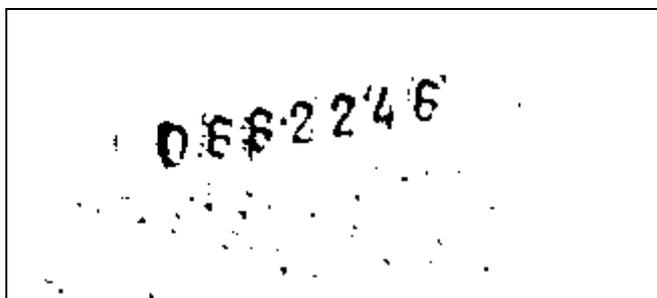


Figura 4.31 – Imagem resultante da binarização local proposta por Niblack [11] com parâmetros ($N = 200$ e $\text{weight} = -0.8$).

A partir desse ponto, é interessante analisar a redução do espaço de busca, proposta em 4.2.8. Ao contrário do processamento de correlação com imagem e padrão

em tons de cinza, que não altera seu resultado em função da etapa de redução do espaço de busca, a etapa de binarização modifica seu comportamento em função da região de imagem analisada como pode ser verificado na figura 4.32.

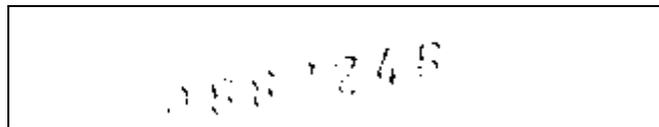


Figura 4.32 – Imagem resultante da binarização local proposta por Niblack [11] com parâmetros ($N = 200$ e $\text{weight} = -0.8$) após redução no espaço de busca.

4.2.11 – Comparação com o banco de padrões

Com relação ao presente trabalho, é de interesse identificar nas estampas de um tarugo os dígitos numéricos compreendidos no intervalo de 0 a 9. Para realizar tal tarefa, foram coletados de imagens teste da LIT que geraram o banco de padrões utilizado.

Uma deficiência da operação de correlação em processamento digital de imagens é que tal método possui sensibilidade à escala, ou seja, se fosse coletada uma região contendo um dígito da imagem, realizando-se uma redução no tamanho desse dígito, e então o correlacionando com a imagem original, será percebido uma redução drástica no valor de máximo de correlação. Uma maneira de se amenizar o efeito de escala é armazenar no banco de padrões amostras de diferentes escalas para cada dígito. O efeito negativo dessa estratégia é o aumento considerável do tempo de processamento proporcional ao número de imagens presentes no banco de padrões. Assim:

$$T = N_1 \cdot N_2 \cdot t_{\text{normcorr2}} \quad (4-1)$$

Onde:

- T → tempo total de processamento da etapa de comparação com o banco de padrões;
- N_1 → número de dígitos ou caracteres possíveis, no presente caso $N_1 = 10$;
- N_2 → número de padrões em diferentes escalas por dígito;
- $t_{\text{normcorr2}}$ → tempo de execução de cada operação de correlação cruzada normalizada.

Com relação à (4-1) pode-se afirmar que os únicos parâmetros que podem ser ajustados de maneira a interferir no desempenho global são os parâmetros N_2 e $t_{\text{normcorr2}}$. O aumento do parâmetro N_2 torna o sistema mais robusto, mas tem como consequência o aumento no tempo global de processamento. O parâmetro $t_{\text{normcorr2}}$ pode ser reduzido das seguintes formas:

1. Reduzindo o tamanho da maior imagem a ser correlacionada, assim como é feito com a redução do espaço de busca, reduzindo drasticamente o tempo total;
2. Realizando otimizações na parte de compilação da biblioteca de fft utilizada [6], a fim de torná-la específica para uma determinada estação de trabalho alvo. Além disso, também é possível aumentar ainda mais o desempenho fazendo-se

uso de um recurso da biblioteca denominado “fftw3 wisdom” onde são armazenados pré-processamentos que agilizam o uso de suas funções;

3. Aumentado a capacidade de processamento da estação de trabalho em que será instalado o sistema.

Em termos de desempenho geral, a concepção original era de que fosse possível para o sistema realizar todo o processamento em um tempo de 5 a 10 segundos. Diante de tal fato é possível prever qual o valor desejado para $t_{normxcorr2}$ fazendo uma aproximação de que $T \approx T_{total}$, levando-se em conta que a etapa de comparação é definitivamente a etapa mais lenta de todo o processo.

Na tabela 4.1 podem ser vistos os tempos máximos de processamento desejados para a etapa de correlação cruzada dependente dos requisitos impostos pelo processo e variando em função do número de padrões escolhidos para representar cada dígito.

Tabela 4.1 – Evolução do tempo de processamento com o aumento do número de padrões por dígito

N_2	$t_{normxcorr2}$ (seg)	
	(T = 5 seg.)	(T = 10 seg.)
1	0,5000	1,0000
2	0,2500	0,5000
3	0,1667	0,3333
4	0,1250	0,2500
5	0,1000	0,2000

Além do tempo de processamento, a escolha do banco de padrões é crucial para determinar o índice de acertos final do sistema.

4.2.12 – Montagem da seqüência numérica reconhecida

Após a etapa de correlação com os padrões, ou casamento de padrões, obtêm-se valores de picos de correlação distintos e suas coordenadas espaciais. Todos os valores obtidos atendem ao critério configurado anteriormente de possuírem um valor de correlação acima do limiar (normalmente limiar $\geq 0,6$). Além disso, todos os picos estão distantes uns dos outros de uma distância mínima de $2r$, onde r também é um parâmetro configurável e arbitrado pelo usuário.

Três possíveis situações são esperadas a partir de agora:

1. *Nenhuma das matrizes de correlação retornou valor acima do limiar desejado* – assim sendo, então o sistema deve sinalizar um alarme requisitando auxílio ao usuário;
2. *Foi encontrado um número de picos candidatos inferior ao mínimo esperado* – atualmente, as estampas dos tarugos possuem 7 dígitos impressos que representam o número da ordem de produção. Se o número de candidatos for inferior ao número de dígitos, então isso deverá ser refletido na confiança global da identificação. Essa questão será discutida mais a frente;

3. Foi encontrado um número de picos superior ao mínimo esperado – diante dessa situação, o sistema escolhe os 7 picos com maiores valores de correlação dentre os candidatos.

Diante das possíveis situações esperadas pelo sistema, foi arbitrada uma maneira de associar uma confiança global ao processo de reconhecimento. Essa confiança global é função do número de dígitos reconhecidos e dos valores de correlação para cada candidato encontrado. Assim sendo, definiu-se:

$$C_g = \frac{n}{N_d} \cdot \sqrt[n]{c_1 \cdot c_2 \cdot \dots \cdot c_n} \quad (4-2)$$

Onde:

- C_g → confiança global associada ao reconhecimento;
 n → número de dígitos reconhecidos em seqüência de uma OP agendada;
 N_d → número esperado de dígitos (atualmente 7);
 c_i → valor de correlação do pico “i”.

A EQ. (4-2) evidencia que a confiança global deve ser proporcional ao número de dígitos reconhecidos e também à média geométrica dos valores de correlação encontrados. Por exemplo, em um caso em que se tenha encontrado 3 dígitos com $c_1=c_2=c_3=1$ a confiança global é $n/N_d = 3/7 \approx 0,429$. A média geométrica foi escolhida porque no seu caso valores elevados de correlação não influenciam tanto o resultado diante de uma comparação com a média aritmética. O “mascaramento” proporcionado pela média aritmética pode ser verificado na figura 4.33 onde pode-se notar que os valores de média estão sempre mais próximos da menor medida.

No gráfico da figura 4.33 é feita a média entre “a” e “b”, onde $a = 0,6$ e b varia no intervalo de 0 a 1. Pode ser verificado que a média geométrica fica mais próxima dos menores valores da média, o que para o sistema em questão é interessante, pois evita reconhecimentos indevidos durante sua operação.

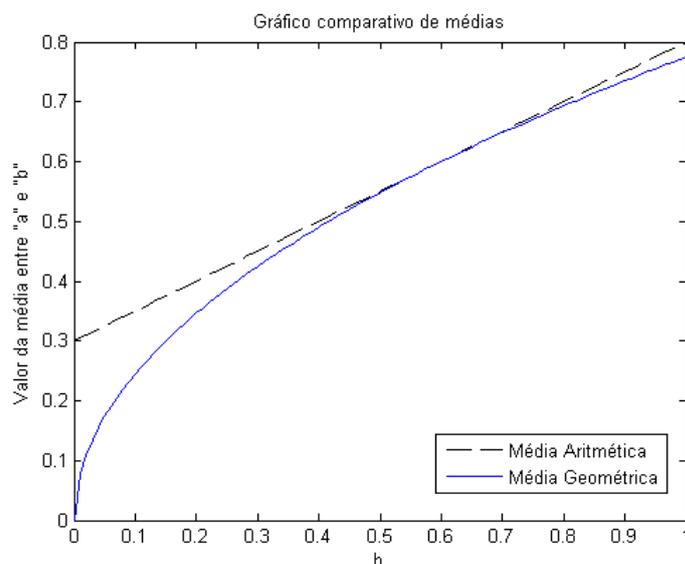


Figura 4.33 – Comparativo do desempenho da média geométrica versus média aritmética.

O próximo passo do processamento de identificação é encontrar o parâmetro “n” especificado na EQ. (4-2). É necessário evidenciar o fato de que, durante a escolha dos picos de correlação, há uma ordenação de maneira que eles fiquem organizados segundo a coordenada do eixo das abscissas. Portanto, quando é retornada a seqüência numérica reconhecida, os dígitos se encontram nas coordenadas espaciais devidas. Tomando um exemplo onde o sistema retorna uma seqüência reconhecida “626”, isso significa que a ordem de produção identificada possui tais caracteres em seqüência, sendo que isso possa significar qualquer número do tipo “x626xxx”, “x6x2x6x”, ou “xx62xx6”. A informação de coordenadas espaciais dos dígitos apenas garante a ordem dos mesmos na imagem, e isso leva a procurar na lista de ordens de produção por seqüências numéricas que possuam aqueles dígitos identificados em qualquer combinação possível, desde que eles estejam em seqüência. Após isso, é estimado um valor de confiança para o reconhecimento a partir da EQ. (4-2). O parâmetro “n” é então o número de dígitos reconhecidos que foram encontrados em uma ordem de produção agendada.

A identificação de um número menor de dígitos do que o esperado se refletirá consequentemente na confiança final do sistema para o candidato identificado, como pode ser visto na figura 4.34.

- seqüência identificada:

6 - $c_1 = 0,6$

2 - $c_2 = 0,8$

6 - $c_3 = 0,65$

- lista de ordens de produção agendadas:

0741269 $n = 2$

0682246 ou 0682246 $n = 3$

0543232 ou 0543232 $n = 1$

0111111 $n = 0$

0317266 $n = 2$

- ordem de produção identificada:

0682246 - $\text{confiança} = 3/7 * (0,6 * 0,8 * 0,65)^{1/3} = 0,2907$

Figura 4.34 – Exemplo de reconhecimento e cálculo da confiança global do sistema para a ordem de produção identificada

Em função dos padrões escolhidos ou da própria impressão com pouca definição feita nos tarugos, a identificação de um número de caracteres inferior ao número de dígitos impressos se torna um fato bastante comum. Existe a possibilidade de no final do processamento ser encontrados vários candidatos a ordem de produção atual. Para evitar erros de reconhecimento, pode ser adicionada uma informação redundante que é o tamanho da bitola do tarugo. O tamanho da bitola é fixo para todos os tarugos de uma mesma ordem de produção, e pode ser estimado pelas metodologias empregadas no item 4.2.6. Para realizar tal processamento, é necessário que na lista de ordens de produção diária, que atualmente é um dos arquivos de configuração do sistema, esteja contida a informação do tamanho de bitola, para cada ordem de produção. Outra tarefa importante é a configuração, por parte do usuário, da relação pixel/milímetro, para haver uma conversão de unidades da imagem para dimensões reais.

4.2.13 – Planejamento do software de identificação

O primeiro passo para o desenvolvimento de sistemas de softwares que possuem tempo de vida relativamente longo é pensar em como gerenciar a sua complexidade. Uma das abordagens possíveis é organizar o código utilizando metodologias Orientadas a Objetos.

O uso de metodologias orientadas a objetos tem como objetivos:

- Gerar ilusão de simplicidade – tornar fácil o uso e reuso de componentes do software;
- Decomposição – promover uma separação organizada de um sistema maior em vários subsistemas, de maneira que fiquem mais fáceis sua análise e manipulação;
- Abstração – representar problemas com modelos muitas vezes desprezando detalhes não essenciais;
- Hierarquia
 - Herança – forma de reutilização de código de maneira incremental: cada “nova geração” de objetos herda funcionalidades de seus antecessores;
 - Composição – gerar objetos e comportamentos complexos baseados em componentes primitivos mais simples e suas interações;

Além disso, é possível utilizar ferramentas como a linguagem UML (Unified Modeling Language) para projetar e analisar o comportamento de um sistema complexo de softwares [9].

A ferramenta científica MATLAB propicia um ambiente muito cômodo para testes e desenvolvimentos de rotinas. Entretanto, o presente trabalho necessitava de uma aplicação que fosse rápida, eficiente, robusta, e que fosse possível realizar diagnósticos em tempo real. O MATLAB não atende a todos esses requisitos, sendo então encarado, no presente trabalho, como apenas uma forma de desenvolvimento de estratégias.

O sistema foi implementado em Visual C++, idealizado para conter 3 módulos:

1. Módulo de comunicação com o sistema de automação;
2. Módulo de processamento de imagens;
3. Módulo de controle da câmera utilizando o protocolo PELCO [4].

O módulo 1 ainda não foi desenvolvido, pois algumas das informações que devem ser obtidas do sistema de automação estão atualmente sendo obtidas por meio de arquivos textos de configuração. Entretanto, esse módulo é necessário para que o sistema fique “*online*” de maneira a recuperar a lista de ordens de produção diárias, além de receber o comando de captura de imagem por meio do sensor instalado na linha e também fazer a parametrização da linha de inspeção de tarugos quando uma nova ordem de produção agendada for identificada.

O módulo 2 possui resumidamente os seguintes componentes:

1. Classes que replicam as facilidades do MATLAB, como operações matriciais, álgebra linear, etc.;
2. Biblioteca de manipulação de imagens (CxImage [5]) com funções úteis como carregamento de diferentes tipos de arquivos de imagens, interpolação, filtragens simples, etc.;
3. Biblioteca de processamento rápido e eficiente da DFT (Discrete Fourier Transform) (FFTW [6]), que é a mesma biblioteca utilizada pelo MATLAB;
4. Funções de processamento de imagem genérico como as de correlação cruzada normalizada utilizando a FFTW [6], binarização automática (OTSU [10] ou NIBLACK [11]), etc.;
5. Rotinas espelho das desenvolvidas em MATLAB, que têm como foco específico o problema da identificação do tarugo, como as rotinas de segmentação do tarugo, rotação, etc.

Algumas rotinas, como a nova estratégia de segmentação do tarugo, a redução de espaço de busca e a binarização utilizando o método Niblack [11] ainda não foram convertidas do ambiente MATLAB para o programa em Visual C++.

O módulo 3 é responsável pela comunicação e comando da câmera PELCO/PTZ. Existem dois tipos de protocolos utilizados em equipamentos desse tipo, o protocolo tipo “P” e o “D”, e ambos têm suas mensagens descritas em [4]. Foi implementado o Protocolo P, tentando comandar a câmera por meio da porta serial do computador ligada ao sistema da câmera por um conversor (RS232C para RS485/422). Testes preliminares foram realizados sem muito sucesso, e como a estratégia de identificação era mais importante, o comando por meio da comunicação serial foi deixado para trabalhos futuros, embora uma possível comunicação com a câmera pudesse melhorar o desempenho do sistema. Controlar o zoom e posicionamento adiciona mais tempo ao processamento, mas pode aumentar a confiança do reconhecimento.

A figura 4.35 mostra um diagrama de seqüência UML do software planejado, sendo que algumas etapas ainda não foram implantadas, evidenciando as etapas que se seguem a partir da execução da operação e como as principais classes e componentes se interagem a fim de fornecer o reconhecimento final.

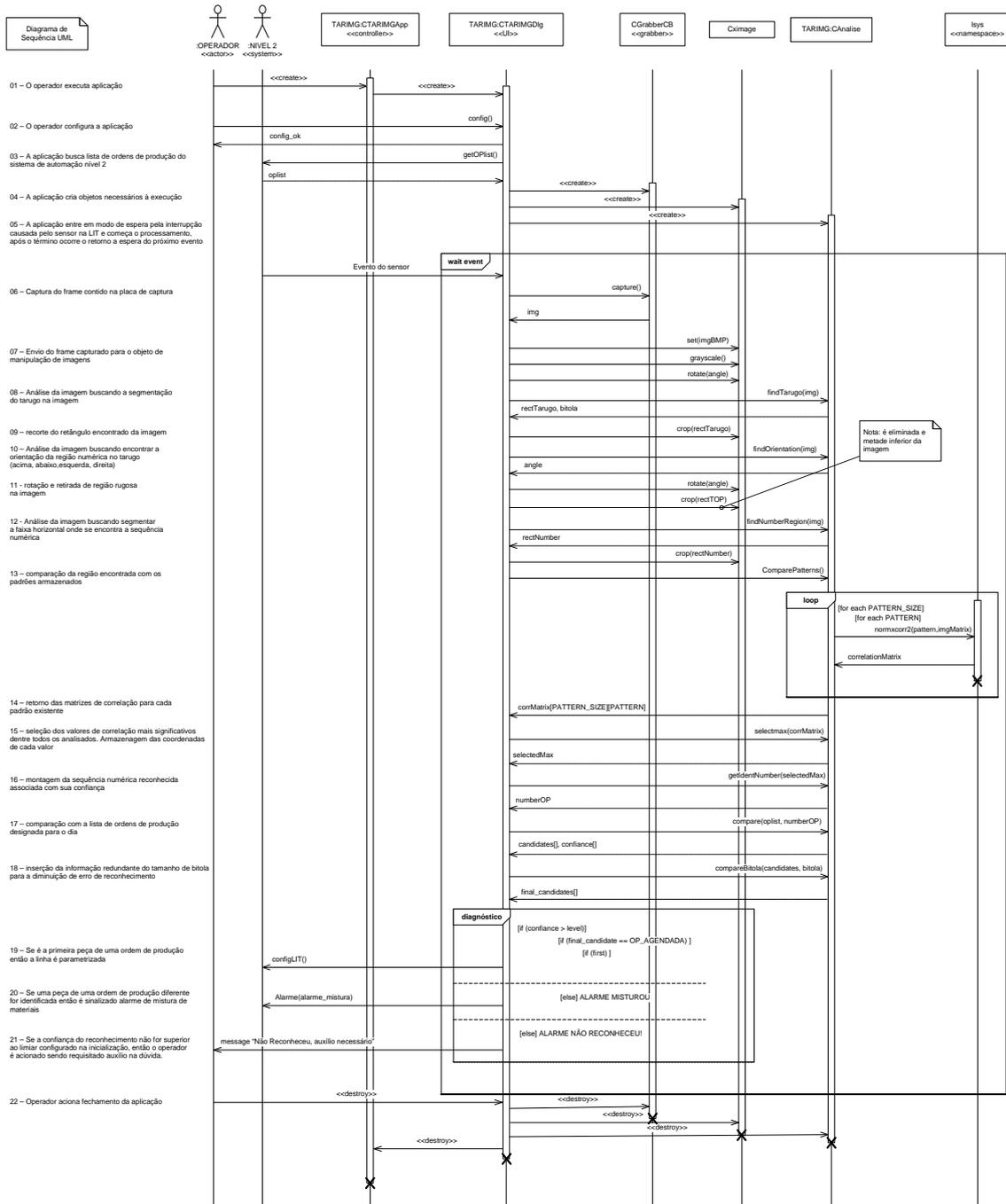


Figura 4.35 – Diagrama (resumido) de seqüência UML do software planejado

4.3 – Conclusões

No presente capítulo, são apresentadas as metodologias efetivamente aplicadas visando desenvolver um sistema consistente de processamento de imagens. Primeiramente foram descritos os equipamentos que compõe a linha de inspeção de tarugos. Após isso são propostas melhorias no processo. Finalmente é demonstrado o funcionamento de cada etapa de processamento de imagens escolhida especificamente para o presente problema.

5 – RESULTADOS E DISCUSSÕES

Contando com a colaboração dos funcionários da AÇOMINAS S/A, foi possível coletar uma quantidade razoável de imagens. O processo de coleta de imagens tomou o seguinte rumo:

1. **Coleta de imagens sem iluminação e sem uma preocupação com a escala numérica na imagem** – foi o primeiro passo tomado no processo de análise do problema. Foram coletadas **101** imagens correspondentes a OP 0668687, seguindo uma metodologia que visava adquirir imagens do mesmo tarugo em 3 enquadramentos distintos. Não existe a garantia e a preocupação com o fator de escala no momento, pois o problema ainda não tinha sido detectado. O vislumbre inicial era de que é interessante para o sistema detectar onde está a face do tarugo na cena. Em seguida deve ser enquadrada totalmente essa região da face. Após o enquadramento da face deve ser localizada a seqüência numérica e seu conseqüente enquadramento. Todos os posicionamentos são feitos pelo sistema da câmera, só que para realizar essa tarefa automaticamente é necessária uma comunicação entre os controladores da câmera e o PC. Além disso, o tempo da dinâmica associada à movimentação pode interferir significativamente no tempo total de processamento. Uma conseqüência dessa coleta inicial de imagens foi a conclusão que é inviável trabalhar sem um sistema de iluminação adequado, pois as imagens não oferecem bom contraste nesses casos. Na figura 5.1 podem ser vistas várias posições de enquadramento da face do tarugo sem condicionamento de iluminação e sem preocupação com a escala numérica na imagem;



(A)

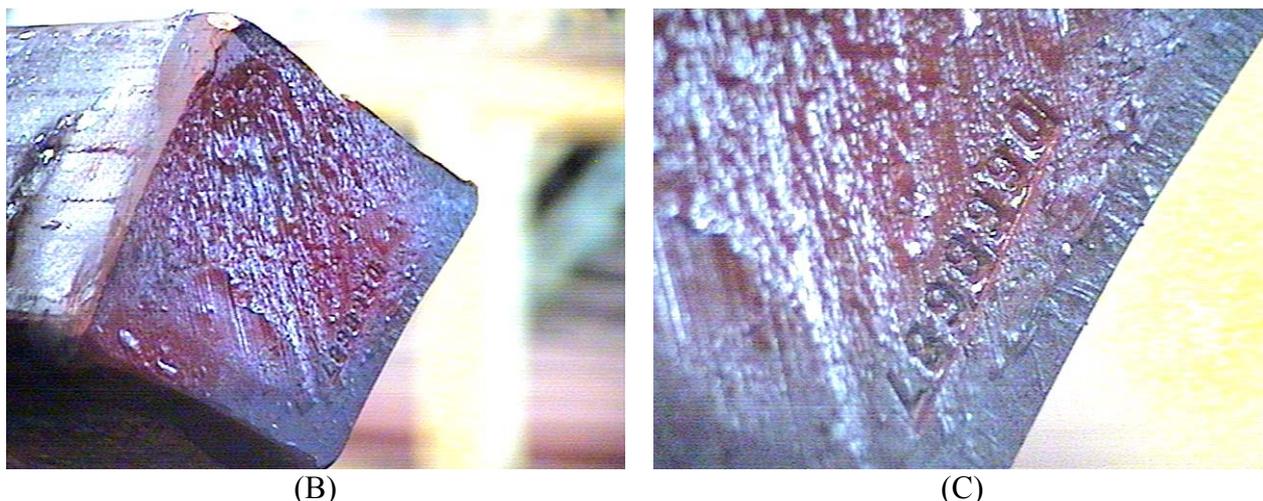
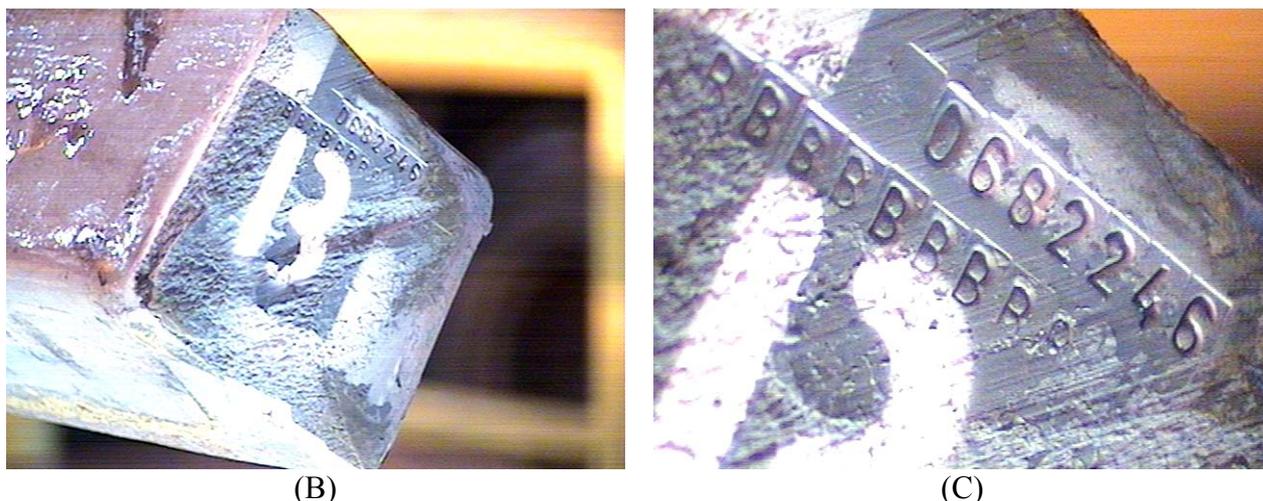


Figura 5.1 – Em (A): Enquadramento da cena. Em (B): Enquadramento da face do tarugo. Em (C): Enquadramento da região numérica. OBS: procedimentos feitos manualmente

- 2. Coleta de imagens com iluminação adequada e sem preocupação com a escala numérica na imagem** – o esquema de iluminação proposto melhorou sensivelmente a qualidade das imagens, adicionando maior contraste em regiões de rugosidade e da estampa. Neste ensaio foram coletadas **85** imagens no total, correspondentes às OP's 0679498, 0682246 e 0684271. Uma característica marcante nesse ensaio é a de que sujeitas a uma iluminação do tipo fluorescente a face do tarugo aparece com tons azulados nas imagens capturadas. A maior parte dessas imagens não possuía pintura na face, exceto em poucos casos onde se demarcava “T”(peça de topo) ou “B”(peça de base) utilizando tinta branca. Na figura 5.2 podem ser vistas várias posições de enquadramento da face do tarugo na situação de iluminação adequada, mas sem preocupação em manter uma mesma escala numérica na imagem.



(A)



(B) (C)
Figura 5.2 – Em (A): Enquadramento da cena. Em (B): Enquadramento da face do tarugo. Em (C): Enquadramento da região numérica. OBS: procedimentos feitos manualmente e peça pintada como “B”(base)

- 3. Coleta de imagens com iluminação adequada e com preocupação com a escala numérica na imagem** – após testes preliminares utilizando imagens coletadas com iluminação adequada percebeu-se que a dimensão com que a estampa numérica aparece na imagem deve ser aproximadamente a mesma em todas as imagens, para que metodologias de reconhecimento de padrões baseadas em correlação funcionassem corretamente. Foi então fixada uma posição de enquadramento da câmera de maneira que somente o tarugo de 160 mm de aresta ocupará toda a imagem. Em seguida foram coletadas **189** imagens das OP's 0741269, 0741281 e 0741626 que por acaso possuíam pintura adicional nas faces (aço para forjaria). A estratégia de pintar faces de tarugos com cores distintas é uma tentativa de evitar misturas de peças no caso de metais utilizados em processos mais críticos. Na figura 5.3 pode ser verificada uma imagem com condicionamento de iluminação adequado e com preocupação de manter certa escala numérica.

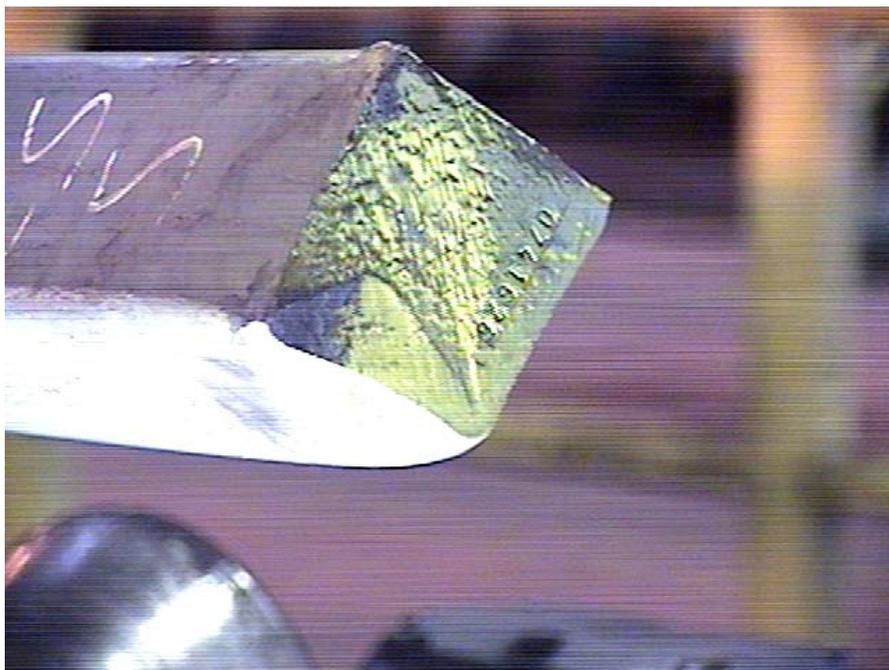


Figura 5.3 – Tarugo da ordem de produção 0741626 com pintura amarela em sua face

Apesar do número de imagens coletadas ser relativamente alto a quantidade de informação útil contida é bastante reduzida, elas não forneceram material suficiente para a tarefa de criação do banco de padrões de comparação. Outro fator interessante na análise é de que, mesmo para um ser humano, boa parte das imagens não fornece informação suficiente para que seja identificado com certeza o número da OP na imagem. Imagens da terceira coleta sofreram efeito de ruído, e por isso foram adquiridas várias imagens do mesmo tarugo para tentar, utilizando uma média aritmética dos *pixels*, reduzir o efeito do ruído.

Antes de ser realizada qualquer análise foram selecionados 2 grupos de imagens que permitem a um ser humano identificar algumas características básicas do tarugo, como localização da face e seqüência numérica da OP. As características dos grupos escolhidos são:

- Grupo 1 (G1) – 25 imagens selecionadas da segunda coleta com a face do tarugo quase que totalmente enquadrada na imagem sem preocupação com a escala que os números ocorrem na imagem. Algumas imagens possuem pinturas brancas na face para identificar topo ou base, mas não possuem pintura especial de outra cor;
- Grupo 2 (G2) – 41 imagens selecionadas da terceira coleta, sendo que a maioria foi descartada por excesso de ruído, mesmo tentando eliminá-lo com média entre imagens.

5.1 – Correção da rotação conhecida

Nas figuras 5.4-A e 5.4-B podem ser vistos resultados da etapa inicial de correção da rotação conhecida de maneira que as arestas do tarugo fiquem aproximadamente paralelas às arestas da imagem.

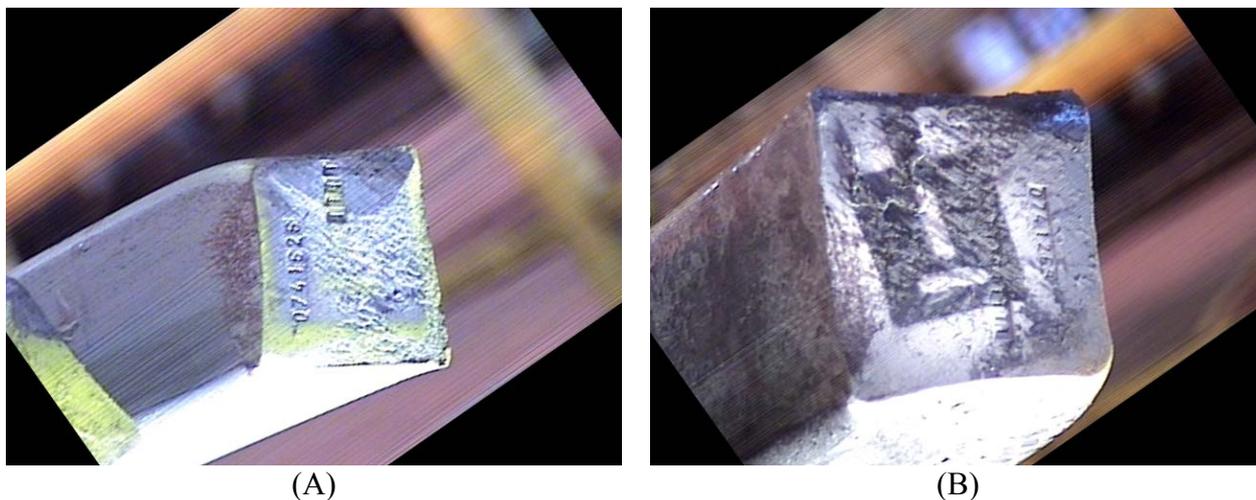


Figura 5.4 – Exemplos de imagens com rotação corrigida (rotação com interpolação bilinear)

Todas as imagens inicialmente passam por um processo de rotação conhecida, supondo um ângulo de rotação de 35° no sentido anti-horário. Em média, a rotação gera imagens bem ajustadas. Em alguns casos a rotação pode não deixar a seqüência numérica totalmente na vertical ou horizontal, isso poderá influir nas etapas seguintes, principalmente na de comparação com os padrões, pois a correlação também é sensível à rotação.

5.2 – Segmentação do tarugo na cena

A estratégia de segmentação do tarugo na cena inicialmente desenvolvida em ambiente MATLAB por Ramalho [1] e implantada em Visual C++ no presente trabalho mostrou-se bastante eficiente em imagens com nenhuma ou pouca pintura sobre a face do tarugo. A estratégia inicial, no entanto, não apresenta bons resultados se a imagem de entrada tem alguma pintura sobre a face do tarugo. A estratégia utilizando perfis de variância na imagem foi adotada para tentar resolver tal problema. A seguir são demonstrados os resultados de cada estratégia.

5.2.1 – Estratégia inicial implantada

A definição de acerto nessa etapa é um tanto quanto subjetiva, basicamente é considerado acerto se a imagem recortada resultante preservasse a seqüência numérica e não excedesse demasiadamente na estimação da localização das arestas na face do

tarugo. A tabela 5.1 mostra o índice de acertos da metodologia inicial de segmentação do tarugo na cena.

Tabela 5.1 – Acertos da metodologia para cada grupo de imagens

IMAGENS	Acerto da Etapa (%)
GRUPO 1	96,00
GRUPO 2	48,78

As figuras 5.5, 5.6 e 5.7 exemplificam imagens resultantes da metodologia de segmentação.



Figura 5.5 – Recorte resultante da metodologia com erro desprezível (ACERTO)

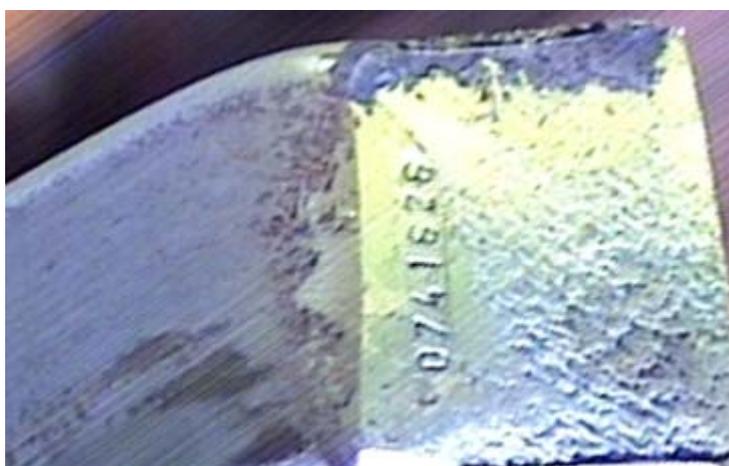


Figura 5.6 – Recorte resultante da metodologia com área além da necessária (ERRO)



Figura 5.7 – Recorte resultante da metodologia com área inferior à necessária (ERRO)

Como era de se esperar, essa alternativa funciona muito bem para imagens com a face predominantemente azulada (GRUPO 1), pois trata-se de uma estratégia de segmentação baseada em cores.

5.2.2 – Alternativa utilizando perfil de variância

A figura 5.8 exemplifica o resultado da detecção de bordas em uma imagem coletada para um caso de identificação bem sucedida.

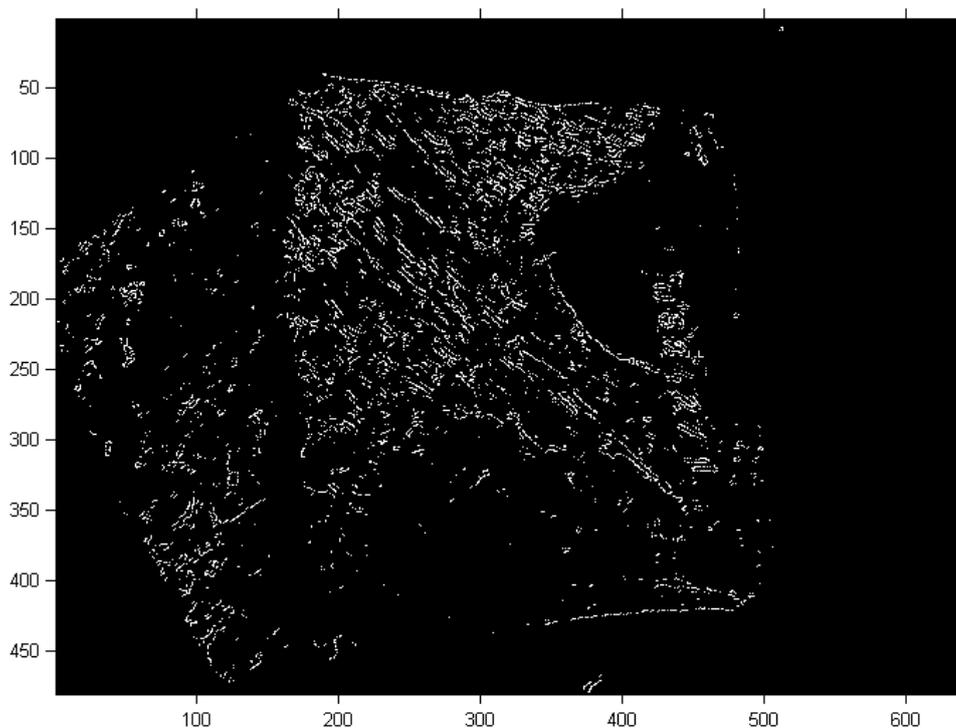


Figura 5.8 – Detecção de bordas usando operador de Sobel e limiar automático (ACERTO)

A figura 5.9 mostra o perfil de variância resultante a partir da imagem da figura 5.8. A figura 5.10 mostra o resultado do diagnóstico nebuloso para o perfil exposto na figura 5.8.

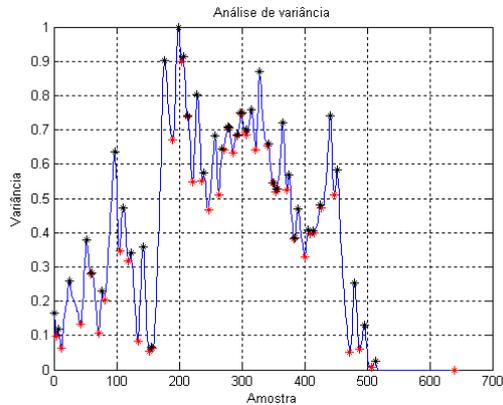


Figura 5.9 – Perfil de variância das linhas horizontais da imagem

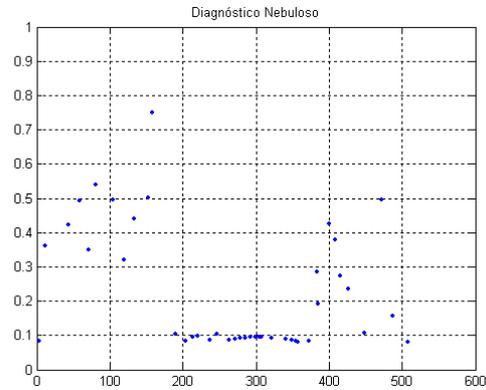


Figura 5.10 – Diagnóstico nebuloso indicando limites

A partir da análise do perfil de variância pelo sistema de inferência nebuloso é possível determinar os limites que indicam a presença da face do tarugo.

A figura 5.11 exemplifica o resultado da detecção de bordas em uma imagem coletada para o caso de uma identificação duvidosa.

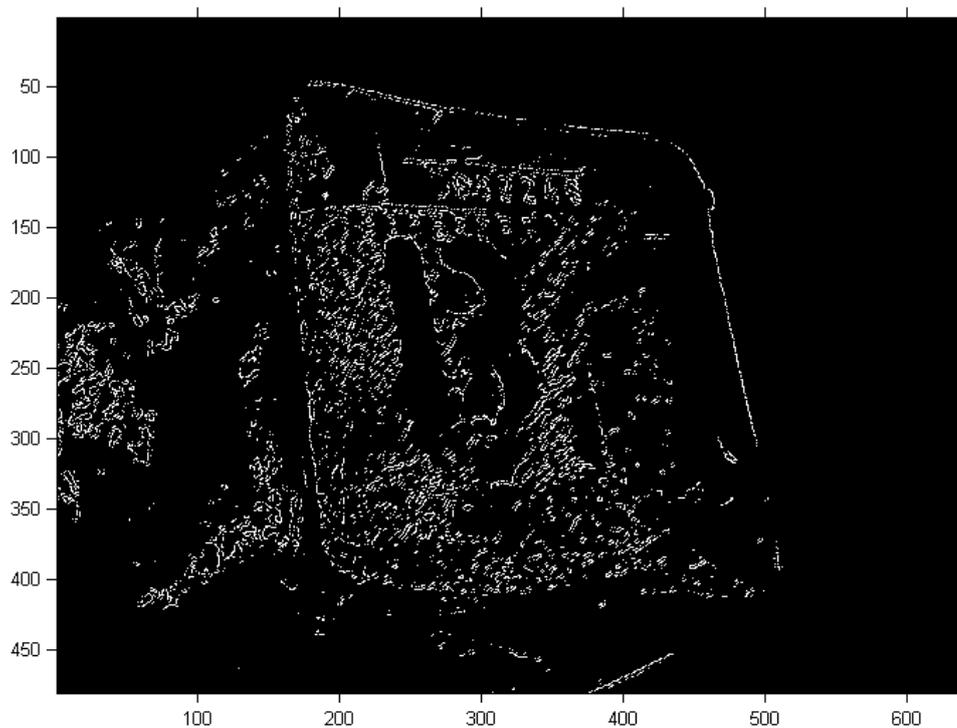


Figura 5.11 – Detecção de bordas usando operador de Sobel [12] e limiar automático (ERRO)

A figura 5.12 mostra o perfil de variância resultante a partir da imagem da figura 5.11. A figura 5.13 mostra o resultado do diagnóstico nebuloso para o perfil exposto na figura 5.11.

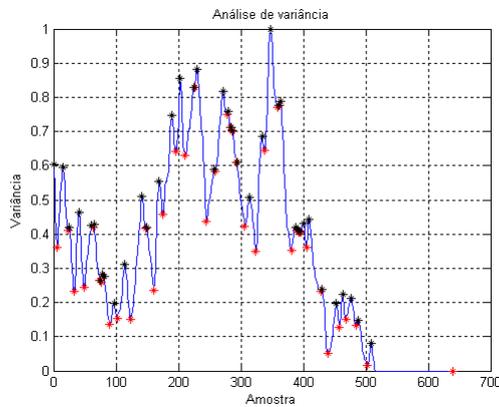


Figura 5.12 – Perfil de variância das linhas horizontais da imagem

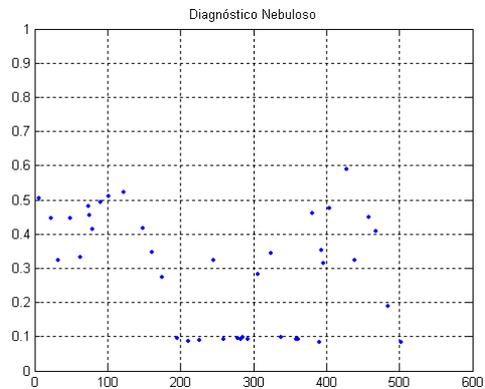


Figura 5.13 – Diagnóstico nebuloso indicando limites

Em alguns momentos o perfil de variância não oferece informações suficientes para uma delimitação correta da face. O diagnóstico errado por parte do sistema nebuloso é consequência de uma detecção de bordas sem muita qualidade, que gerará um perfil com informações insuficientes. Outros problemas estão relacionados à proximidade de outros tarugos que pode influenciar na localização dos desses limites. A figura 5.14 mostra um exemplo de falha na identificação.

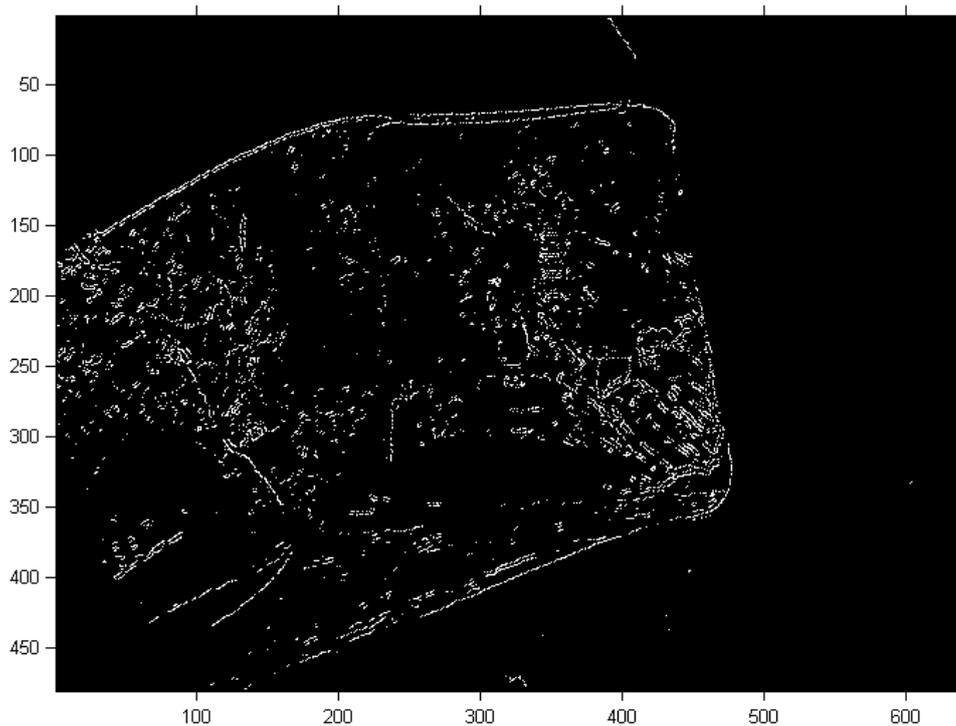


Figura 5.14 – Detecção de bordas usando operador de Sobel [12] e limiar automático (ERRO)

A figura 5.15 mostra o perfil de variância resultante a partir da imagem da figura 5.14. A figura 5.16 mostra o resultado do diagnóstico nebuloso para o perfil exposto na figura 5.14.

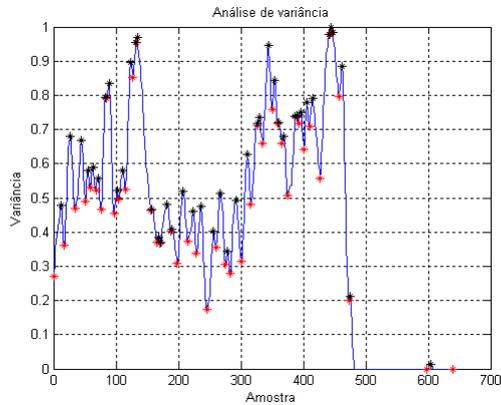


Figura 5.15 – Perfil de variância das linhas verticais da imagem

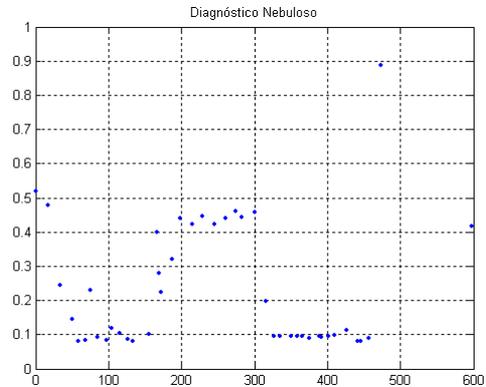


Figura 5.16 – Diagnóstico nebuloso indicando limites

A figura 5.17 mostra um exemplo de falha na identificação devido à falta de informação na região das bordas superiores, mesmo assim a metodologia indicou corretamente os limites indicados pelo perfil de variância.

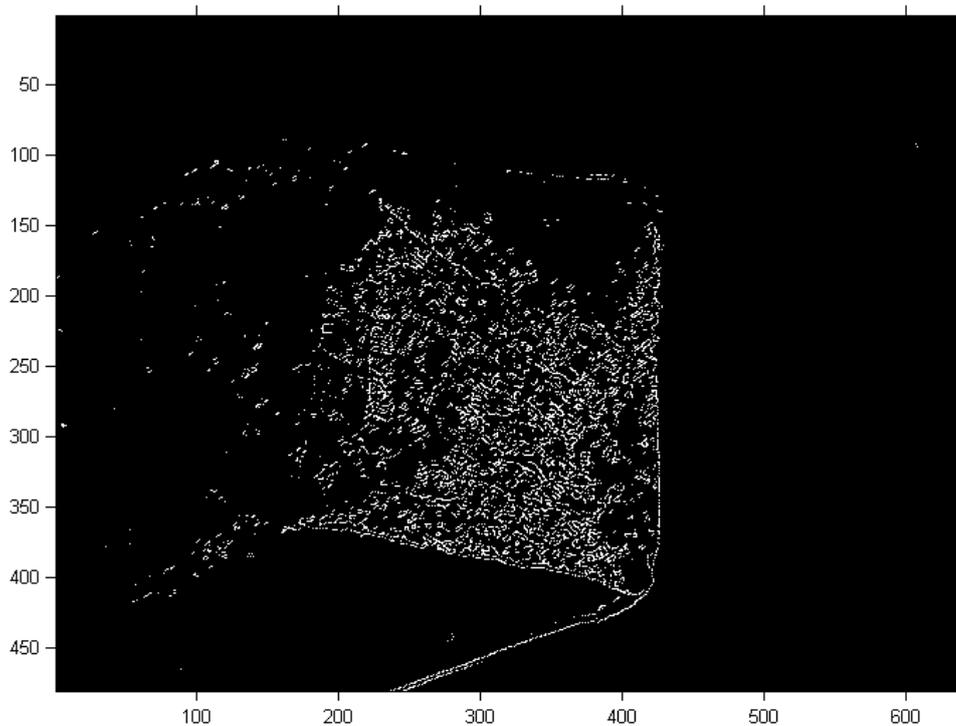


Figura 5.17 – Detecção de bordas na FIG.13-C usando operador de Sobel [12] e limiar automático (eixos coordenados conforme convenção ditada pela varredura de vídeo)

A figura 5.18 mostra o perfil de variância resultante a partir da imagem da figura 5.17. A figura 5.19 mostra o resultado do diagnóstico nebuloso para o perfil exposto na figura 5.17.

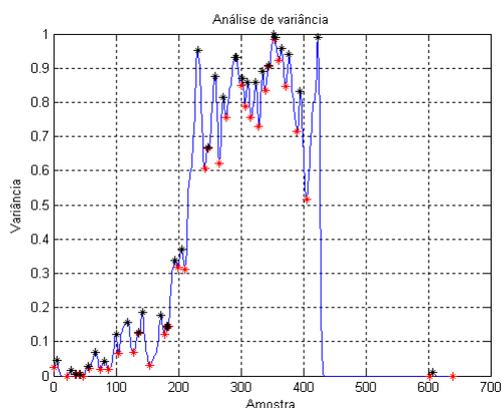


Figura 5.18 – Perfil de variância das linhas verticais da imagem

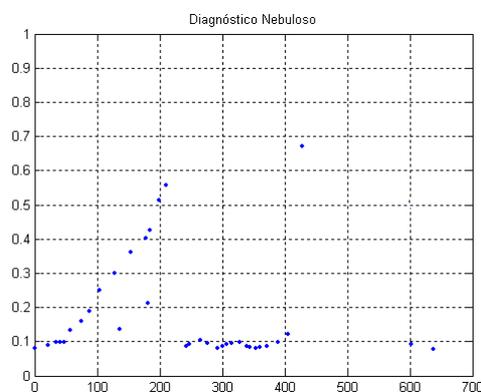


Figura 5.19 – Diagnóstico nebuloso indicando limites

A tabela 5.2 mostra o índice de acertos da metodologia de segmentação baseada em análise do perfil de variância e diagnóstico nebuloso.

Tabela 5.2 – Acertos da metodologia para cada grupo de imagens

IMAGENS	Acerto da Etapa (%)
GRUPO 1	68,00
GRUPO 2	85,37

5.3 – Rotação

A etapa de rotação espera receber um tarugo recortado adequadamente e, baseado nas áreas de rugosidades de um tarugo conseguir estimar a localização da seqüência numérica (direita, esquerda, acima ou abaixo). Excluindo-se as imagens que já foram recebidas como incorretas da última etapa temos os níveis de acerto indicados pela tabela 5.3.

Tabela 5.3 – Acertos da metodologia para cada grupo de imagens

IMAGENS	Acerto da Etapa (%)
GRUPO 1	88,24
GRUPO 2	91,43

A correção de rotação é influenciada por pinturas irregulares na face do tarugo, que podem mascarar rugosidades em sua face. Uma maior discussão desse processo e seus índices de acerto foram feitos de maneira mais aprofundada por Ramalho [1].

5.4 – Redução do espaço de busca

A Etapa de redução de espaço de busca tem como objetivo único o de diminuir a imagem onde serão procurados padrões semelhantes aos presentes na base de dados. Essa tarefa é de fundamental importância, tem a função de aliviar a tarefa seguinte, pois o tempo de processamento na etapa de correlação é drasticamente influenciado pelo tamanho das imagens analisadas. O critério para considerar que houve sucesso ou não foi baseado no fato de que ocorreu uma redução na imagem sem a perda da região numérica. A figura 5.20 mostra um exemplo de imagem que sofrerá redução de tamanho preservando-se a região numérica, o resultado da metodologia para esse caso é positivo.

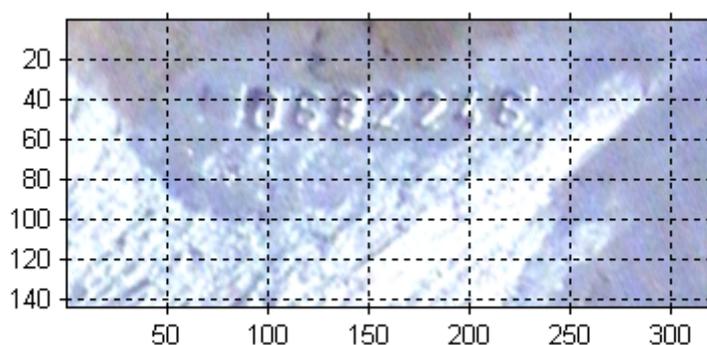


Figura 5.20 – Imagem já devidamente rotacionada e recortada.

A figura 5.21 mostra o perfil de variância resultante a partir da imagem da figura 5.20. A figura 5.22 mostra o resultado do diagnóstico nebuloso para o perfil exposto na figura 5.20.

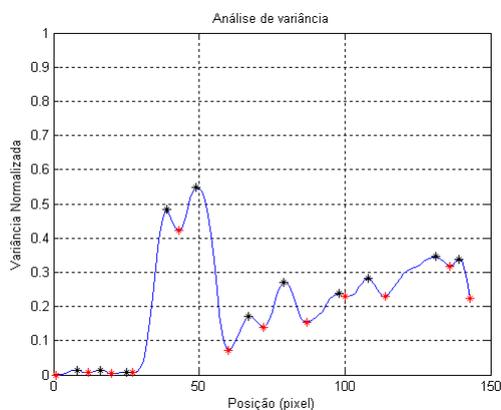


Figura 5.21 – Perfil de variância das linhas horizontais

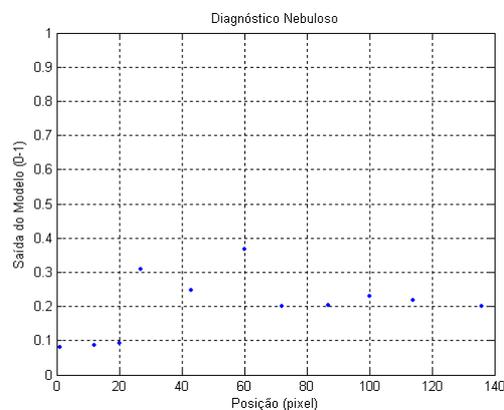


Figura 5.22 – Diagnóstico nebuloso indicando limites da região numérica

Em alguns casos, se não existir um contraste suficiente, a detecção da faixa horizontal que compreende a região numérica não será possível. Esse fato será refletido em valores, ínfimos, presentes no vetor de perfil de variâncias. A figura 5.23 mostra um exemplo onde a metodologia não conseguirá identificar a região numérica na imagem.

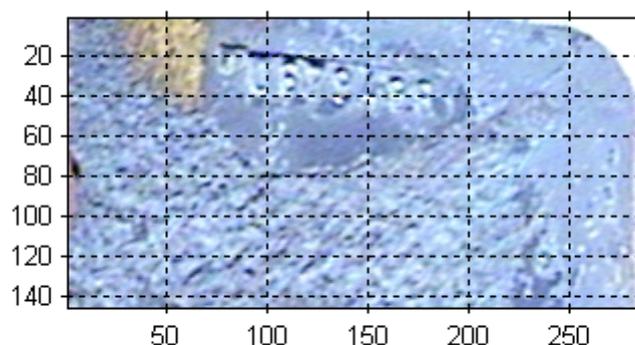


Figura 5.23 – Imagem já devidamente rotacionada e recortada

A figura 5.24 mostra o perfil de variância resultante a partir da imagem da figura 5.23. A figura 5.25 mostra o resultado do diagnóstico nebuloso para o perfil exposto na figura 5.23.

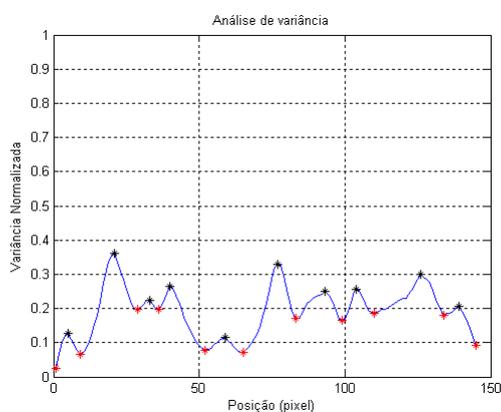


Figura 5.24 – Perfil de variância das linhas horizontais

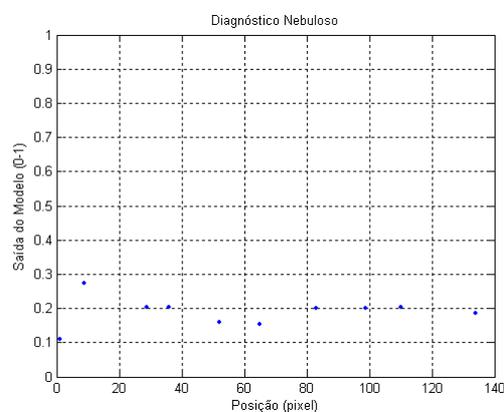


Figura 5.25 – Diagnóstico nebuloso indicando limites da região numérica

A tabela 5.4 mostra o índice de acertos da metodologia de redução do espaço de busca baseada em análise do perfil de variância e diagnóstico nebuloso.

Tabela 5.4 – Acertos da metodologia para cada grupo de imagens

IMAGENS	Acerto da Etapa (%)
GRUPO 1	80,00
GRUPO 2	75,00

5.5 – Correlação cruzada normalizada após binarização

Diante das imagens resultantes das etapas anteriores é realizada a identificação numérica propriamente dita. A identificação foi feita criando-se um banco de padrões composto por fragmentos de imagens coletados das próprias imagens de teste, para tentar conseguir um índice de acerto maior comparando-se com padrões gerados artificialmente. Caracteres como “3” e “5” não foram encontrados nas amostras das OP’s. O idealizado era de que o banco de padrões possuísse 4 exemplos de caractere para cada dígito de 0 a 9, ou seja, 40 padrões no total. A diferença entre os padrões é basicamente a escala, que é um artifício para minimizar efeitos de variação de escala. A inserção de muitos padrões no sistema tem como consequência o aumento no tempo de processamento, trata-se de uma razão de compromisso entre eficiência e confiabilidade no reconhecimento.

Como pode ser percebido pelas tabelas de acerto no reconhecimento (Tabelas 5.5 e 5.6), a maior parte dos caracteres foi reconhecida com índices insatisfatórios. Tal fato se deve a diversos fatores, dentre eles se destacam:

1. Qualidade da imagem – sem uma boa definição na imagem fica inviável o reconhecimento de padrões;
2. Alta correlação de alguns padrões com artefatos presentes na imagem – o caractere “1”, por exemplo, devido ao formato escolhido pela estampagem possuem correlação alta com qualquer risco vertical presente na imagem;
3. Correlação alta entre caracteres – caracteres como “0”, “3” (não está presente nas imagens coletadas), “6”, “8” e “9” possuem alta similaridade, sendo que qualquer falha na impressão pode ocasionar em uma confusão na identificação mesmo se ela for feita por um ser humano;
4. Favorecimento do caractere 7 – o alto índice de reconhecimento do caractere 7 também é devido a um artefato na imagem, que é uma linha tênue que ocorre frequentemente acima da região numérica. Tal fato gera uma tendência a identificar o caractere sete mais facilmente ou leva a classificar um caractere “1” como “7” indevidamente.

A tabela 5.5 mostra o número de imagens utilizadas por OP na etapa de reconhecimento e em que grupo as imagens ocorrem.

Tabela 5.5 – Número de imagens úteis por OP

OP	GRUPO 1 (número de imagens)	GRUPO 2 (número de imagens)
0679498	4	0
0682246	7	0
0684271	14	0
0741269	0	17
0741281	0	4
0741626	0	20

A tabela 5.6 mostra o índice de acertos no reconhecimento de cada caracter.

Tabela 5.6 – Acertos da metodologia para cada grupo de imagens

CARACTER	% de acertos no GRUPO 1	% de acertos no GRUPO 2
0	16,00	0,00
1	0,00	0,00
2	57,14	41,46
4	12,00	36,59
6	28,00	45,95
7	77,78	78,05
8	12,00	25,00
9	25,00	5,88

5.6 – Conclusões

No presente capítulo é apresentado um conjunto de resultados para cada etapa do processamento proposta. As estratégias de identificação do tarugo na cena e a parte de identificação dos padrões (caracteres) não obtiveram desempenho satisfatório, indicando uma revisão das mesmas visando uma melhoria de seu resultado e de sua robustez.

6 – CONCLUSÕES

É interessante analisar que a divisão do trabalho de processamento de imagem em etapas distintas e independentes pode facilitar a análise do problema. Entretanto, conforme é aumentado o número de etapas faz com que a confiança global caia exponencialmente conforme descrito a seguir.

A probabilidade de acerto de todas as etapas é condicionada ao acerto das etapas anteriores, o que pode levar a um nível de acerto global muito baixo se existir várias etapas de processamento (figura 6.1). Tentar agrupar etapas pode também não ser a solução. A escolha do número de etapas deve ser analisada com cuidado. O número de etapas também está diretamente ligado ao tempo de processamento, que para o sistema ideal proposto tem que girar em torno de 5 segundos para o processamento de todas as etapas. Metodologias como filtragem de imagens, correlação possuem ordem de complexidade elevada e se utilizadas excessivamente levarão a um acréscimo drástico no tempo de processamento. Tentativas de apenas melhorar o processamento da máquina onde é feita a análise tem caráter paliativo e pode não resolver o problema, sendo que o mais interessante é analisar a necessidade e coerência de cada ferramenta/etapa antes de adotá-la.

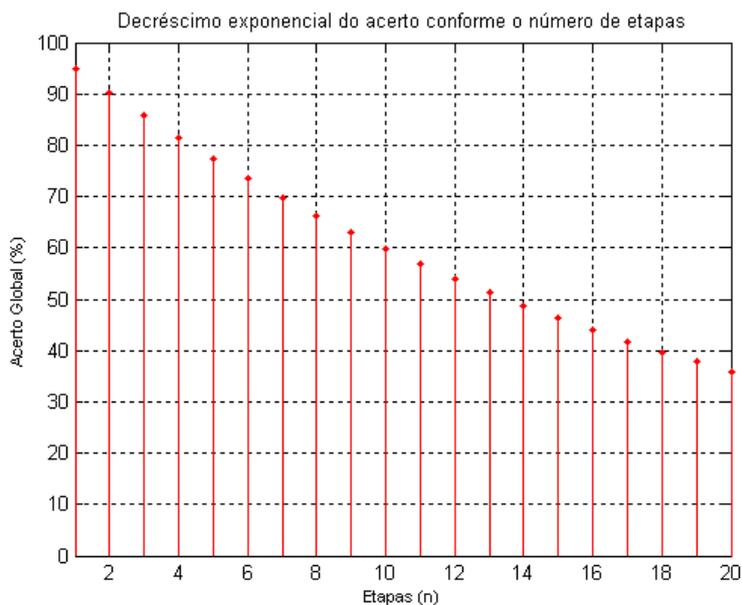


Figura 6.1 – Resultado de etapas consecutivas com confiança 0,95. Gráfico representa $0,95^n$, onde n é o número de etapas

Uma alternativa interessante e aparentemente viável de mudança no processo é a de, na pintura de tarugos provenientes de ordens de produções específicas, ser coberta apenas a parte rugosa. Tal estratégia torna a metodologia inicial de segmentação da face do tarugo válida novamente, supondo que agora o objeto de busca é um triângulo.

Uma alternativa para um melhor ajuste das metodologias de diagnóstico nebuloso é mudar o modelo Mamdani [20] para uma estrutura que suportasse treinamento. Dessa maneira fica bem mais simplificado o treinamento e possibilita erros menores no processo.

Um ponto crucial, talvez o mais importante, para a viabilidade do sistema de reconhecimento de ordens de produção, é a qualidade da estampagem numérica. Os caracteres numéricos presentes na estampa não devem, **obrigatoriamente**, possuir dependência entre eles, ou seja, não devem ser semelhantes morfologicamente entre si (correlacionados). Caracteres altamente correlacionados podem ser confundidos gerando incerteza na identificação.

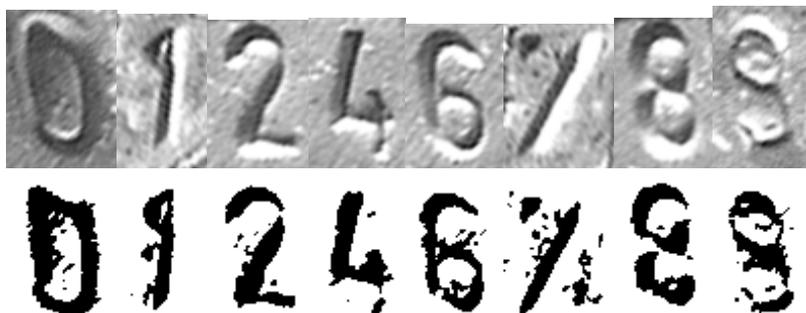


Figura 6.2 – Padrões coletados na LIT (imagens com zoom manual da câmera) e sua imagem resultante da binarização local proposta por Niblack [11] com parâmetros ($N = 31$ e $\text{weight} = -0.8$)

Mudanças no processo podem não ser possíveis em alguns casos, mas durante o decorrer da elaboração deste trabalho estava sendo feita uma mudança na estampagem. Opiniões e discussões feitas sobre a estampagem auxiliarão a definir um rumo nessa mudança.

A falta da presença dos padrões “3”, “5” foi consequência do não aparecimento desses caracteres nas OP’s dos dias correspondentes à coleta de imagens “online”.

Dentre as imagens coletada durante o presente trabalho foram coletados padrões para a composição do banco de amostras que servem como referência no processo de identificação. A partir da Tabela 6.1 pode-se concluir que muitos deles estão altamente correlacionados entre si e para que o sistema gere resultados satisfatórios é necessário eliminar essa dependência ou reduzi-la a valores aceitáveis de maneira que não comprometam a identificação. Outra observação prática realizada foi a de que o dígito “1” escolhido pela estampagem atual tem o formato muito simples, ele possui correlação alta com qualquer traço ou entalhe vertical que aparece na imagem.

Tabela 6.1 – Correlação entre padrões reais coletados

DÍGITO	0	1	2	4	6	7	8	9
0	1,00	0,41	0,30	0,38	0,51	0,38	0,47	0,43
1	0,41	1,00	0,40	0,41	0,34	0,35	0,35	0,36
2	0,30	0,41	1,00	0,41	0,46	0,46	0,58	0,43
4	0,41	0,45	0,49	1,00	0,42	0,38	0,44	0,45
6	0,51	0,32	0,46	0,44	1,00	0,31	0,71	0,56
7	0,38	0,34	0,49	0,37	0,31	1,00	0,33	0,27
8	0,49	0,33	0,58	0,42	0,71	0,32	1,00	0,54
9	0,43	0,34	0,43	0,45	0,56	0,27	0,54	1,00

O presente trabalho não cumpriu completamente seus objetivos, que estavam relacionados à implantação “online” do sistema, entretanto, apontou críticas e discussões válidas visando melhoramentos tanto do processo quanto das estratégias adotadas para o reconhecimento de padrões. A discussão é, portanto, válida indicando também perspectivas para desenvolvimentos de temas relacionados com o que muitas vezes é denominado de visão computacional.

Perspectivas Futuras

O processo de iluminação da face do tarugo deve ser encarado como crítico para o reconhecimento e o melhor condicionamento deverá permitir resultados mais satisfatórios do sistema. Uma abordagem utilizada atualmente em sistemas de inspeção que valoriza a evidência maior de detalhes se baseia em iluminação ultravioleta que pode ser facilmente inserida apenas trocando-se a lâmpada fluorescente por uma chamada lâmpada de “luz negra”. Tal iluminação tem como objetivo enfatizar detalhes e melhorar o contraste da imagem no vídeo.

Foi verificado que no presente sistema a sensação de nitidez e número de tonalidades da imagem no monitor analógico é melhor do que na imagem já digitalizada, entretanto a análise automática da imagem é feita em um computador após um processo de digitalização que reduz bastante a qualidade da imagem dependendo do conversor analógico para digital (*placa de captura de vídeo*). Além do processo de digitalização implicar em perda de informação devido a essa amostragem não ideal tem-se que um componente de ruído pode estar inserido no cabo coaxial que leva no sinal de vídeo para o ponto de análise. A figura 6.3 demonstra uma componente de ruído intensa de fonte supostamente vibratória ou eletromagnética ainda não identificada, que inviabiliza o processo de identificação. Uma solução possível para tal problema é a substituição da câmera por uma totalmente digital, ou seja, um dispositivo que gerasse o processo de digitalização localmente e enviasse essa informação para o ponto de análise. Câmeras de vídeos digitais também são livres do recurso do efeito de entrelaçamento de linhas na imagem e podem possuir resolução (amostragem), maior que a maioria das placas de captura de vídeo analógico disponíveis no mercado. Em processos de inspeção onde a análise é feita pelo operador é mais interessante o uso de câmera e monitor analógicos que enfatizam melhor os detalhes. Entretanto, para

processos onde a digitalização é obrigatória, o uso de câmeras de vídeo digital pode fornecer resultados melhores.

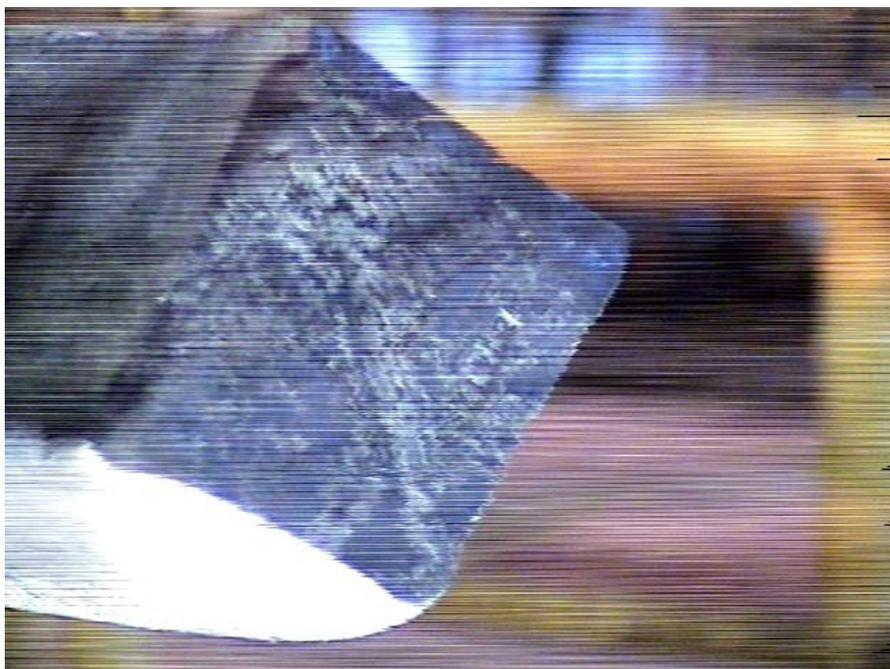


Figura 6.3 – Imagem coletada com a presença de ruído intenso

A figura 6.4 mostra um problema a ser solucionado que é a oclusão ocasional de um tarugo por outro. A figura 6.5 mostra um exemplo de estampagem em local indevido na região central da face do tarugo.

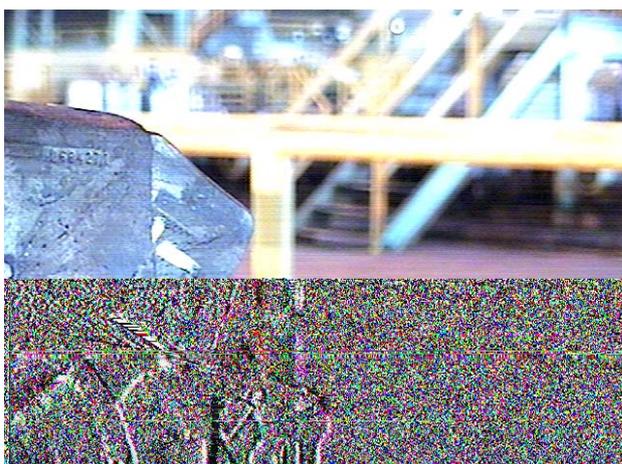


Figura 6.4 – Tarugo posicionado atrás de outro, impedindo a sua visualização

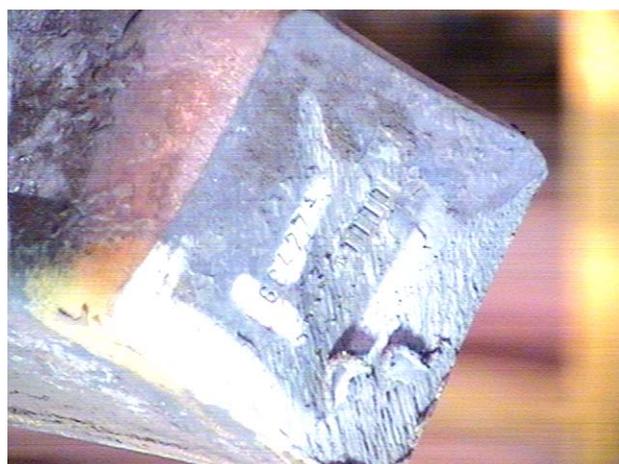


Figura 6.5 – Ocorrência de estampagem em local indevido

Modificações no processo e na coleta de imagens devem ser feitas. Com relação ao ponto de coleta de imagens ele deve ser modificado para uma localização onde se garanta uma distância entre a câmera e a face do tarugo, isso minimizará problemas de reconhecimento decorrentes de variações de escala. Um maior controle de qualidade na

estampagem deve ser feito para evitar impressões em locais indevidos onde o sistema não espera encontrar dígitos a serem reconhecidos.

Existe ainda uma possibilidade de comando do posicionamento da câmera de vídeo e do seu conjunto de lentes por meio do protocolo serial PELCO [4]. Tal alternativa permite mudar o zoom e posicionamento da câmera remotamente. Essa alternativa foi temporariamente abandonada, pois necessita de diversos testes, sendo que o protocolo não é liberado oficialmente pela PELCO. Supondo que o comando de zoom e posicionamento esteja ajustado corretamente utilizando o computador, esse processo adiciona mais tempo à etapa de processamento, que é da ordem de aproximadamente 1 segundo (dinâmica de movimentação dos equipamentos).

A imagem da figuras 6.6 representa o resultado de uma detecção de bordas usando SOBEL. A transformada de Hough é aplicada no resultado da detecção de bordas e os resultados podem ser vistos na figura 6.7, a transformada de Hough pode fornecer informações sobre segmentos de retas, presentes nas imagens. Isso pode ser usado na parte inicial de correção da rotação do tarugo, primeiramente identificando as retas que supostamente sejam candidatas a componentes da face e após isso deverá ser feita a devida correção da rotação na imagem digitalmente utilizando interpolação compensando o ângulo da reta. Também pode ser utilizado como mecanismo redundante na etapa de segmentação da face do tarugo. Uma desvantagem desse método é encontrar uma combinação viável de seus parâmetros de configuração para atender à maior parte das possibilidades.

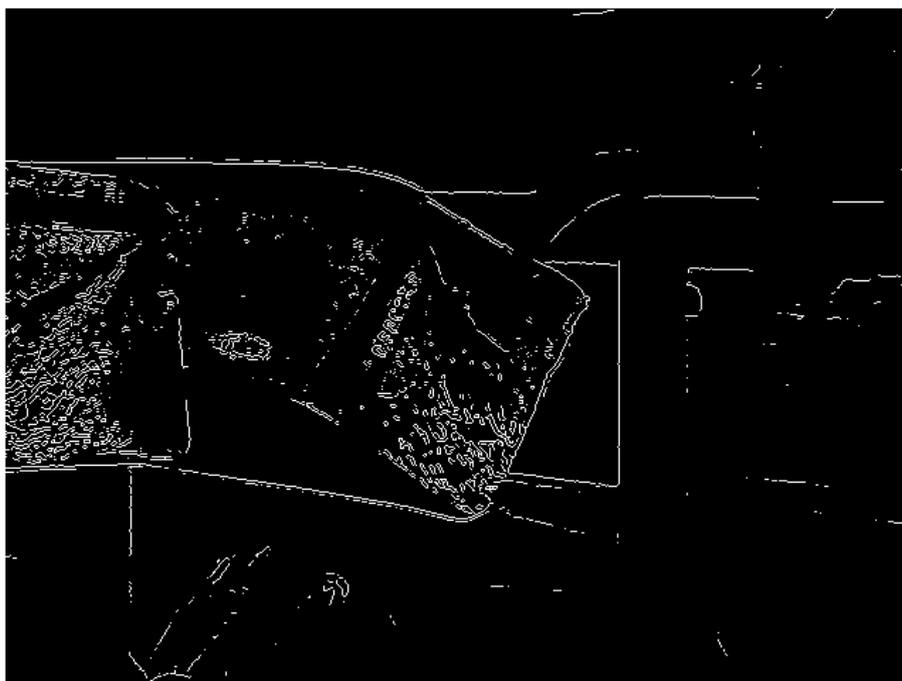


Figura 6.6 – Detecção de bordas com SOBEL e limiarização automática

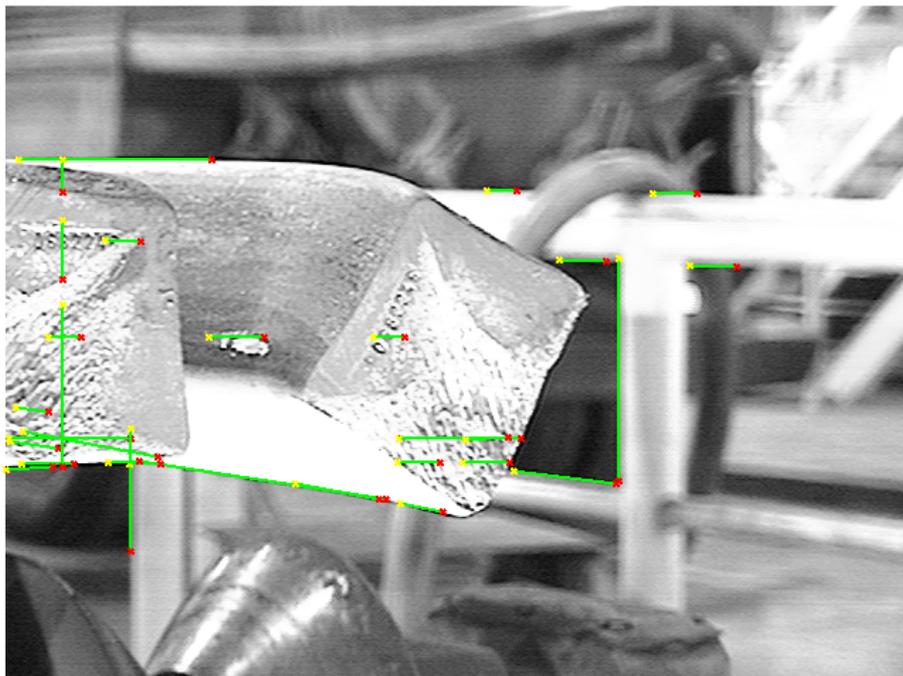


Figura 6.7 – Linhas detectadas por Hough (tamanho mínimo das linhas escolhidas = 20 pixels)

Por se tratar de um problema complexo e cheio de etapas ainda restam tarefas e ajustes personalizados a serem feitos para tornar o sistema completamente “online”. Dentre elas pode se destacar:

- 1 – Comunicação com o sistema de automação – tanto para a coleta de informação de sensores como para parametrização (configuração dos equipamentos em função das dimensões da OP) da LIT;
- 2 – Estimações dos momentos mais propícios para gerar alarmes em função da confiança do reconhecimento – deverá ser permitido ao operador auxiliar o sistema em caso de dúvida;
- 3 – Buscar lista diária de OP’s do sistema;
- 4 – Coleta de padrões – deverá ser feita até que o banco de padrões forneça um reconhecimento satisfatório, pois incrementa a redundância da identificação.

7 – REFERÊNCIAS

- [1]. RAMALHO; M.C. **Ambiente computacional para identificação da ordem de produção estampadas em tarugos de aço da AÇOMINAS**. Tese de mestrado, Universidade Federal de Minas Gerais, Julho de 2005.
- [2]. OLIVEIRA; D. R.. **Especificações para identificações da OP nos tarugos**. Relatório Técnico da AÇOMINAS, 2003.
- [3]. **SX11 Software Development Kit**. [Online]. Disponível: <http://www.sensoray.com/downloads/X11%20Software%20Manual%20v_2_1.PDF>
- [4]. **Pelco P and D protocol implementation in C#**. [Online]. Disponível: <<http://www.codeproject.com/useritems/PelcoPDinC.asp>>
- [5]. **CxImage (free C++ image processing and conversion library)**. [Online]. Disponível: <<http://www.xdp.it/cximage/>>
- [6]. **FFTW (Fastest Fourier Transform in the West)**. [Online]. Disponível: <<http://www.fftw.org/>>
- [7]. **Open Source Computer Vision Library**. [Online]. Disponível: <<http://www.intel.com/technology/computing/opencv/>>
- [8]. **DirectX SDK (Software Development Kit)**. [Online]. Disponível: <<http://msdn.microsoft.com/directx/sdk/>>
- [9]. **Object Management Group**. [Online]. Disponível: <<http://www.uml.org/>>
- [10]. OTSU; N. **A threshold selection method from gray-level histograms**. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62-66, 1979
- [11]. NIBLACK; W. **An introduction to Digital Image Processing**. Prentice-Hall, 1986
- [12]. SOBEL; I. **Pattern Classification and Scene Analysis**. Chapter 7 - Representation and Initial Simplifications, page 271. John Wiley & Sons, 1973.
- [13]. HOUGH; P.V.C. **Methods and means for recognizing complex patterns**. U.S. Patent 3069654, December 1962.

- [14]. RUSS; J.C. **The Image Processing Handbook**. CRC Press LLC, third edition, 1999.
- [15]. **Técnicas de Segmentação de Imagens para Recuperação de Informações Visuais**. Disponível: <<http://atlas.ucpel.tche.br/~vbastos/index.htm>>
- [16]. DUAN; T.D., DU; T.L.H, PHUOC; T.V., HOANG; N.V. **Building an Automatic Vehicle License-Plate Recognition System**. Intl. Conf. in Computer Science. February 21-24, 2005, Can Tho, Vietnam.
- [17]. LEWIS; J. P. **Fast Normalized Cross-Correlation**. Industrial Light & Magic. [Online]. Disponível: <<http://www.idiom.com/~zilla/Papers/nvisionInterface/nip.html>>
- [18]. COOLEY, J.W. AND TUKEY, J.W. **An algorithm for the machine calculation of complex Fourier series**. Mathematics of Computation. vol. 19, n.o 90, pp. 297-301, 1965.
- [19]. BRIGHAM; E. OREN. **The Fast Fourier Transform and Its Applications**. Englewood Cliffs, NJ: Prentice-Hall, Inc., 448 pp, 1988.
- [20]. JANG, J.S.R.; Sun C., MIZUTANI, E.. *Neuro-Fuzzy and Soft Computing*. Prentice Hall, 1997.
- [21]. **How Stuff Works**. [Online]. Disponível: <<http://www.howstuffworks.com/>>
- [22]. **Gerdau S. A. Gerdau Açominas – Produtos**. [Online]. Disponível:<<http://www.acominas.com.br/br/produtos/tarugos.asp>>
- [23]. GONZALES; R. C., WOODS; R. E. **Digital Image Processing**. Addison-Wesley. Publishing Company, second edition, 1992.
- [24]. BUTZERS, P.L.; JANSCHKE, S. **Mellin-Fourier series and the classical Mellin transform**. Computers & Mathematics with Applications, Volume 40, Issue 1, July 2000, Pages 49-62.
- [25]. FLAJOLET, P.; GOURDON, X.; DUMAS, P. **Mellin transforms and asymptotics: Harmonic sums**. Theoretical Computer Science, Volume 144, Issues 1-2, 26 June 1995, Pages 3-58.
- [26]. DERRODE, S.; GHORBEL, F. **Shape analysis and symmetry detection in gray-level objects using the analytical Fourier–Mellin representation**. Signal Processing, Volume 84, Issue 1, January 2004, Pages 25-39.
- [27]. SAZBON, D.; ZALEVSKY, Z.; RIVLIN, E.; MENDLOVIC, D. **Using Fourier/Mellin-based correlators and their fractional versions in navigational tasks**. Pattern Recognition, Volume 35, Issue 12, December 2002, Pages 2993-2999.
- [28]. SHARMA, K.K.; JOSHI, S.D. **On scaling properties of fractional Fourier transform and its relation with other transforms**. Optics Communications, Volume 257, Issue 1, 1 January 2006, Pages 27-38

- [29]. KAN, C.; SRINATH, M.D. **Invariant character recognition with Zernike and orthogonal Fourier–Mellin moments**. *Pattern Recognition*, Volume 35, Issue 1, January 2002, Pages 143-154.
- [30]. RUANAIDH, J.J.K.Ò.; PUN, T. **Rotation, scale and translation invariant spread spectrum digital image watermarking**. *Signal Processing*, Volume 66, Issue 3, 28 May 1998, Pages 303-317.
- [31]. JIN, A.T.B.; LING, D.N.C.; SONG, O.T. **An efficient fingerprint verification system using integrated wavelet and Fourier–Mellin invariant transform**. *Image and Vision Computing*, Volume 22, Issue 6, 1 June 2004, Pages 503-513.
- [32]. LI, Y.; WONG, W.C.K. **An efficient and robust method to detect object centre**. *Pattern Recognition*, Volume 30, Issue 5, May 1997, Pages 659-671.
- [33]. PARK, H.J.; YANG, H.S. **Invariant object detection based on evidence accumulation and Gabor features**. *Pattern Recognition Letters*, Volume 22, Issue 8, June 2001, Pages 869-882
- [34]. KIM, E.Y.; PARK, S.H. **Automatic video segmentation using genetic algorithms**. *Pattern Recognition Letters*, Volume 27, Issue 11, August 2006, Pages 1252-1265.
- [35]. ZHANG, H.; GAO, W.; CHEN, X.; ZHAO, D. **Object detection using spatial histogram features**. *Image and Vision Computing*, Volume 24, Issue 4, 1 April 2006, Pages 327-341.
- [36]. HOWARD, D.; ROBERTS, S.C.; RYAN, C. **Pragmatic Genetic Programming strategy for the problem of vehicle detection in airborne reconnaissance**. *Pattern Recognition Letters*, Volume 27, Issue 11, August 2006, Pages 1275-1288.
- [37]. KASTRINAKI, V.; ZERVAKIS, M.; KALAITZAKIS, K. **A survey of video processing techniques for traffic applications**. *Image and Vision Computing*, Volume 21, Issue 4, 1 April 2003, Pages 359-381.
- [38]. ZHANG, W.; FANG, X.Z.; YANG, X. **Moving vehicles segmentation based on Bayesian framework for Gaussian motion model**. *Pattern Recognition Letters*, Volume 27, Issue 9, 1 July 2006, Pages 956-967.
- [39]. BERGBOER, N.H.; POSTMA, E.O.; HERIK, H.J.V.D. **Context-based object detection in still images**. *Image and Vision Computing*, In Press, Corrected Proof, Available online 17 April 2006.
- [40]. JI, X.; WEI, Z.; FENG, Y. **Effective vehicle detection technique for traffic surveillance systems**. *Journal of Visual Communication and Image Representation*, Volume 17, Issue 3, June 2006, Pages 647-658.
- [41]. WU, Y.; SHEN, J.; DAI, M. **Traffic object detections and its action analysis**. *Pattern Recognition Letters*, Volume 26, Issue 13, 1 October 2005, Pages 1963-1984.

- [42]. ZHOU, Q.; MA, L.; CHELBERG, D. **Adaptive object detection and recognition based on a feedback strategy**. Image and Vision Computing, Volume 24, Issue 1, 1 January 2006, Pages 80-93.
- [43]. DESCLÉE, B.; BOGAERT, P.; DEFOURNY, P. **Forest change detection by statistical object-based method**. Remote Sensing of Environment, Volume 102, Issues 1-2, 30 May 2006, Pages 1-11.
- [44]. MILLER, O.; PIKAZ, A.; AVERBUCH, A. **Objects based change detection in a pair of gray-level images**. Pattern Recognition, Volume 38, Issue 11, November 2005, Pages 1976-1992.
- [45]. HALL, O.; HAY, G.J. **A Multiscale Object-Specific Approach to Digital Change Detection**. International Journal of Applied Earth Observation and Geoinformation, Volume 4, Issue 4, November 2003, Pages 311-327.
- [46]. PARK, H.J.; YANG, H.S.. **Invariant object detection based on evidence accumulation and Gabor features**. Pattern Recognition Letters, Volume 22, Issue 8, June 2001, Pages 869-882.
- [47]. AMER, A.; DUBOIS, E.; MITICHE, A. **Rule-based real-time detection of context-independent events in video shots**. Real-Time Imaging, Volume 11, Issue 3, June 2005, Pages 244-256.
- [48]. CHO, J.; KIM, S. **Object detection using multi-resolution mosaic in image sequences**. Signal Processing: Image Communication, Volume 20, Issue 3, March 2005, Pages 233-253.
- [49]. MINDRU, F.; TUYTELAARS, T.; GOOL, L.V.; MOONS, T. **Moment invariants for recognition under changing viewpoint and illumination**. Computer Vision and Image Understanding, Volume 94, Issues 1-3, April-June 2004, Pages 3-27.
- [50]. WALTHER, D.; RUTISHAUSER, U.; KOCH, C.; PERONA, P.. **Selective visual attention enables learning and recognition of multiple objects in cluttered scenes**. Computer Vision and Image Understanding, Volume 100, Issues 1-2, October-November 2005, Pages 41-63.
- [51]. BERWICK, D.; LEE, S.W. **Spectral gradients for color-based object recognition and indexing**. Computer Vision and Image Understanding, Volume 94, Issues 1-3, April-June 2004, Pages 28-43.
- [52]. BISCHOF, H.; WILDENAUER, H.; LEONARDIS, A. **Illumination insensitive recognition using eigenspaces**. Computer Vision and Image Understanding, Volume 95, Issue 1, July 2004, Pages 86-104.
- [53]. ULLMAN, S.; BART, E. **Recognition invariance obtained by extended and invariant features**. Neural Networks, Volume 17, Issues 5-6, June-July 2004, Pages 833-848.

- [54]. SONG, B.S.; LEE, K.M.; LEE, S.U. **Model-Based Object Recognition Using Geometric Invariants of Points and Lines**. Computer Vision and Image Understanding, Volume 84, Issue 3, December 2001, Pages 361-383.
- [55]. BHANU, B.; LIN, Y. **Stochastic models for recognition of occluded targets**. Pattern Recognition, Volume 36, Issue 12, December 2003, Pages 2855-2873.
- [56]. LEE, Y.; PARK, R. **A surface-based approach to 3-D object recognition using a mean field annealing neural network**. Pattern Recognition, Volume 35, Issue 2, February 2002, Pages 299-316.
- [57]. PERNER, P.; JÄNICHEN, S.; PERNER, H.. **Case-based object recognition for airborne fungi recognition**. Artificial Intelligence in Medicine, Volume 36, Issue 2, February 2006, Pages 137-157.
- [58]. BERWICK, D.; LEE, S.W. **Spectral gradients for color-based object recognition and indexing**. Computer Vision and Image Understanding, Volume 94, Issues 1-3, April-June 2004, Pages 28-43.
- [59]. ZHUGE, Y.; UDUPA, J. K.; SAHA, P. K. **Vectorial scale-based fuzzy-connected image segmentation**. Computer Vision and Image Understanding, Volume 101, Issue 3, March 2006, Pages 177-193.
- [60]. PARAGIOS, N.; DERICHE, R. **Geodesic active regions and level set methods for motion estimation and tracking**. Computer Vision and Image Understanding, Volume 97, Issue 3, March 2005, Pages 259-282.
- [61]. NIKOLAEV, D.P.; NIKOLAYEV, P.P. **Linear color segmentation and its implementation**. Computer Vision and Image Understanding, Volume 94, Issues 1-3, April-June 2004, Pages 115-139.
- [62]. PHAM, T.V.; SMEULDERS, A.W.M. **Object recognition with uncertain geometry and uncertain part detection**. Computer Vision and Image Understanding, Volume 99, Issue 2, August 2005, Pages 241-258.
- [63]. BERGEVIN, R.; BUBEL, A. **Object-level structured contour map extraction**. Computer Vision and Image Understanding, Volume 91, Issue 3, September 2003, Pages 302-334.
- [64]. CAMPBELL, R.J.; FLYNN, P.J. **A Survey Of Free-Form Object Representation and Recognition Techniques**. Computer Vision and Image Understanding, Volume 81, Issue 2, February 2001, Pages 166-210.
- [65]. MAYER, H. **Automatic Object Extraction from Aerial Imagery—A Survey Focusing on Buildings**. Computer Vision and Image Understanding, Volume 74, Issue 2, 1 May 1999, Pages 138-149.
- [66]. CHAU, C.; SIU, W. **Adaptive dual-point Hough transform for object recognition**. Computer Vision and Image Understanding, Volume 96, Issue 1, October 2004, Pages 1-16.

- [67]. LU, J.; YUAN, X.; YAHAGI, T. **A method of face recognition based on fuzzy clustering and parallel neural networks**. *Signal Processing*, Volume 86, Issue 8, August 2006, Pages 2026-2039.
- [68]. YANG, J.; YANG, M. **A control chart pattern recognition system using a statistical correlation coefficient method**. *Computers & Industrial Engineering*, Volume 48, Issue 2, March 2005, Pages 205-221.
- [69]. LIU, C.; SAKO, H. **Class-specific feature polynomial classifier for pattern classification and its application to handwritten numeral recognition**. *Pattern Recognition*, Volume 39, Issue 4, April 2006, Pages 669-681.
- [70]. WOOD, J. **Invariant pattern recognition: A review**. *Pattern Recognition*, Volume 29, Issue 1, January 1996, Pages 1-17.
- [71]. ULLMAN, S.; SOLOVIEV, S. **Computation of pattern invariance in brain-like structures**. *Neural Networks*, Volume 12, Issues 7-8, 11 October 1999, Pages 1021-1036.
- [72]. SUK, T.; FLUSSER, J. **Combined blur and affine moment invariants and their use in pattern recognition**. *Pattern Recognition*, Volume 36, Issue 12, December 2003, Pages 2895-2907.
- [73]. WANG, R. **A hybrid learning network for shift-invariant recognition**. *Neural Networks*, Volume 14, Issue 8, October 2001, Pages 1061-1073.
- [74]. PEI, S.; LIN, C. **Image normalization for pattern recognition**. *Image and Vision Computing*, Volume 13, Issue 10, December 1995, Pages 711-723.
- [75]. WANG, S.; LIN, W. **A new self-organizing neural model for invariant pattern recognition**. *Pattern Recognition*, Volume 29, Issue 4, April 1996, Pages 677-687.
- [76]. CHEN, G.Y.; BUI, T.D.; KRZYŻAK, A. **Rotation invariant pattern recognition using ridgelets, wavelet cycle-spinning and Fourier features**. *Pattern Recognition*, Volume 38, Issue 12, December 2005, Pages 2314-2322.
- [77]. WONG, W.; SIU, W.; LAM, K. **Generation of moment invariants and their uses for character recognition**. *Pattern Recognition Letters*, Volume 16, Issue 2, February 1995, Pages 115-123.
- [78]. WOOD, J.; TAYLOR, J.S. **A unifying framework for invariant pattern recognition**. *Pattern Recognition Letters*, Volume 17, Issue 14, 30 December 1996, Pages 1415-1422.
- [79]. HEIKKILÄ, J. **Pattern matching with affine moment descriptors**. *Pattern Recognition*, Volume 37, Issue 9, September 2004, Pages 1825-1834.

- [80]. KHABOU, M.A.; HERMI, L.; RHOUMA, M.B.H. **Shape recognition using eigenvalues of the Dirichlet Laplacian**. Pattern Recognition, In Press, Corrected Proof, Available online 3 March 2006.
- [81]. ROHRMUS, D.R. **Invariant and adaptive geometrical texture features for defect detection and classification**. Pattern Recognition, Volume 38, Issue 10, October 2005, Pages 1546-1559.
- [82]. LIU, D., LAM, K.; SHEN, L.. **Illumination invariant face recognition**. Pattern Recognition, Volume 38, Issue 10, October 2005, Pages 1705-1716.
- [83]. YANG, L.; SUEN, C.Y.; BUI, T.D.; ZHANG, P. **Discrimination of similar handwritten numerals based on invariant curvature features**. Pattern Recognition, Volume 38, Issue 7, July 2005, Pages 947-963.
- [84]. FINLAYSON, G.; HORDLEY, S.; SCHAEFER, G.; TIAN, G.Y.. **Illuminant and device invariant colour using histogram equalization**. Pattern Recognition, Volume 38, Issue 2, February 2005, Pages 179-190.
- [85]. JIN, L.; TIANXU, Z. **Fast algorithm for generation of moment invariants**. Pattern Recognition, Volume 37, Issue 8, August 2004, Pages 1745-1756.