

**JOSÉ MARCELO GOMES**

**DESENVOLVIMENTO E AVALIAÇÃO  
DE UM PROTÓTIPO CLASSIFICADOR DE TÁBUAS  
USANDO TÉCNICAS DE VISÃO ARTIFICIAL**

**Tese apresentada à Universidade  
Federal de Viçosa, como parte das  
exigências do Programa de Pós-  
Graduação em Engenharia Agrícola,  
para obtenção do título de *Doctor  
Scientiae*.**

VIÇOSA  
MINAS GERAIS – BRASIL  
2007

**JOSÉ MARCELO GOMES**

**DESENVOLVIMENTO E AVALIAÇÃO  
DE UM PROTÓTIPO CLASSIFICADOR DE TÁBUAS  
USANDO TÉCNICAS DE VISÃO ARTIFICIAL**

**Tese apresentada à Universidade  
Federal de Viçosa, como parte das  
exigências do Programa de Pós-  
Graduação em Engenharia Agrícola,  
para obtenção do título de *Doctor  
Scientiae*.**

APROVADA: 26 de janeiro de 2007.

---

Prof. Ricardo Marius Della Lucia  
(Co-Orientador)

---

Prof. Haroldo Carlos Fernandes

---

Prof. Nerilson Terra Santos

---

Prof. Joseph Kalil Khoury Junior

---

Prof. Francisco de Assis de Carvalho Pinto  
(Orientador)

À meu filho Lucas, que é minha razão de viver.  
À minha esposa Silvane, pelo permanente apoio.  
À minha mãe Áurea, cujos ensinamentos guiam a minha caminhada.

*“... sem caridade e sem amor, eu nada seria...”*

## **AGRADECIMENTOS**

A Deus, por todas as oportunidades.

À Universidade Federal de Viçosa, ao Departamento de Engenharia Agrícola e à Escola Agrotécnica Federal de Catu-BA, pela oportunidade de realização do curso.

Ao Professor Francisco de Assis de Carvalho Pinto pela orientação.

Ao Professor Daniel Marçal de Queiroz e Professor Ricardo Marius Della Lucia pela có-orientação e estímulo.

Aos Professores Haroldo Carlos Fernandes, Joseph Kalil Khoury Junior e Nerilson Terra Santos pela ajuda e participação na banca examinadora.

À todos os colegas e amigos do PROVISAGRO pelo auxílio, amizade e bom convívio.

Sinceros agradecimentos a todos que colaboraram para que este trabalho fosse realizado.

## BIOGRAFIA

JOSÉ MARCELO GOMES, filho de Herades Rodrigues Gomes (*in memorian*) e Áurea Dutra Gomes, nasceu em 26 de dezembro de 1969, em Aimorés, Estado de Minas Gerais.

Em 1989, iniciou o curso de Informática na Universidade Federal de Viçosa, graduando-se em dezembro de 1993.

Trabalhou no serviço público federal como Analista de Sistemas, durante o período de 1994 a 1997.

Em 1994, iniciou o Curso de Mestrado em Ciência da Computação, na área de concentração em Informática na Educação, na Universidade Federal de Minas Gerais, defendendo tese em abril de 1997.

Trabalha no serviço público federal como Professor de Ensino de 1<sup>o</sup> e 2<sup>o</sup> Graus, desde 1997.

Em agosto de 2003, iniciou o Curso de Doutorado em Engenharia Agrícola, área de concentração em Mecanização Agrícola no Departamento de Engenharia Agrícola da Universidade Federal de Viçosa, defendendo tese em janeiro de 2007.

## SUMÁRIO

RESUMO.....	viii
ABSTRACT.....	x
1 INTRODUÇÃO.....	1
1.1 Objetivo geral.....	4
1.2 Objetivos específicos.....	5
2 REVISÃO DE LITERATURA.....	6
2.1 Classificação de madeira serrada e automação.....	8
2.2 Reconhecimento de padrões para o processo de classificação.....	12
2.3 Análise de Componentes Principais.....	19
2.4 Avaliação de um classificador.....	22
3 MATERIAL E MÉTODOS.....	25
3.1 Apresentação do protótipo.....	25
3.2 Interface de usuário.....	27
3.3 Desenvolvimento do classificador estatístico.....	28
3.4 Eliminação de características por Análise de Componentes Principais.....	32
3.5 Processamento das imagens.....	32
3.6 Reclassificação baseada em características morfológicas e posição.....	36
3.7 Características morfológicas e informações de posição.....	37
3.8 Avaliação do resultado da classificação de blocos e das tábuas.....	38
3.9 Análise das imagens e classificação das tábuas.....	39
3.10 Avaliação do desempenho do protótipo.....	43
4 RESULTADOS E DISCUSSÃO.....	45
4.1 Interface do usuário.....	45
4.2 Parâmetros do classificador estatístico.....	47
4.3 Eliminação variáveis originais por Análise Componentes Principais.....	48
4.4 Resultado da classificação de blocos de imagens.....	50
4.5 Classificação de tábuas.....	53
4.6 Resultados da classificação.....	54
4.7 O comportamento funcional do protótipo.....	59
5 CONCLUSÕES.....	61
6 SUGESTÕES PARA TRABALHOS FUTUROS.....	62
7 REFERÊNCIAS BIBLIOGRÁFICAS.....	63
8 APÊNDICES.....	70

## LISTA DE FIGURAS

<b>Figura 1</b> – Conceitos do coeficiente Kappa.....	24
<b>Figura 2</b> – Visão geral do protótipo de classificação de madeira serrada por um sistema de visão artificial.....	36
<b>Figura 3</b> – Exemplos de blocos de imagem 64x64 pixels dos agrupamentos.....	29
<b>Figura 4</b> – Imagens parciais obtidas de uma tábua (a), junção das imagens parciais para formação da imagem da tábua (b) e imagem da tábua após a retirada da esteira (c).....	34
<b>Figura 5</b> – Fluxograma de classificação de uma face da tábua (Norma comercial).....	41
<b>Figura 6</b> – Fluxograma de classificação de uma face da tábua (Norma ABNT).....	42
<b>Figura 7</b> – Fluxograma de classificação da tábua: A, B = classificação das faces (1 a 6); nf=soma nós firmes; nc=soma nós cariados; ns=soma nós soltos.....	43
<b>Figura 8</b> – Caixa de diálogo de escolha do modo de operação.....	45
<b>Figura 9</b> – Caixa de diálogo para seleção da regra de classificação.....	45
<b>Figura 10</b> – Caixa de diálogo informando o estado de espera por tábua.....	46
<b>Figura 11</b> – Caixa de diálogo com o resultado da leitura parcial da tábua.....	46
<b>Figura 12</b> – Caixa de diálogo com o resultado da classificação final da tábua.....	47

## LISTA DE TABELAS E QUADROS

<b>Tabela 1</b> – Classificação supervisionada versus classificação não supervisionada: vantagens e desvantagens.....	17
<b>Quadro 1</b> – Matriz de confusão para blocos de imagens da amostra de teste.....	51
<b>Quadro 2</b> – Matriz de confusão para imagens das tábuas classificadas pela regra da ABNT, sem o uso da reclassificação dos blocos.....	55
<b>Quadro 3</b> – Matriz de confusão para imagens das tábuas classificadas pela regra da ABNT, com o uso da reclassificação dos blocos.....	56
<b>Quadro 4</b> – Matriz de confusão para imagens das tábuas classificadas pela regra comercial, sem o uso da reclassificação dos blocos.....	57
<b>Quadro 5</b> – Matriz de confusão para imagens das tábuas classificadas pela regra comercial, com o uso da reclassificação dos blocos.....	58

## RESUMO

GOMES, José Marcelo, D.Sc., Universidade Federal de Viçosa, janeiro de 2007. **Desenvolvimento e avaliação de um protótipo classificador de tábuas usando técnicas de visão artificial.** Orientador: Francisco de Assis de Carvalho Pinto. Co-orientadores: Daniel Marçal de Queiroz e Ricardo Marius Della Lucia.

Muito esforço tem sido despendido por pesquisadores no sentido de se aplicar automação aos processos produtivos. A indústria de produção de madeira tem um lugar de destaque nesta atividade, porque a madeira é matéria prima de produtos com demanda crescente, como na produção de móveis, que exige peças com alta qualidade. A classe de qualidade de uma peça de madeira serrada é determinada pelos defeitos que ela possui e por algumas características associadas a esses defeitos como: tamanho, posição, quantidade e tipo. O objetivo deste trabalho foi desenvolver e avaliar um protótipo para classificação de tábuas de madeira de eucalipto com base em imagens digitais. Foi desenvolvido e testado um classificador para reconhecimento de defeitos em tábuas de eucalipto levando-se em consideração as características de percentis dos valores de pixel do histograma, além de informações morfológicas e de posição dos defeitos. Foi implementado no algoritmo de classificação as etapas de processamento e análise de imagens para classificação de tábuas. O passo seguinte foi a construção e teste de um protótipo de classificação de madeira serrada que utiliza um sistema de visão artificial. Ele é composto de uma esteira rolante por onde são inseridas as tábuas para serem conduzidas para sob a câmera de leitura de suas faces e obtenção das imagens. O protótipo pode utilizar tanto a norma da Associação Brasileira de Normas Técnicas (ABNT) quanto a norma

comercial das serrarias para a classificação. O processo pode ser acompanhado na tela do microcomputador que apresenta em seguida a imagem da tábua com o resultado final da classificação da mesma. As normas sofreram simplificações para viabilizar sua aplicação. A taxa de acerto no processo de classificação foi de 64 e 81% usando, respectivamente, as normas da ABNT e comercial. A produtividade do protótipo desenvolvido foi de  $7,9 \text{ m}^3 \text{ h}^{-1}$  na classificação de madeira serrada de eucalipto.

## ABSTRACT

GOMES, José Marcelo, D.Sc., Universidade Federal de Viçosa, January of 2007. **Development and evaluation of a prototype for sawn wood classification using artificial vision technics.** Adviser: Francisco de Assis de Carvalho Pinto. Co-Advisers: Daniel Marçal de Queiroz and Ricardo Marius Della Lucia.

Much effort has been expended by researchers to apply automation to productive processes. The sawn wood industry has a promising place in this activity, because lumber is a raw material for products with increasing demand, such as furniture, which requires high quality pieces. The quality of the lumber is determined by the defects it presents and some characteristics correlated to these defects such as: size, position, amount and type. The goal of the present research was to develop and evaluate a prototype to classify eucalyptus lumber using digital images. A classifier was developed and tested to recognize defects in eucalyptus lumbers taking in consideration the histogram values of the percentiles of pixels, morphologic information and position of the defects. Image processing and analysis for lumber classification was implemented in the classification algorithm. The following step was the construction and testing of a prototype for the classification of lumber using a machine vision system. It is built in a conveyor belt where the lumbers are inserted and conducted for image acquisition. The prototype can use both, the Brazilian standard (ABNT) and a commercial rule for lumber classification. The process can be followed in the microcomputer screen that shows the lumber images with its final grade. The rules were simplified to make possible its application. The overall accuracy in the classification process was 64 and 81% percent using the ABNT and commercial rules, respectively. The productivity of the developed prototype was  $7.9 \text{ m}^3 \text{ h}^{-1}$  in eucalyptus lumber grading.

## 1 INTRODUÇÃO

Nas últimas décadas muito esforço tem sido despendido por pesquisadores no sentido de se aplicar automação aos processos produtivos. A indústria, por exemplo, tem se empenhado em usar a tecnologia da computação no sentido de aumentar a produtividade para redução de custos. Para ser competitivo, é preciso atualizar a linha de produção visando incrementar a qualidade do produto, melhorar as condições de trabalho, minimizar custos para aumentar lucros, ou até mesmo, manter uma margem aceitável nos lucros uma vez que a competição tem reduzido consideravelmente o preço de produtos e serviços em geral.

A indústria de produção de madeira tem um lugar de destaque no que diz respeito a automação do processo produtivo de madeira serrada, porque a madeira é matéria prima de produtos com demanda cada dia maior: casas inteiras construídas quase que sem o uso de alvenaria; uso intenso na construção civil na forma de suporte ao processo de edificação; construção de janelas, portas e pisos; e pela indústria de móveis em geral. Esta última aplicação do uso da madeira tem sido alvo de estudos para automação de seu processo produtivo, não só visando o aumento de produtividade, como também melhoria do produto final que é a madeira serrada (KLINKHACHORN et al., 1995; BHANDARKAR et al., 2002; CONNERS et al., 1989A; KLINE et al., 2003; OJA et al., 2003; LEE et al., 2003).

Uns dos fatores que afetam sensivelmente a produtividade nos ambiente de serraria são os graduadores humanos, pois esta é uma atividade estressante porque exige grande atenção por longos períodos de tempo. Os

graduadores são aqueles trabalhadores das serrarias que observam as faces das tábuas (madeira serrada) para definir sua classificação em termos de qualidade. A qualidade de uma peça de madeira serrada, ou seja, seu valor, é determinada por várias características dos defeitos que ela possui, como: tamanho, posição, quantidade e tipo (LEE et al., 2003). Quanto menos defeitos uma peça de madeira tiver maior seu valor (SARIGUL et al., 2003).

Segundo KLINE et al. (2003), os graduadores humanos possuem uma taxa de acerto na classificação em torno de 48%. Portanto, um sistema automatizado para realização desta tarefa tem uma margem grande para prover aumento na taxa de acerto, dado que o desempenho de classificadores humanos é relativamente baixo. Além disto, os equipamentos não são afetados por fadiga ou estresse, que podem causar diminuição de rendimento.

Uma opção que tem sido objeto de estudo para automação do processo de classificação de madeira serrada é a utilização de um sistema de visão artificial (LU, 1997; KAUPPINEN, 1999; KLINKHACHORN et al., 1995). Nestes sistemas, a cena é adquirida por uma câmera, processada e uma graduação, ou valor, é atribuído à peça de madeira. Segundo GONZALEZ e WOODS (2000), um sistema de visão artificial pode operar em uma velocidade maior que a de classificadores humanos, ininterruptamente, e prover qualidade superior.

De maneira geral, o problema de inspeção de superfícies pode ser dividido nos seguintes sub-problemas: detecção do defeito, classificação do mesmo e controle do processo (IIVARINEN e VISA, 2000). Dessa maneira, um sistema automático de classificação de peças de madeira serrada necessita de um sistema de visão artificial para localizar, identificar e quantificar defeitos na madeira serrada e classificar as peças com base nesta quantificação.

Os sistemas de visão artificial para inspeção de superfícies de peças de madeira são constituídos de três etapas: aquisição, processamento e análise de imagens. Na primeira etapa, utilizam-se os sensores ópticos para obtenção das cenas. A etapa de processamento da imagem tem como objetivo a

segmentação e reconhecimento (identificação e medição) dos objetos de interesse. Na última etapa, a informação é transformada de sua forma de imagem digital para expressões que designam a classificação da peça da madeira como, por exemplo, pertencente a uma determinada classe ou peça a ser descartada (KHOURY JUNIOR, 2004).

As normas utilizadas comercialmente nas serrarias para classificação de madeira serrada não levam em consideração tantos defeitos na hora da identificação da classe da peça como a Norma NBR 14.806 (ABNT, 2002B). Esta foi uma das dificuldades na obtenção de uma exatidão maior que 65% na classificação das faces de tábuas com base na NBR 14.806 no trabalho de KHOURY JUNIOR (2004), pois alguns defeitos são parecidos e portanto fáceis de serem confundidos como, por exemplo, os diferentes tipos de nós (soltos, cariadados, vazados), ou o fendilhado com a racha. Observou-se também, que alguns desses defeitos possuem o mesmo peso para a norma de classificação e que, portanto, podem ser considerados como “iguais”, sem que haja perda significativa no resultado do processo de classificação. Há trabalhos que levaram em consideração um menor número de defeitos para efeitos de classificação: KLINE et al. (2003) consideraram 10 tipos de defeitos no desenvolvimento de uma máquina com múltiplos sensores para classificação de madeira serrada, alcançando 63% de exatidão. KLINKHACHORN et al. (1995) implementaram a identificação de 7 tipos de defeitos em seu trabalho de aplicação de automação na indústria de produção de peças de madeira serrada.

Outros problemas são os defeitos de dimensão reduzida, como o furo de inseto, que também dificulta a sua identificação (KLINE et al., 2003; KHOURY JUNIOR, 2004). Geralmente, as normas comerciais não consideram o furo de inseto, pois a madeira proveniente de reflorestamento é menos suscetível a este problema do que a madeira nativa (ARACRUZ, 2004).

Na etapa de processamento das imagens existem várias características que podem ser extraídas das mesmas com o objetivo de realizar a identificação de defeitos e posterior classificação. Em trabalhos recentes

KAUPPINEN (1999) e KHOURY JUNIOR (2004) mostraram que os percentis dos valores dos pixels das imagens são características simples de calcular e permitem discriminações satisfatórias dos defeitos de madeira serrada e implementação em tempo real. Por outro lado, a inclusão de técnicas que levam em consideração outras características dos defeitos, como, por exemplo, seus aspectos morfológicos e características de posição onde mais ocorrem nas tábuas, pode melhorar a discriminação entre os mesmos e, portanto, aumentar a taxa de acerto no processo de classificação.

Segundo IIVARINEN e VISA (2000), a forma é uma importante característica de um objeto. O objetivo do reconhecimento de formas é encontrar descrições que contêm informação suficiente para melhor caracterizar os objetos de interesse bem como facilitar a distinção uns dos outros. Resultados satisfatórios foram obtidos por estes autores na classificação de defeitos de madeira utilizando-se os aspectos morfológicos dos mesmos.

O presente trabalho faz parte da linha de pesquisa iniciada pelo trabalho de KHOURY JUNIOR (2004) que tem como objetivo principal o desenvolvimento de um sistema para classificação automática de madeira serrada. Uma hipótese desta pesquisa é de que é possível aumentar a taxa de acerto na utilização de Sistemas de Visão Artificial para classificação de madeira serrada. Para isso, deve-se levar em consideração as características morfológicas e de posição dos defeitos encontrados junto com características espectrais (percentis dos valores de pixel do histograma) de imagens coloridas. Outra hipótese é de que levando em consideração normas menos complexas pode-se melhorar a taxa de acerto de classificação.

### ***1.1 Objetivo geral***

Este trabalho teve como objetivo geral desenvolver e avaliar um protótipo para classificação de tábuas de madeira de eucalipto com base em um sistema de visão artificial.

## **1.2 Objetivos específicos**

1. Construir um protótipo de classificação de madeira serrada utilizando um sistema de visão artificial;
2. Desenvolver uma interface de usuário para compor o sistema de visão artificial do protótipo;
3. Desenvolver e testar um classificador para reconhecimento de defeitos em tábuas de eucalipto levando-se em consideração percentis dos valores de pixel do histograma, as características morfológicas e localização de cada defeito;
4. Implementar algoritmos de classificação de tábuas de eucalipto com base em uma norma oficial e uma comercial;
5. Avaliar o desempenho do protótipo.

## 2 REVISÃO DE LITERATURA

A utilização de madeira vem sendo feita pela humanidade desde os seus primórdios.

Nas últimas décadas, a madeira foi sendo mais estudada e compreendida, o que foi dando a ela inúmeros outros usos. A madeira é hoje matéria prima para grande gama de produtos como: laminados, compensados, chapas de madeira aglomerada, celulose, papel, energia, madeira serrada, etc.

O consumo de madeira cresceu tentando acompanhar o crescimento da população mundial, causando um significativo impacto nos estoques de madeiras nativas das florestas, principalmente as tropicais. Estes recursos vêm se tornando escassos e incapazes de acompanhar proporcionalmente o aumento do consumo (ROCHA, 2000). No Brasil, no ano de 1998, em torno de 69% da madeira consumida era proveniente de florestas naturais, portanto, ficando evidente que alguma atitude precisava ser tomada para frear o desmatamento (MACIEL, 1998).

Uma das utilizações da madeira é na forma serrada. Assim requer um grande volume com características específicas necessárias para tal utilização. O interesse pela utilização de espécies de rápido crescimento, como fonte de matéria prima para a obtenção de produtos sólidos da madeira, cresce constantemente devido as restrições impostas ao uso de madeiras oriundas de florestas tropicais nativas, além das dificuldades da exploração da floresta tropical e às grandes distâncias destas às regiões de consumo (PONCE, 1995; SILVA, 2000). Dados publicados pela FAO (Organização das Nações

Unidas para a Agricultura e a Alimentação) indicam para o Brasil um crescimento médio anual por volta de 6% no consumo de madeira serrada, pulando de 17,6 milhões de m<sup>3</sup> no ano de 1989, para algo em torno de 59,2 milhões de m<sup>3</sup> em 2010 (COUTO, 1995). Tais fatos são apontados como fatores que levaram à busca de espécies de rápido desenvolvimento para atender às demandas das indústrias madeireiras e de serrarias. No que diz respeito a espécies de rápido crescimento, o eucalipto se apresenta como uma opção de significativo potencial, pela sua capacidade produtiva, adaptabilidade aos diversos ambientes e pela variedade de espécies, tornando possível atender aos requisitos dos mais diversos da indústria madeireira (COUTO, 1995). No Brasil as espécies do gênero *Eucalyptus* são mais frequentes nas regiões Sudeste, Centro-Oeste e Nordeste.

A pouca utilização do eucalipto para obtenção de madeira serrada é devido a algumas características que a tornam de difícil desdobro, o que provoca nela o surgimento de vários defeitos, como rachaduras de topo, que inviabilizam o seu uso. Apesar da grande necessidade de madeira serrada para áreas como construção civil, móveis, etc., os eucaliptos eram pouco utilizados, exceto para a produção de carvão, celulose e chapas de fibras (SIMÕES et al., 1980). Mas, pesquisas científicas recentes (SILVA, 2003) começaram a mudar a história desta madeira no Brasil e no mundo. De madeira de má qualidade e acusado de agredir a natureza, o eucalipto vem se transformando em alternativa de madeira de qualidade para utilização na indústria de móveis e construção civil. A produção de eucalipto é apontada como tendo potencial de reduzir as pressões sobre florestas nativas, como a Amazônia, na obtenção de madeira de qualidade (ROCHA, 2000).

São cultivadas no Brasil várias espécies de eucalipto com uma grande faixa de densidade, constituição química e anatômica e, portanto, com grandes diferenças nas propriedades físicas e químicas da madeira. Estas variações tornam possível o uso desta madeira para os mais variados fins (VITAL e DELLA LUCIA, 1986). Pode-se dizer que o eucalipto tem todas as características para ser a principal madeira de serraria do país. Para isso é

necessária investigação intensiva, não só sob o ponto de vista tecnológico e silvicultural (SILVA, 2000; SILVA, 2003; ROCHA, 2000), mas também investimentos no processo de classificação das peças de madeira após a serragem.

## **2.1 Classificação de madeira serrada e automação**

Ao se pensar na utilização da madeira para fins mais nobres, como a produção de móveis e painéis, a classificação torna-se uma etapa de grande importância pois é através dela que se seleciona peças, após a serragem, com alto grau de qualidade.

Segundo CARREIRA (2003), na América do Norte a maior parte da madeira serrada, por volta do ano 2000, ainda era manualmente classificada. Porém o volume de madeira classificada mecanicamente vinha aumentando com o passar do tempo. Este processo era totalmente manual, realizado por indivíduos chamados graduadores, que ficavam próximos às serras, esperando as peças de madeira. Assim que desdobradas as toras, essas pessoas pegavam as tábuas e, olhando rapidamente as suas faces, faziam a identificação e quantificação dos defeitos, para então darem uma classificação final para a peça (CARREIRA, 2003). Como esta é uma tarefa estressante, pouco produtiva do ponto de vista do percentual de acerto, e razoavelmente insalubre devido ao barulho das serras e ao pó de madeira, pesquisas têm sido conduzidas no sentido da automação deste processo.

Nas duas últimas décadas esta tem sido área de interesse de pesquisadores individuais e grupos de pesquisa (CONNERS et al., 1989B), nas empresas privadas e universidades. Antes de 1989, pouca pesquisa científica havia sido desenvolvida nesta área.

CONNERS et al. (1983) desenvolveram e usaram uma técnica para localizar defeitos em imagens em escala de cinza de madeira bruta. A separação entre madeira e fundo era feita através de limiarização. Depois a imagem era dividida em blocos e um teste estatístico foi utilizado para separar

madeira limpa de defeito. No estágio de classificação tentaram diferenciar os tipos de defeitos através de análise de textura sobre dependência espacial dos níveis de cinza. A identificação de madeira limpa foi considerada alta (93,8%), mas a separação entre os tipos de defeitos não foi satisfatória.

CONNERS et al. (1989B) conduziram estudos com o intuito de testar a possibilidade de uso de um sistema de visão artificial para localizar variações na superfície de madeira serrada. Três experimentos de análise de imagens digitais foram conduzidos para investigar métodos de segmentação e problemas causados pela variação de coloração da madeira. Concluiu-se que, naquela época esta era uma tarefa difícil pois exigia muito recurso computacional, porém viável, exigindo pesquisa adicional.

Segundo FUNK et al. (2003) pesquisas foram conduzidas no sentido de mostrar que o uso de imagens RGB poderia incrementar a diferenciação entre defeitos em comparação com imagens em escala de cinza. Conseguiu-se melhorar em torno de 20% a taxa de acerto, mas observou-se também que o aumento significativo na quantidade de informações fornecidas pelas cores aumentaram também os desafios presentes no processo de análise como, por exemplo, o tempo de processamento.

CHO et al. (1990) construíram um sistema de visão artificial para identificação de defeitos de madeira bruta serrada. O sistema identificava quatro tipos de defeitos: nós, furos em geral, esmoado e “fendilhado/racha” (que, neste trabalho foram considerados como sendo apenas um mesmo defeito). Foram selecionadas 30 tábuas de madeira das seguintes espécies: cerejeira, carvalho, álamo-preto e bordo. O sistema foi testado com o objetivo de apenas identificar os defeitos, sem quantificá-los. Seu desempenho foi considerado limitado, mas a tarefa de identificação e quantificação de defeitos foi considerada viável através do uso de visão artificial.

Analisando as atividades nas serrarias, KLINE et al. (1992) observaram que a produção de madeira serrada para venda por classe de qualidade era considerada pequena devido a vários fatores: complexidade das normas de classificação, inabilidade e fadiga dos operadores, maquinário inadequado,

entre outros. Assim, concluíram que para melhorar a produção seria necessário a automação deste processo. Várias tecnologias foram estudadas como o uso de imagens RGB, laser para exame de superfície, raios-X, algoritmos computacionais, entre outras. Os mesmos autores concluíram então que, para que um protótipo de identificação, quantificação de defeitos e classificação de madeira serrada, retornasse resultados satisfatórios, seria necessário a integração de vários sistemas de exame da superfície da madeira, tendo os algoritmos computacionais várias fontes na tomada de decisão no processo de identificação dos defeitos.

KLINKHACHORN et al. (1995) também projetaram um sistema de visão artificial para classificação de madeira serrada com o objetivo de comparar seus resultados com o de classificadores humanos. Quinze tábuas de carvalho foram selecionadas, aplainadas e submetidas ao sistema. Quatro tábuas foram classificadas em classes inferiores e três em classes superiores, apresentando o sistema 55% de acerto.

Segundo FUNK et al. (2003), na investigação de nove algoritmos de segmentação para detecção de defeitos em madeira, concluiu-se que, de maneira geral, a maior fraqueza dos algoritmos está no estágio de segmentação da imagem.

Com o objetivo de melhorar a taxa de acerto alcançada por máquinas com apenas um sensor, na classificação de madeira serrada, KLINE et al. (2003) projetaram e construíram uma máquina com múltiplos sensores para esta tarefa. O protótipo possuía um mecanismo de leitura gerando imagens através de uma câmera RGB, um mecanismo de geração de imagens através de raios-x e outro mecanismo de geração de imagens através de raios laser. O sistema alcançou resultados melhores do que os de graduadores humanos que realizaram a mesma tarefa, com as mesmas tábuas, na linha de produção. Os graduadores, em média, apresentaram uma taxa de acerto em torno de 48%, enquanto a máquina de múltiplos sensores alcançou acerto de 63%.

Outros autores realizaram pesquisas apenas no intuito de otimizar

algumas das tarefas envolvidas no processo de classificação de madeira serrada.

FUNCK et al. (2003) avaliaram vários algoritmos de segmentação de imagens de madeira serrada para detectar seis tipos diferentes de defeitos. Os autores concluíram que, devido a complexidade do problema e à grande variedade de tipos diferentes de madeira, não existe um algoritmo que possa ser dito o melhor, mas sim aquele que retorna melhores resultados para classificação de uma espécie de madeira em específico.

KAUPPINEN (1999) testou um algoritmo de identificação de defeitos em imagens de madeira serrada. O autor transformou o espaço de cor RGB para outros espaços com o objetivo de verificar qual deles retornaria melhores resultados na identificação de defeitos. Ele concluiu que a melhor classificação foi obtida com o espaço RGB, porém foi baixa a diferença na classificação comparada com a cor transformada para escala de cinza, deixando claro a possibilidade de se utilizar câmeras monocromáticas neste processo.

Este mesmo autor pesquisou a identificação dos defeitos de tábuas de pinos por um classificador de k-vizinhos mais próximos, utilizando-se características de percentis do histograma das bandas RGB de imagens coloridas. Alcançou-se uma taxa de erro de classificação de aproximadamente 34%. O autor concluiu que as características de percentis são fáceis de se calcular em tempo real e possuem boa capacidade discriminante de defeitos de madeira.

KHOURY JUNIOR (2004) desenvolveu e testou Redes Neurais Artificiais (RNA) para identificar defeitos em madeira serrada de eucalipto. Foram utilizadas as características de percentis das bandas RGB de imagens obtidas com câmeras coloridas. Os classificadores foram considerados satisfatórios na separação entre defeitos e madeira limpa, e na identificação dos defeitos em madeira serrada desta espécie.

## **2.2 Reconhecimento de padrões para o processo de classificação**

Os humanos desenvolveram bem a habilidade de lidar com padrões. Eles são capazes de distingui-los com rapidez e clareza. É parte da nossa natureza, ao observar um objeto, fazer observações, as quais são comparadas com as informações e comportamentos previamente conhecidos por nós. Por esse processo somos capazes de reconhecer o objeto observado. Para nós esta é uma tarefa simples e automática, porém, no contexto da inteligência artificial, o reconhecimento de padrões não é uma tarefa assim trivial. Até os dias de hoje, ainda não existe nenhuma máquina ou software que emprega inteligência artificial capaz de reproduzir a capacidade de reconhecimento do cérebro dos seres humanos. Técnicas de reconhecimento de padrões são alvos de pesquisas desde os anos 60, época em que se iniciou o desenvolvimento da ciência da informação por intermédio da computação. Não foram poucos os métodos criados, nas quatro últimas décadas, com este fim, como Árvores de Decisão, Redes Neurais Artificiais, Algoritmos Genéticos, entre outros (CARVALHO, 2005; CERA, 2005; MÁXIMO e FERNANDES, 2005).

O reconhecimento de padrões, artificialmente, pode ser dividido em duas categorias principais: a de reconhecimento de objetos concretos e a de identificação de itens abstratos. Os elementos abstratos seriam itens sem forma física como, por exemplo, a resolução para um determinado problema. O reconhecimento de itens concretos tem como alvo, por exemplo, impressões digitais, assinaturas, objetos das mais variadas formas, faces, enfim, itens do mundo real, com os quais lidamos no nosso dia a dia (CERA, 2005).

Considerando então uma base de dados constituída de diversos exemplos, e que cada um possui um tipo específico de padrão associado a ele, um padrão nada mais é do que o tipo do exemplo, um rótulo que o caracterize ou que o classifique. Os exemplos das bases de dados são geralmente medições ou observações sobre determinado assunto, definindo o

domínio do processo de aprendizagem (CARVALHO, 2005).

Assim, a solução do problema de reconhecimento de padrões envolve três níveis de processamento para cada novo elemento a ser inserido na base de dados (CERA, 2005): a) filtragem da entrada; b) extração de características do objeto a ser classificado e; c) sua classificação propriamente dita.

A etapa de filtragem da entrada visa eliminar dados desnecessários ou distorcidos fazendo com que esta apresente apenas dados relevantes para discriminação do objeto que precisa ser reconhecido. A etapa de extração de características consiste da análise dos dados de entrada a fim de se obter informações que serão usadas no processo de reconhecimento. A fase de classificação é aquela onde, através de estudo das características da entrada filtrada, o objeto em análise é reconhecido como pertencente a uma determinada categoria dentre as várias categorias conhecidas previamente, ou como integrante de um dos grupos formados a partir de um certo instante (CERA, 2005).

De acordo com TOU e GONZALEZ (1974) são três as metodologias mais usadas para o reconhecimento automático de padrões: a heurística, a sintática e a matemática.

Os métodos heurísticos baseiam-se na intuição e experiência humanas. Cada problema requer a aplicação de regras de decisão especializadas, de forma que a estrutura e desempenho dependem do conhecimento prévio que se tem do problema.

Os métodos sintáticos (também conhecidos por linguísticos) fazem uso do conceito de propriedades comuns para descrever as relações hierárquicas existentes entre os elementos, e, portanto, caracterizá-los por intermédio de elementos primitivos.

Os métodos matemáticos baseiam-se em regras de classificação, utilizando propriedades comuns e conceitos de agrupamento, expressos na forma de equações matemáticas para esse fim. Tais métodos são determinísticos ou estatísticos. Determinísticos quando baseiam-se em uma estrutura matemática que não emprega explicitamente as propriedades

estatísticas das classes em consideração. Estatísticos quando baseiam-se em regras de classificação matemática que são derivadas de uma estrutura estatística.

Classificadores estatísticos são derivados do teorema de Bayes, no qual determina-se a probabilidade a *posteriori*,  $P(C_j|v)$ , de que um indivíduo  $v$  seja classificado na classe  $C_j$ . A probabilidade a *posteriori* é obtida para cada uma das classes  $C_j$  por meio do produto da probabilidade a *priori* de ocorrência da classe,  $P(C_j)$ , pela probabilidade condicional da característica  $v$  dentro da classe  $C_j$ ,  $P(v|C_j)$ . A probabilidade a *priori*  $P(C_j)$  pode ser estimada pela frequência de ocorrência das classes na amostra. O indivíduo será classificado na classe que apresentar maior valor da probabilidade a *posteriori*  $P(C_j|v)$ .

Assumindo uma distribuição gaussiana para o valor de  $P(v|C_j)$ , pode-se estimar a probabilidade a *posteriori*,  $P(C_j|v)$ , pela seguinte expressão (GONZALES e WOODS, 2000):

$$D_j(v) = -\frac{1}{2}[(v - \bar{v}_j)^T (M_j)^{-1} (v - \bar{v}_j)] - \frac{1}{2} \ln |M_j| + \ln(P(C_j)) \quad (1)$$

em que:  $D_j(v)$  = probabilidade a *posteriori* ou valor da função discr. da classe  $C_j$ ;

$v$  = vetor (coluna) de características de um dado bloco;

$\bar{v}_j$  = vetor (coluna) de média das características da classe  $C_j$ ;

$M_j$  = matriz de covariância da classe  $C_j$ .

O desenvolvimento dos classificadores pode ser realizado de duas maneiras: considerando-se uma covariância para cada classe, ou que todas as classes possuam uma covariância comum (KHOURY JUNIOR, 2004). A covariância comum para as classes é obtida pela média aritmética simples das covariâncias das classes. Assim, considerando uma mesma matriz de covariância para todas as classes, a Equação 1 assume a forma reduzida da Equação 2 e é conhecida como função discriminante linear:

$$D_j(v) = -\frac{1}{2} \bar{v}_j^T (M)^{-1} \bar{v}_j + v^T (M)^{-1} \bar{v}_j + \ln(P(C_j)) \quad (2)$$

em que:  $D_j(v)$  = valor da função discriminante linear da classe  $C_j$  dado  $v$ ;

$v$  = vetor (coluna) de características de um dado bloco;

$\bar{v}_j$  = vetor (coluna) de média das características da classe  $C_j$ ;

$M$  = matriz de covariância comum das classes  $C_j$ .

Após determinados os valores de  $D_1(v)$ ,  $D_2(v)$ , ...,  $D_j(v)$  das  $j$  classes de defeitos da tábua, o maior desses valores é onde o indivíduo  $v$  tem maior probabilidade de pertencer.

VIEIRA (2000) diz que embora a normalidade assumida algumas vezes não seja observada na prática, esse classificador apresenta resultados satisfatórios, e que a função de densidade normal multivariada é bastante utilizada pela facilidade de entendimento. O autor diz ainda que o desempenho do classificador depende do quanto as estimativas do “vetor de médias amostrais” e da matriz de “variância e covariância amostral” se aproximam dos respectivos “vetor real de médias da população” e “matriz de variância e covariância populacional”, pois a sub-amostragem pode prejudicar a acurácia no processo de classificação.

Os classificadores também podem ser divididos em supervisionados e não supervisionados.

Quando usa-se um conjunto de dados conhecidos e disponíveis, e que o classificador foi desenhado explorando esta informação conhecida à priori, este processo é designado por reconhecimento supervisionado de padrões (MÁXIMO e FERNANDES, 2005). Porém, existe um outro tipo de tarefa de reconhecimento de padrões para os quais dados de classes conhecidas não estão disponíveis. Neste tipo de problema, tem-se um conjunto de vetores de características  $v$  e o objetivo é decifrar as semelhanças fundamentais e agrupá-los por critérios de similaridade. Já neste caso o processo é designado

de reconhecimento não-supervisionado de padrões. Um algoritmo de agrupamento pode ser utilizado para revelar os grupos nos quais vetores de características são agrupados no espaço de características N dimensional. Os pontos que correspondem a informações semelhantes são passivos de serem agrupados.

Outro aspecto importante no reconhecimento de padrões não supervisionado é a definição de “semelhança” entre dois vetores de características e a escolha da medida apropriada para o efeito. É preciso que se faça a escolha de uma estrutura algorítmica que irá agrupar os vetores com base na medida de semelhança adotada, pois algoritmos diferentes podem levar a resultados distintos, necessitando da avaliação crítica do conhecedor do problema na interpretação dos resultados.

Como a classificação supervisionada é o processo no qual amostras de identidade conhecida são utilizadas para classificar instâncias de identidade desconhecida, é preciso que um perito identifique nas imagens de treinamento as classes de informação de seu interesse e separe por regiões que melhor representem estas classes. Dentro destas regiões ele separa áreas de treinamento que contêm os elementos que serão usados como representantes da classe. A partir deles são extraídas determinadas características das classes, que serão usadas para a classificação dos demais indivíduos.

Segundo VIEIRA (2000), o processo de classificação supervisionada apresenta vantagens e desvantagens em relação à classificação não-supervisionada. A tabela a seguir é um quadro comparativo entre os dois processos:

**Tabela 1:** Classificação supervisionada versus classificação não supervisionada: vantagens e desvantagens.

	<b><i>Não supervisionada</i></b>	<b><i>Supervisionada</i></b>
<b>Vantagens</b>	<ul style="list-style-type: none"> <li>→ Não exige conhecimento prévio profundo das classes de interesse;</li> <li>→ Minimiza a possibilidade de erro humano;</li> <li>→ As classes identificadas são mais uniformes no que diz respeito a composição espectral.</li> </ul>	<ul style="list-style-type: none"> <li>→ O analista tem maior controle sobre o processo (escolhe o número e o tipo de classes, as regiões de amostragem e o número de amostras). Esta característica é importante principalmente quando a classificação visa a um objetivo específico;</li> <li>→ A classificação supervisionada é balizada pelas classes pré-definidas pelo analista, desta forma não é necessário agrupar partes da imagem classificada para formar conjuntos de elementos similares para se definir as classes de interesse;</li> <li>→ O analista tem condições de identificar possíveis imprecisões graves pela análise dos dados de treinamento.</li> </ul>
<b>Desvantagens</b>	<ul style="list-style-type: none"> <li>→ Dificuldades podem surgir na formação dos grupos de interesse;</li> <li>→ Controle limitado na formação de classes e definição de suas identidades;</li> <li>→ As propriedades espectrais das classes podem mudar com o passar do tempo;</li> <li>→ Em geral assume agrupamentos em forma esférica, o que nem sempre é verdade.</li> </ul>	<ul style="list-style-type: none"> <li>→ O analista impõe uma determinada estrutura de classificação aos dados, através da definição prévia das classes de informação. Estas classes podem não corresponder ao total das classes reais existentes nas cenas a serem imageadas, ou não ser separáveis no espaço N-dimensional;</li> <li>→ As áreas de treinamento selecionadas podem não apresentar todas as características das classes que representam. Esta característica pode ter muita influência, principalmente quando a área a ser classificada é grande, complexa ou inacessível;</li> <li>→ Os dados de treinamento são ordinariamente separados em categorias pré-definidas que, podem não representar todas as categorias existentes.</li> </ul>

Fonte: VIEIRA (2000).

Os classificadores de padrões por funções de distância são simples de implementar e intuitivos pois permitem fácil entendimento do problema. Na

técnica de Distância Mínima, por exemplo, as áreas de treinamento definem regiões no espaço de dados N-dimensional. Cada uma das regiões pode ser representada por um ponto, denominado centróide. O processo de classificação baseia-se na projeção de cada vetor de padrões de entrada neste espaço N-dimensional e no cálculo da distância N-dimensional entre ele e os centróides. O vetor será classificado na classe que corresponder à menor distância calculada.

Espera-se que esse método produza resultados práticos e satisfatórios apenas quando as classes tendem a se apresentar desagrupadas. Nesse método, algoritmos de agrupamento de classes desempenham papel importante. Uma vez que a proximidade de um padrão desconhecido ao padrão das classes será utilizado para sua classificação, utiliza-se o termo Classificador de Distância Mínima (GONZALES e WOODS, 2000).

No caso N-dimensional, cada classe é representada por um vetor médio dos vetores das características dos elementos que a compõem, calculado previamente a partir dos dados disponíveis.

Muito embora seja de implementação simples, esta técnica é bastante influenciada pela sobreposição de classes e pela variabilidade dentro de cada classe. É possível construir modelos sofisticados deste classificador usando diferentes técnicas para medição de distâncias e para a definição dos centróides. Uma maneira de atribuir um vetor desconhecido a uma classe é calculando-se a “distância euclidiana simples” dele a todas as classes. Aquela classe cuja distância computada for a menor, o vetor será atribuído como pertencente a mesma (GONZALES e WOODS, 2000).

Assim, minimizar a probabilidade de erro é o equivalente a particionar o espaço de características em J regiões, para uma tarefa com J classes. Se duas ou mais regiões já forem claramente separadas, ou seja, formam agrupamentos distintos e bem definidos, as superfícies de decisão que as separam também fica claramente definida. Porém, se as regiões  $J_i$  e  $J_j$  forem contíguas, então elas serão separadas, utilizando-se algum critério, em algum ponto intermediário por uma superfície de decisão no espaço multidimensional

de características.

### **2.3 Análise de Componentes Principais**

A ACP tem como objetivo reduzir o número de variáveis, com o intuito de facilitar a análise e interpretação dos dados, ou seja, parte-se do princípio que, combinando um conjunto de variáveis quaisquer, pode-se construir um novo agrupamento, menos numeroso, de variáveis principais e que resumam a informação contida no grupo inicial. Isto pode ser feito utilizando-se a matriz de covariância ( $M_c$ ) por meio de poucas combinações lineares dos dados originais. Essas combinações geram coordenadas, conhecidas como componentes principais, não correlacionadas entre si. Essa técnica procura verificar se poucos componentes explicam a maior parte da variação dos dados. O primeiro componente principal é o que carrega a maior proporção da variância dos dados originais. O segundo componente, ortogonal em relação ao primeiro, é o próximo que carrega a maior proporção da variância dos dados originais explicando aquela parte da variância não explicada pelo primeiro, e assim sucessivamente. A falta de correlação entre os componentes significa que eles estão medindo diferentes dimensões dos dados (REGAZZI, 2000).

Sendo o número de variáveis originais igual a  $p$ , para se reproduzir a variabilidade total dos dados, são necessários  $p$  componentes principais, mas, em geral, a maior parte desta variabilidade deve-se a um número  $k$  de componentes menor que  $p$ . Assim, as  $p$  variáveis originais com  $n$  observações cada, podem ser reduzidas a um conjunto com o mesmo número de observações mas com apenas  $k$  componentes principais (FERREIRA, 1996).

Na ACP os autovalores  $\lambda_i$  da matriz  $M_c$  e os seus correspondentes autovetores  $e_i$  são calculados, sendo que os componentes principais são combinações lineares das  $p$  variáveis aleatórias  $(x_1, x_2, \dots, x_{p-1}, x_p)$  calculados por:

$$\begin{aligned}
Y_1 &= e_1' x = e_{11}x_1 + e_{12}x_2 + \dots + e_{1(p-1)}x_{p-1} + e_{1p}x_p \\
Y_2 &= e_2' x = e_{21}x_1 + e_{22}x_2 + \dots + e_{2(p-1)}x_{p-1} + e_{2p}x_p \\
&\vdots \quad \vdots \quad \vdots \quad \vdots \quad \ddots \quad \vdots \quad \vdots \\
Y_p &= e_p' x = e_{p1}x_1 + e_{p2}x_2 + \dots + e_{p(p-1)}x_{p-1} + e_{pp}x_p
\end{aligned} \tag{3}$$

em que:

$Y_i$  = componente principal  $i$  ( $1 \leq i \leq p$ );

$e_{ij}$  = elemento  $j$  do autovetor  $i$  ( $1 \leq i \leq p$ ,  $1 \leq j \leq p$ );

$x_i$  = característica  $i$  do vetor  $X$ .

A proporção da variância total explicada pelo  $k$ -ésimo componente principal ( $Y_k$ ) é obtida pela Equação 4:

$$P_{(Y_k)} = \frac{\lambda_k}{\sum_{i=1}^p \lambda_i}, k=1,2, \dots, p. \tag{4}$$

em que:

$P_{(Y_k)}$  = proporção da variância total explicada pelo  $k$ -ésimo CP ( $Y_k$ );

$\lambda_i$  = autovalores da matriz de covariância ( $1 \leq i \leq p$ ).

A proporção da variação total explicada pelos  $k$  primeiros componentes principais é obtida pela Equação 5:

$$P_{(Y_{1-k})} = \frac{\sum_{i=1}^k \lambda_i}{\sum_{j=1}^p \lambda_j}, k=1,2, \dots, p. \tag{5}$$

em que:

$P_{(Y_{1-k})}$  = proporção da variação total explicada pelos  $k$  primeiros CPs;

$\lambda_i$  = autovalores da matriz de covariância ( $1 \leq i \leq p$ ).

O coeficiente de correlação entre os componentes principais  $Y_i$  e as

variáveis originais  $x_k$  é calculado utilizando-se a Equação 6:

$$\rho_{Y_i, x_k} = e_{ki} \frac{\sqrt{\lambda_i}}{\sqrt{\sigma_{kk}}}; i, k = 1, 2, \dots, p. \quad (6)$$

*em que:*

$\rho_{Y_i, x_k}$  = coef. de correlação entre os CPs  $Y_i$  e as variáveis originais  $x_k$ ;

$e_{ki}$  = autovetor  $i$  do autovalor  $k$  da matriz de covariância;

$\sigma_{kk}$  = variância da variável original  $k$ .

Quando a maior parte da variância estiver acumulada nos primeiros componentes principais, então esses podem substituir as  $p$  variáveis originais sem muita perda de informação.

A padronização dos dados (média zero e variância um) é recomendada quando uma variável tem variância muito mais elevada que as outras, portanto teria uma influência grande no primeiro componente principal (REGAZZI, 2000).

As variáveis padronizadas são adimensionais, sendo de variação comparável, pois estão reduzidas à mesma escala, colocando dados discrepantes em contato com os grupos aos quais tem maior similaridade.

A ACP também pode ser utilizada para descarte de variáveis originais, adotando-se o critério da avaliação da importância das variáveis a partir das correlações entre cada variável original e os últimos componentes que retêm proporção mínima da variação total (REGAZZI, 2000). Assim, a variável que possui maior correlação com o componente principal que possui o menor autovalor é a menos importante para explicar a variância total e, portanto, passível de descarte. A seguir, o próximo componente de menor autovalor é examinado, podendo-se descartar a variável com a maior correlação com ele, que não tenha sido previamente descartada.

Apesar da grande aceitação e uso da ACP pela comunidade científica, algumas pesquisas criticam o procedimento de descarte de variáveis através

desta técnica, pela possibilidade de se eliminarem variáveis que possuem pesos consideráveis nos primeiros componentes, especialmente quando estes acumulam parte significativa da variação total (ALVES 2002). No entanto, alguns pesquisadores avaliaram a eficiência do descarte comparando os resultados do processamento com as variáveis originais e as selecionadas (ALVES, 2002). Eles concluíram pela eficiência do método pois observaram que foi eficiente o uso de ACP como critério de descarte.

#### ***2.4 Avaliação de um classificador***

O critério de classificação deve ser avaliado de acordo com o desempenho do classificador na classificação de instâncias que não foram usadas na sua fase de desenvolvimento (SAS, 1999). Quando o mesmo conjunto de dados é utilizado tanto para definir como para avaliar o classificador, a contagem de erro obtida apresenta uma tendência otimista e é denominada taxa de erro aparente (CONGALTON, 1991B). Para reduzir a tendência, o conjunto de dados pode ser dividido em dois, um para obtenção da função discriminante e outro para a estimativa da taxa de erro. A desvantagem desse método é a redução do tamanho da amostra.

CONGALTON (1991B) afirma que a matriz de confusão (também chamada de matriz de erros ou tabela de contingência) é uma forma útil para representar a exatidão, uma vez que as exatidões das diferentes classes estão claramente descritas, mostrando os erros de inclusão (comissão) e de omissão (exclusão) presentes na classificação. Uma matriz de confusão é um arranjo quadrado em linhas e colunas e que expressa o número de amostras de cada classe atribuídas às diferentes classes. Esta matriz é o ponto de partida para uma série de técnicas estatísticas descritivas e analíticas. Alguns autores sugerem que a técnica descritiva mais simples seja a exatidão global, que pode ser computada dividindo-se o total de amostras corretamente classificadas (soma dos valores da diagonal principal) pelo número total de

itens na matriz. A exatidão global é calculada utilizando-se a Equação 7:

$$E = 100 \frac{1}{n} \sum_{i=1}^j x_{ii} \quad (7)$$

*em que*, E = porcentagem de exatidão global na classificação;

$x_{ii}$  = valor na linha i e coluna i da matriz de confusão;

n = número total de amostras submetidas a classificação;

j = número total de classes.

CONGALTON (1991A) afirma que tradicionalmente utiliza-se o número total de amostras da classe submetidas à classificação para cálculo da exatidão das classes. O valor encontrado é denominado exatidão do produtor. Por outro lado, se for utilizado o número de amostras efetivamente atribuídas às classes, o valor encontrado é denominado exatidão do usuário. A exatidão do produtor indica a probabilidade de uma amostra ser corretamente classificada e é uma medida do erro de omissão. Por outro lado, a exatidão do usuário indica a probabilidade de um item classificado representar aquela classe, e é uma medida do erro de comissão (STORY e CONGALTON, 1991).

ROSENFELD e FITZPATRICK-LINS (1991) avaliaram um coeficiente de concordância denominado Kappa. Esta estatística é uma medida do acerto real menos o acerto por acaso:

$$K = \frac{E - P_c}{1 - P_c} \quad (8)$$

*em que*, K = estimativa do coeficiente Kappa;

E = proporção de itens classificados corretamente (exatidão global);

$P_c$  = proporção esperada de acerto ao acaso.

O coeficiente Kappa possui uma escala de variação de valores entre

-1,00 e 1,00. Quanto maior o valor obtido para o Kappa melhor foi o resultado da classificação. O valor zero significa que o acerto obtido pelo classificador foi igual ao acerto numa classificação ao acaso. Valores positivos de Kappa significam que o acerto é maior que uma classificação ao acaso e negativos significam que o acerto é pior que uma classificação ao acaso.

O Kappa é consensualmente considerado como coeficiente que estabelece relação intra-classes, ou seja, leva em consideração a matriz de confusão como um todo, não usando apenas os dados da diagonal principal (acerto global) mas também os demais dados presentes nas outras células da matriz (CONGALTON, 1991B; VIEIRA, 2000; ROSENFELD e FITZPATRICK-LINS, 1991).

Apesar do coeficiente de concordância Kappa vir sendo amplamente utilizado em medidas de acurácia de classificadores, não há uma fundamentação teórica para indicar quais os níveis de variação aceitáveis deste coeficiente numa classificação. Segundo GABOARDI (2003), existem duas propostas de escala para associação de um nível de qualidade da classificação dos valores de Kappa. Uma sugerida por J. L. Fleiss em 1981, e outra por LANDIS e KOCH (1977). Estes conceitos de Kappa e seus respectivos valores são apresentados na Figura 1:



Figura 1 - Conceitos do coeficiente Kappa.

### **3 MATERIAL E MÉTODOS**

O projeto foi executado em três etapas distintas. Na primeira, foi feita a integração entre uma câmera RGB e um protótipo que operasse de maneira semelhante às esteiras transportadoras das serrarias. Esta etapa foi conduzida no Laboratório de Mecanização Agrícola do Departamento de Engenharia Agrícola da Universidade Federal de Viçosa-MG. Na segunda etapa foram desenvolvidos módulos de processamento e análise de imagens para classificar blocos de imagens de madeira serrada e para classificação das tábuas. Na terceira etapa estes módulos foram integrados ao protótipo para a sua avaliação do desempenho operacional. Estas duas últimas etapas foram desenvolvidas no Laboratório de Projeto de Máquinas e Visão Artificial do Departamento de Engenharia Agrícola da Universidade Federal de Viçosa.

#### ***3.1 Apresentação do protótipo***

Foi construído um protótipo de um sistema de visão artificial que simula a esteira transportadora de madeira serrada de uma indústria, de acordo com o esquema da Figura 2.

O protótipo era composto por um sistema de aquisição de imagem perpendicular ao deslocamento da tábua, com iluminação controlada, acoplada a uma placa de aquisição de vídeo instalada em um microcomputador. Ele possuía uma esteira rolante, de 0,41 metros de largura e 3 metros de comprimento, com velocidade controlável. A esteira era impulsionada por um motor de indução trifásico de 0,7355 kW (1 cv), da marca

Eberle, 2,95A/220v/1730rpm, modelo B80a4. A velocidade da esteira foi regulada utilizando um inversor de frequência CIMR-V7AA21P5, da Yaskawa Electric Corporation, ajustado em 9,97 Hertz para trabalhar a aproximadamente  $0,61 \text{ m s}^{-1}$ .

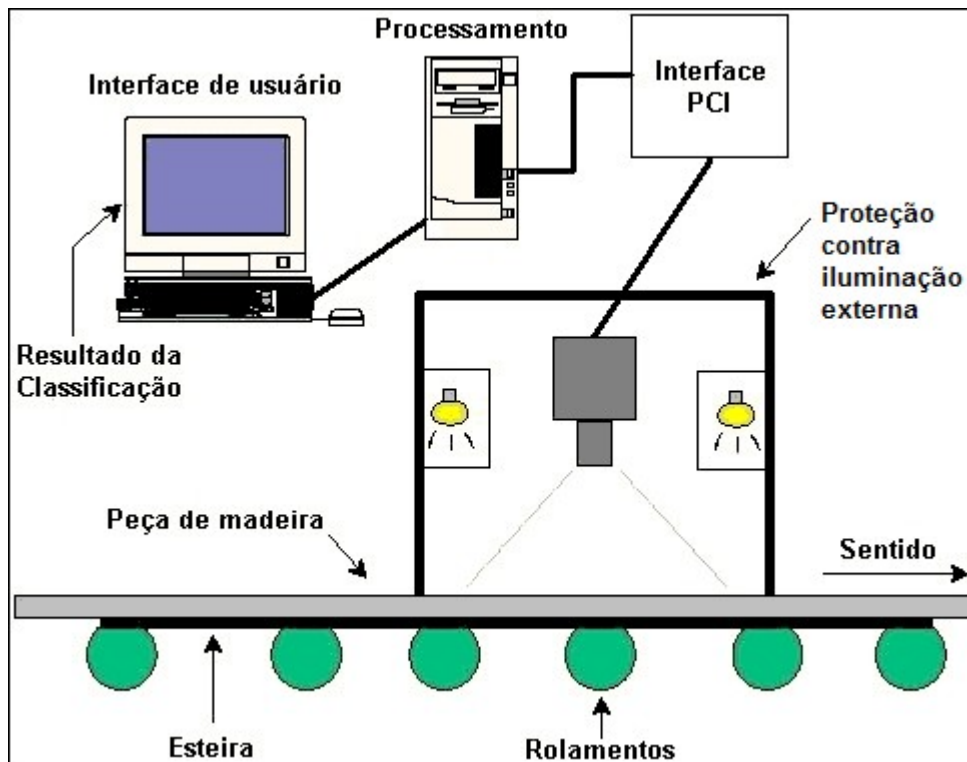


Figura 2 - Visão geral do protótipo de classificação de madeira serrada por um sistema de visão artificial.

As tábuas foram manualmente inseridas no protótipo sobre a esteira rolante. A esteira conduziu a tábua até a câmera que por sua vez captou a imagem desta, em partes, através do software desenvolvido utilizando a linguagem C++ (HORTON'S, 1998; HORTON'S, 1997). Um programa de computador foi desenvolvido, usando o programa Matlab<sup>®</sup> versão 6.5, para montar a imagem final para posterior processamento. Após obtida a imagem de um lado da tábua, a mesma foi repassada no protótipo para aquisição da imagem da outra face, para então determinar a classificação final.

A iluminação foi feita por duas lâmpadas halógenas de 300W instaladas em refletores com difusor de luz. As imagens foram obtidas então em

condição de iluminação controlada. Foi montada uma estrutura metálica de 2,12 metros de altura por 0,86 x 0,67 metros de comprimento e largura, encapsulada por material plástico completamente opaco que continha aberturas nas laterais esquerda e direita para a justa colocação dos refletores e, uma pequena abertura superior apenas para a passagem do cabo de dados e de alimentação da câmera.

A câmera de vídeo era digital, da JAI Corporation, modelo CV-M7+/CL, colorida. Ela foi acoplada ao computador pessoal através de uma placa de aquisição de vídeo da Coreco Imaging, Inc., modelo PC-CamLink, padrão CamLink (PULNIX, 2005). A lente utilizada foi uma CCTV 20HC, da TAMRON Co. Ltd., com distância focal de 25mm, *CCD* de 2/3", *Mount Type C*. A lente permitia o controle manual do foco e da abertura do diafragma (ambos com trava), sendo que para obtenção de imagens o *f-stop* foi definido em 3,5. A câmera foi colocada a 0,91 metros de altura de modo a fornecer uma resolução de 0,24x0,24 mm/pixels semelhante a utilizada por outros autores (KOIVO e KIM, 1989; KHOURY JUNIOR, 2004; KAUPPINEN, 1999) em sistemas de classificação de defeitos em madeira. Esta câmera era equipada com um sensor CCD que permitia aquisição de imagens RGB de 1392(H) x 1024(V) pixels. Via software o tempo de exposição do CCD (*shutter speed*) foi ajustado para 1/800s.

O cabo de dados, ligado a placa de aquisição de vídeo instalada no computador, enviava as imagens capturadas da câmera à memória principal para processamento.

Além da linguagem C++, foi utilizado o conjunto de bibliotecas de funções fornecidas pelo fabricante da câmera para o desenvolvimento do algoritmo de aquisição das imagens.

### **3.2 Interface de usuário**

Foi desenvolvida uma interface de usuário para uma melhor interação com o protótipo. Nesta interface o usuário podia iniciar o processo de

classificação de duas maneiras distintas: *interativa*, em que o sistema fornecia ao usuário informações parciais sobre o processo de classificação da tábua corrente, ou *automática*, em que o usuário apenas escolhia a norma de classificação a ser utilizada e o sistema ficava trabalhando continuamente, bastando apenas inserir outra tábua após a visualização do resultado final de classificação da tábua anterior.

### **3.3 Desenvolvimento do classificador estatístico**

O classificador foi desenvolvido a partir de características extraídas de tábuas amostradas de eucalipto, secas ao ar livre, oriundas da ARACRUZ Celulose S.A. e adquiridas por intermédio do Laboratório de Propriedades Físicas e Mecânicas da Madeira da Universidade Federal de Viçosa. As amostras tiveram seus defeitos quantificados manualmente e foram etiquetadas para obtenção das imagens. Das 53 tábuas obtidas, foram retiradas 11 amostras escolhidas de forma que tivessem os defeitos visuais mencionados na Norma NBR 14.806 (ABNT, 2002B), segundo as definições da NBR 12.551 (ABNT, 2002A), a saber: bolsa de goma, esmoado, fendilhado, podridão (*ou madeira ardida*), medula, nó cariado, nó solto encaixado, nó solto oco, racha na extremidade e furo de inseto.

Foram feitas simplificações na norma da ABNT com base na análise prévia dos dados recolhidos e em algumas das conclusões do trabalho de KHOURY JUNIOR (2004). Este autor estudou as classes de defeitos consideradas pela norma da ABNT e observou que poderiam ser usados alguns critérios de agrupamentos para as classes baseados na sua importância (peso igual na norma de classificação) e separabilidade entre as classes (quais classes que se confundem para o classificador estatístico).

Portanto, utilizando algumas das conclusões do autor e a análise prévia feita dos dados, foram determinados para este trabalho os seguintes agrupamentos de classes:

- *agrupamento 1*: madeira limpa;
- *agrupamento 2*: bolsa de goma, podridão, nó cariado, nó solto encaixado e nó solto oco;
- *agrupamento 3*: esmoado;
- *agrupamento 4*: fendilhado;
- *agrupamento 5*: nó sadio e nó sadio corrompido;
- *agrupamento 6*: racha.

A Figura seguinte exemplifica os defeitos contidos nos agrupamentos.

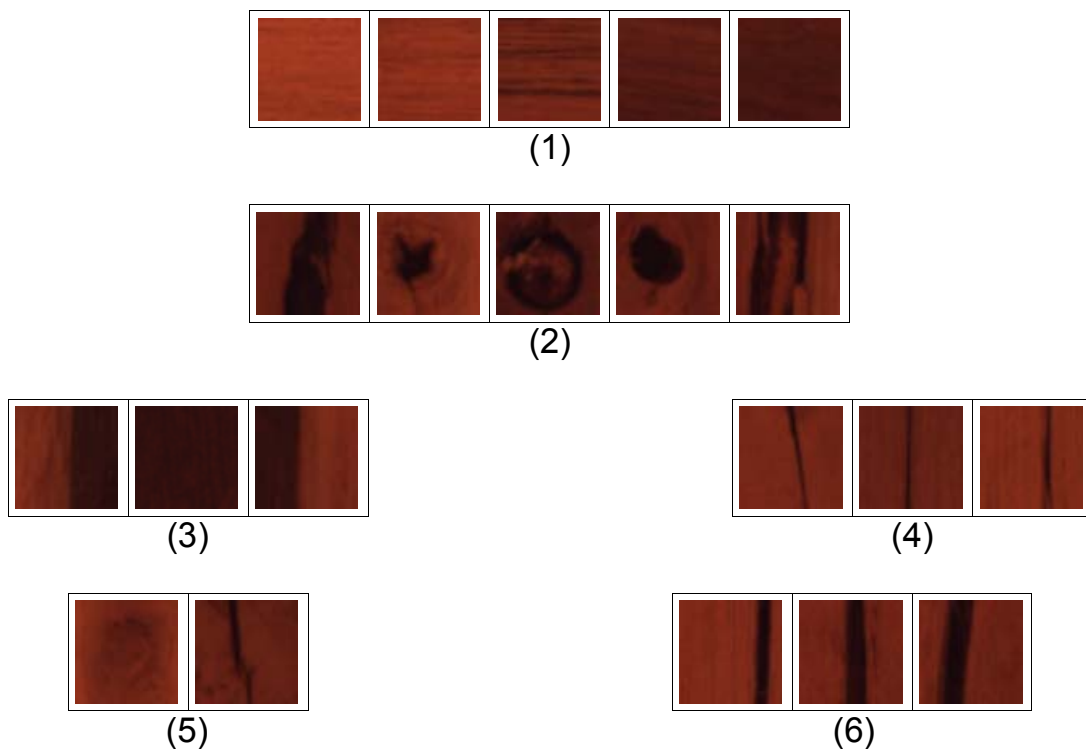


Figura 3 – Exemplos de blocos de imagem 64x64 pixels para os agrupamentos: **Agrupamento 1** – Madeiras Limpas, **Agrupamento 2** – bolsa de goma, nó cariado, nó solto encaixado, nó solto oco e podridão, **Agrupamento 3** – Esmoados, **Agrupamento 4** – Fendilhados, **Agrupamento 5** – Nó sadio e nó sadio corrompido, **Agrupamento 6** – Rachas.

Onze tábuas foram imageadas no protótipo, nas duas faces largas, com a iluminação artificial controlada.

Elas tiveram todos os seus defeitos identificados visualmente e recortados manualmente usando-se o programa Gimp v2.2.8, da GNU

Foundation. Foram utilizados blocos de 64x64 pixels. Os defeitos menores que um bloco foram recortados de modo a ocupar o bloco na posição central do mesmo, e os maiores foram divididos em mais de um bloco. Um subconjunto dos blocos gerados foi usado para o desenvolvimento do classificador.

Calculou-se então as características do histograma para os blocos da imagem original: média, mediana, desvio padrão e valores de percentis 2, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90 e 95% (KAUPPINEN, 1999). Estas quinze características foram calculadas para cada uma das bandas vermelho, verde e azul, totalizando então um vetor coluna com 45 características (ou descritores) para cada bloco.

Com o intuito de minimizar a diferença de iluminação entre as cenas, a diferença entre os percentis 2 e 95% foi utilizada para normalização do percentil de cada bloco, por meio da seguinte fórmula:

$$C_B^x = \frac{P_B^x}{P_B^{95} - P_B^2} \quad (9)$$

*em que:*  $C_B^x$  = característica normalizada do percentil x;

$P_B^x$  = nível de cinza do percentil x;

x = 2, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 95%.

Para os agrupamentos pré-definidos, foram utilizados, no desenvolvimento do classificador, as seguintes quantidades de blocos:

- agrupamento 1: 500 blocos;
- agrupamento 2: 80 blocos;
- agrupamento 3: 80 blocos;
- agrupamento 4: 60 blocos;
- agrupamento 5: 14 blocos;
- agrupamento 6: 80 blocos.

O número de blocos para representar cada agrupamento foi escolhido de modo a cobrir as variações de forma e tamanho que os mesmos apresentavam nas tábuas para se evitar o problema da amostragem desigual, ou seja, sub-amostragem de alguns agrupamentos e super-amostragem de outros. A exceção foi madeira limpa que representava mais de 90% do total de amostras, mas apenas um pequeno subconjunto bem diversificado desta foi utilizado.

Foi então desenvolvido e testado um classificador estatístico de blocos das imagens com base no vetor de características  $v$ . Este classificador foi composto de funções discriminantes no espaço das características (GONZALEZ e WOODS, 2000).

Os parâmetros gerados, a partir desses blocos, para serem usados pelo classificador no cálculo dos valores das 6 funções discriminantes (uma para cada agrupamento), foram:

- vetor de características de cada bloco:  $v$ ;
- vetor de média das características de todos os blocos de cada agrupamento:  $\bar{v}_j$ ;
- matriz de covariância comum dos agrupamentos:  $M_c$ ;
- inversa da matriz de covariância comum dos agrupamentos:  $M_c^{-1}$ .

O parâmetro “matriz de covariância comum” dos agrupamentos foi calculado ponderando-se a matriz de covariância individual de cada agrupamento em relação ao número de blocos amostrados para cada uma dos agrupamentos. Segundo SEARLE (1982), quando o número de blocos usados para a obtenção de vetores de características for diferente para cada agrupamento, deve-se gerar individualmente cada matriz de covariância e depois fazer a ponderação em relação ao número de blocos usados para aquele agrupamento. Em seguida pôde-se fazer a média aritmética simples dessas matrizes para se obter a matriz de covariância comum.

Primeiramente, foram desenvolvidas funções discriminantes utilizando

apenas estas 45 características retiradas do histograma dos blocos da imagem. Os blocos classificados como defeitos de tábuas foram identificados de tal forma que, ao final desta etapa, se obteve uma quantificação de todos os defeitos existentes na tábua. Para aqueles defeitos que apresentaram maior índice de confusão, foi desenvolvido e utilizado outro processo discriminante utilizando as características morfológicas e informações de posição. Foi usado o programa SAS (Statistical Analysis System) para validação das funções discriminantes (SAS, 1999; CENAPAD, 2002).

### ***3.4 Eliminação de características por Análise de Componentes Principais***

Na ACP, a variável que possui maior correlação com o componente principal que possui o menor autovalor é a menos importante para explicar a variância total e, portanto, passível de descarte. A seguir, o próximo componente de menor autovalor é examinado, podendo-se descartar a variável com a maior correlação com ele, que não tenha sido previamente descartada, e assim sucessivamente.

Esta técnica foi utilizada para identificar o conjunto de variáveis potenciais para o descarte de características e classificá-las em ordem crescente de importância. As características menos importantes foram então eliminadas, uma a uma, e os parâmetros do classificador estatístico foram recalculados com as restantes. No momento em que a eliminação de uma delas causou a redução do acerto global o processo foi interrompido. Dessa maneira, foram eliminadas apenas as características que não implicaram na redução do valor do acerto global da classificação das tábuas.

### ***3.5 Processamento das imagens***

As imagens das 11 tábuas foram também utilizadas para desenvolvimento e teste do algoritmo, composto por duas etapas: processamento e análise de imagens. O processamento de imagens teve o

objetivo de segmentar os objetos de interesse (fundo, madeira limpa e defeitos). A análise de imagens teve o objetivo de identificar a qual classe de interesse a tábua pertencia, utilizando os critérios das normas ABNT e de uma classificação comercial.

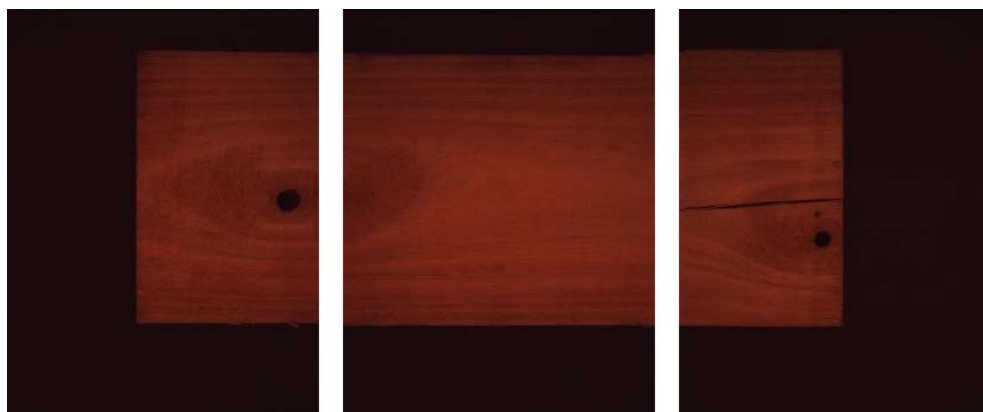
O software desenvolvido em C++ obtinha uma imagem da câmera e a salvava em disco. Como a operação de gravar um arquivo em disco rígido é bem mais lenta do que acessar esta mesma informação na memória RAM do computador, foi utilizado o programa AR RAM Disk v1.0.0.1, da AR Soft, para se criar um “*drive virtual*” na memória RAM do computador. Assim, as imagens parciais puderam ser salvas nele. A partir daí, o Matlab acessava-as tornando a operação de leitura totalmente realizada via memória principal (que, em geral, é de 100 a 800 vezes mais rápida que o disco rígido, dependendo da configuração do hardware).

O primeiro processamento realizado no software desenvolvido no Matlab foi a eliminação de 5 colunas de pixels, nas partes esquerda e direita da imagem, bem como 3 linhas de pixels nas partes superior e inferior. Esta retirada foi realizada porque as imagens obtidas, nas suas extremidades, apresentavam pixels avermelhados que representavam ruídos. Para não deixar de adquirir parte da tábua, o sistema foi programado para permitir pequenas sobreposições entre imagens (aproximadamente 6 linhas de pixel, 3 na parte superior e 3 na inferior), durante o processo de aquisição.

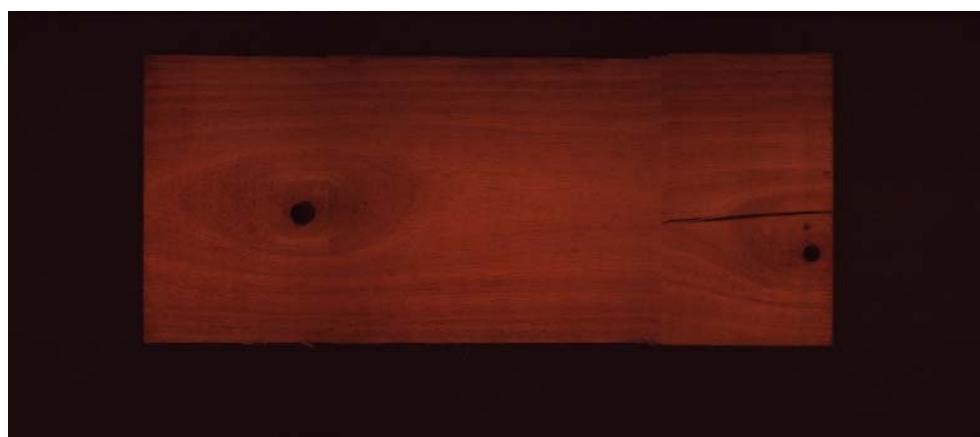
Em seguida foi feita a junção das imagens parciais para formação da imagem final da tábua. Depois foi retirada a parte da esteira através do uso do limiar de soma de pixels para separação entre esteira e tábua. A imagem resultante foi ainda cortada de modo a ter seu número de pixels ajustado, tanto na largura quanto no comprimento, para um valor múltiplo de 64, que foi o tamanho de bloco adotado. Obteve-se então a imagem a ser submetida ao processo de identificação e quantificação de defeitos para posterior classificação.

A Figura 4 exemplifica as etapas para a obtenção da imagem da tábua: imagens parciais adquiridas (Figura 4a), a junção das mesmas para formação

da imagem inicial da tábua com esteira (Figura 4b) e por fim a imagem final da tábua, após a retirada da esteira, que foi submetida ao processo de classificação (Figura 4c).



(a)



(b)



(c)

Figura 4 – Imagens parciais obtidas de uma tábua (a), junção das imagens parciais para formação da imagem da tábua (b) e imagem da tábua após a retirada da esteira (c).

A separação inicial entre defeitos e madeira limpa foi feita através de limiarização (YANG, 2001). O comando utilizado foi “im2bw” do Matlab, que inicialmente transformava a imagem RGB de entrada em uma imagem de 256 níveis de cinza. A seguir limiarizava, ou seja, percorria o histograma da dada imagem para encontrar o valor de pixel (limiar) que separava a imagem em duas categorias (“madeira limpa” e “defeitos”). Como resultado retornava então uma imagem binária<sup>1</sup> com os defeitos representados por pixels pretos e a madeira limpa representada por pixels brancos. O limiar foi previamente escolhido por análise visual da imagem binária. Todas as 11 tábuas foram processadas com o limiar variando de 0,15 até 0,35. Valores abaixo de 0,15 praticamente não realçavam os defeitos pois eliminava-os em sua maioria. Valores acima de 0,35 retornavam muita madeira limpa como defeito. Entre estes valores, o que melhor discriminava os defeitos em relação à madeira limpa foi o valor 0,22.

Portanto, a tarefa de localização dos prováveis defeitos foi realizada da seguinte maneira:

- a) limiarização da imagem da tábua, tornando pretos os prováveis defeitos e branco a madeira limpa;
- b) filtragem da imagem, em 2 etapas:
  - uma operação de “*opening*”<sup>2</sup>, ou seja, uma “*erosão*”<sup>3</sup> para eliminação de conjuntos pequenos de pixels (que não representam defeitos na imagem original e portanto poderiam causar falso positivo) seguida de uma “*dilatação*”<sup>4</sup> para que as informações relevantes, perdidas durante a “*erosão*” sobre os defeitos verdadeiros, fossem restauradas;
  - uma filtragem “*medfilt2*”<sup>5</sup> para eliminação de pequenos conjuntos

---

1 Numa imagem binária, todos os pixels assumem apenas um de dois valores: 0 ou 1.

2 Operação morfológica matemática que suaviza o contorno dos objetos da imagem, quebra ligações estreitas entre objetos e elimina ruídos menores que o elemento estruturante.

3 Operação morfológica que reduz as dimensões dos objetos da imagem, elimina ruídos menores que o elemento estruturante e quebra ligações estreitas entre objetos próximos.

4 Operação morfológica que aumenta as dimensões dos objetos da imagem e preenche pequenos orifícios destes objetos.

5 Remove ruídos tipo “*chuveiro*”. Cada pixel de saída contém o valor mediano de sua vizinhança 5x5 da imagem de entrada.

de pixels restantes que poderiam também causar falsos positivos.

- c) localização dos pixels pretos na imagem filtrada percorrendo-se linha por linha da mesma verificando onde a soma da linha retornava valor maior que zero (o que indica a presença de um ou mais pixels com valor 1). A partir da localização do primeiro pixel preto era retirado da imagem original o bloco correspondente contendo aquele pixel;
- d) este bloco, com probabilidade maior de ser defeito e não madeira limpa, foi então submetido ao classificador estatístico para determinação de seu agrupamento, podendo o mesmo ter sido ainda madeira limpa, mas já com menor possibilidade.

Este processo permitiu que o classificador processasse apenas uma pequena quantidade de blocos da tábua (menos de 10%) já que madeira limpa representava a maior parte de todas as tábuas.

Depois de localizado o primeiro pixel preto da imagem binária (item C acima) este representava o primeiro pixel do próximo bloco contendo um defeito que seria classificado. A seguir, testava-se se o defeito contido no bloco ocupava toda sua largura ou altura, ou seja, se havia uma coluna ou uma linha do bloco completamente preenchida por pixels pretos. Se sim, o processamento seguia para o item D. Mas caso contrário, era feito uma centralização do defeito no bloco antes que fosse retirado da imagem original o bloco correspondente contendo aquele pixel. Esta centralização se mostrou necessária porque na geração de blocos para o treinamento do classificador estatístico, quando o defeito era menor que um bloco o mesmo era cortado dentro de um bloco de 64 x 64 pixels de modo que ocupasse a posição central do mesmo.

### ***3.6 Reclassificação baseada em características morfológicas e informações de posição***

A reclassificação visou corrigir os erros cometidos pelo classificador

bayesiano na classificação dos blocos. Após determinado o agrupamento de defeitos a que um bloco pertencia eram verificadas outras propriedades específicas que cada agrupamento possuía, com o objetivo de confirmar sua classificação.

Todos os blocos classificados como pertencentes aos agrupamentos 2, 3, 4 e 6 foram submetidos a reclassificação com base na posição ou forma do defeito.

### ***3.7 Características morfológicas e informações de posição***

As características morfológicas foram retiradas da imagem binária dos blocos oriunda da classificação com base nas características do histograma.

Foram determinadas duas características morfológicas de cada tipo de defeito: eixo maior e eixo menor. Foram identificadas a coluna e a linha do bloco com maior seqüência de pixels considerada como defeito. O eixo maior é o maior entre os dois valores e o eixo menor, conseqüentemente, o outro valor. Estas características foram usadas na distinção entre os defeitos que as características do histograma apresentaram maiores porcentagens de confusão.

As características de posição foram consideradas em relação a imagem inteira da tábua. São elas: distância em relação a primeira linha da tábua; distância em relação a última linha da tábua; distância em relação a primeira coluna da tábua; distância em relação a última coluna da tábua. Todas as 4 medidas foram feitas em pixels. Estas características também foram usadas na distinção entre os defeitos que apresentaram maiores porcentagens de confusão.

As reclassificações implementadas foram:

- Defeitos do agrupamento 2 localizados nas extremidades superior ou inferior da tábua tiveram sua classificação alterada para agrupamento 6;

- Defeitos do agrupamento 2 localizados nas extremidades direita ou esquerda da tábua tiveram sua classificação alterada para agrupamento 3;
- Defeitos do agrupamento 3 localizados na região central da tábua tiveram sua classificação alterada para agrupamento 2 se estivessem próximos das extremidades esquerda ou direita da tábua;
- Defeitos do agrupamento 3 localizados na região central da tábua tiveram sua classificação alterada para agrupamento 6 se estivessem localizados próximos das extremidades superior ou inferior da tábua;
- Defeitos do agrupamento 4 localizados nas extremidades superior ou inferior da tábua tiveram sua classificação alterada para agrupamento 6;
- Defeitos do agrupamento 6 que ocupavam mais em largura de um bloco do que em altura tiveram sua classificação alterada para agrupamento 2;
- Defeitos do agrupamento 2 que ocupavam menos da metade da largura de um bloco mas ocupavam toda extensão de sua altura tiveram sua classificação alterada para agrupamento 6.

### ***3.8 Avaliação do resultado da classificação de blocos e das tábuas***

Os blocos para teste do classificador estatístico foram os restantes não utilizados na fase de desenvolvimento do mesmo: 9.333 blocos sendo 7781 do agrupamento 1, 368 do agrupamento 2, 697 do agrupamento 3, 175 do agrupamento 4, 34 do agrupamento 5 e 278 do agrupamento 6.

Os resultados da classificação, tanto para blocos quanto tábuas, foram portanto, organizados em uma matriz de confusão, na qual esses resultados foram colocados nas colunas. Na célula pertencente a diagonal principal de cada coluna estavam as identificações corretas para o agrupamento

representado por aquela coluna. No restante de cada uma das colunas estava o número de observações incorretamente classificadas como pertencentes a outros agrupamentos (erro de omissão).

Desta matriz foi estimado o coeficiente Kappa pela seguinte equação:

$$K = \frac{n \sum_{i=1}^j x_{ii} - \sum_{i=1}^j x_{iT} x_{Ti}}{n^2 - \sum_{i=1}^j x_{iT} x_{Ti}} \quad (10)$$

*em que:* K = estimativa do coeficiente Kappa;

$x_{ii}$  = valor na linha i e coluna i da matriz de confusão;

$x_{iT}$  = total da linha i;

$x_{Ti}$  = total da coluna i;

n = número total de amostras;

j = número total de agrupamentos.

### **3.9 Análise das imagens e classificação das tábuas**

Após o processo de reclassificação obteve-se então a quantificação final dos defeitos encontrados em cada uma das faces. Uma vez de posse destes dados foi possível fazer a classificação final da tábua utilizando-se as normas de classificação.

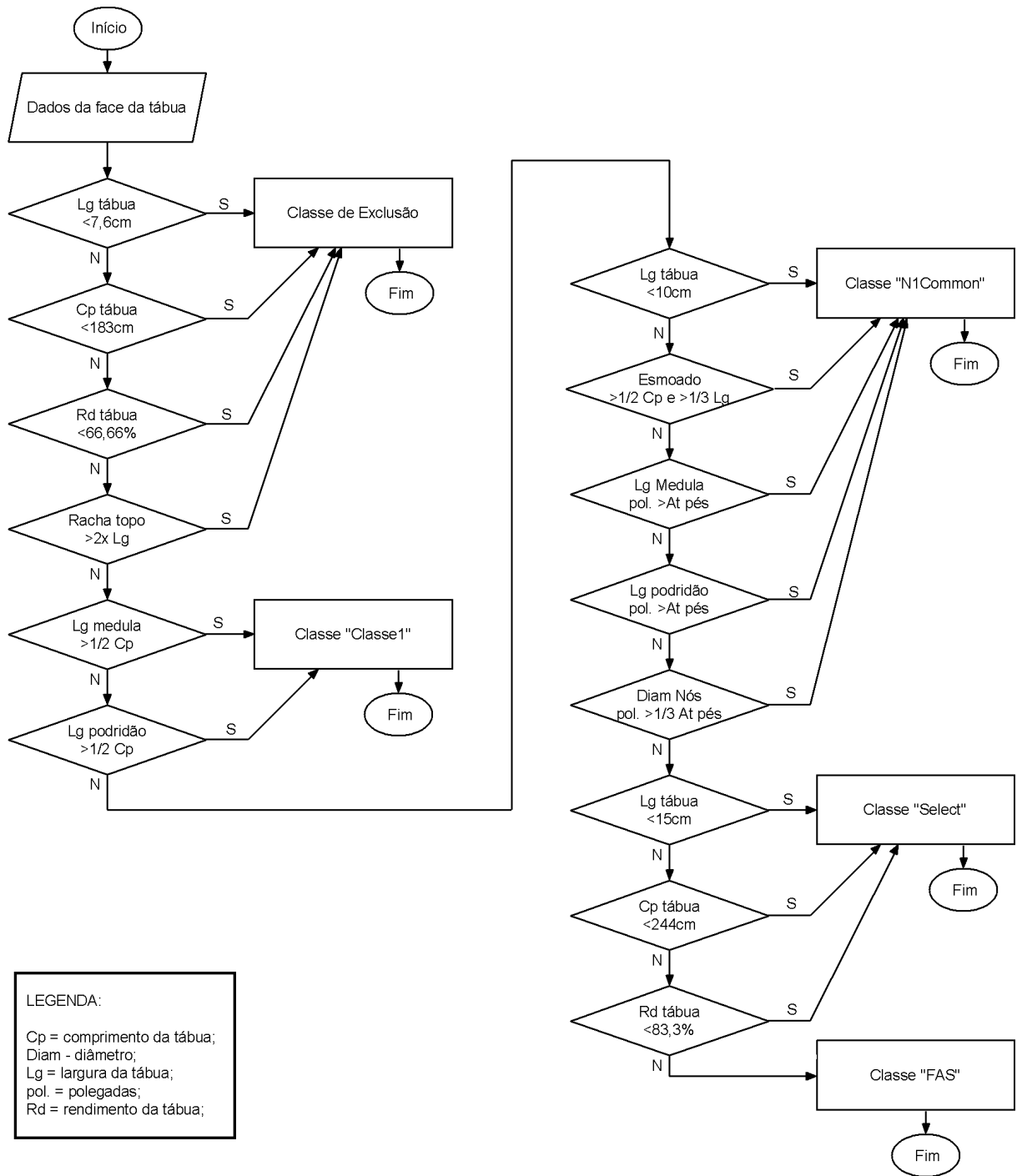
As tábuas foram classificadas conforme critérios estabelecidos pelas normas que consideram não só o tipo de defeito, mas também as suas dimensões e freqüência de ocorrência (ABNT, 2002B).

No presente trabalho, dois algoritmos de classificação de cada face da tábua foram implementados e testados: um com base nos critérios da norma NBR 14.806 (ABNT, 2002B) e outro nos critérios de uma norma comercial.

As regras de decisões de cada norma foram representadas na forma de um “fluxograma de decisão” conforme as Figuras 5 e 6.

A classificação da tábua foi igual a classificação da sua pior face. Porém, no caso da norma da ABNT, quando uma face foi classificada como pertencente a classe 1 ou 4 existem parâmetros referentes a face reversa que deviam ser levados em consideração e que podiam conduzir a classificação da tábua à classe de qualidade imediatamente inferior (2 ou 5, respectivamente), como mostrado na Figura 7.

Os resultados obtidos na classificação das tábuas foram comparados com as classificações feitas por especialistas. A mesma técnica, de “matriz de confusão” utilizada para blocos, foi utilizada aqui, para avaliação de desempenho do protótipo. O erro global foi usado para comparar as classificações usando as duas normas.



**Figura 5 - Fluxograma de classificação de uma face da tábua. (Norma comercial).**

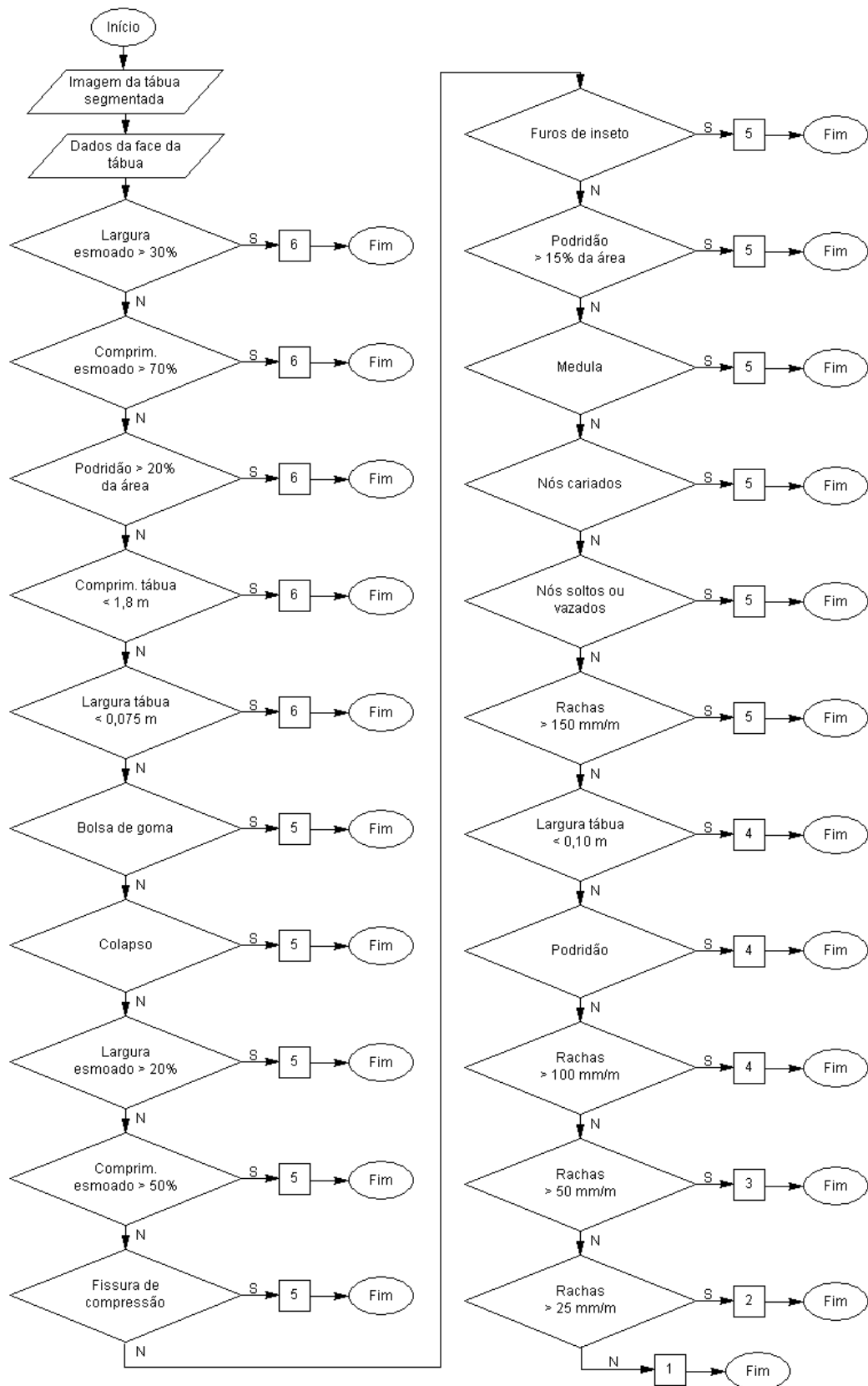
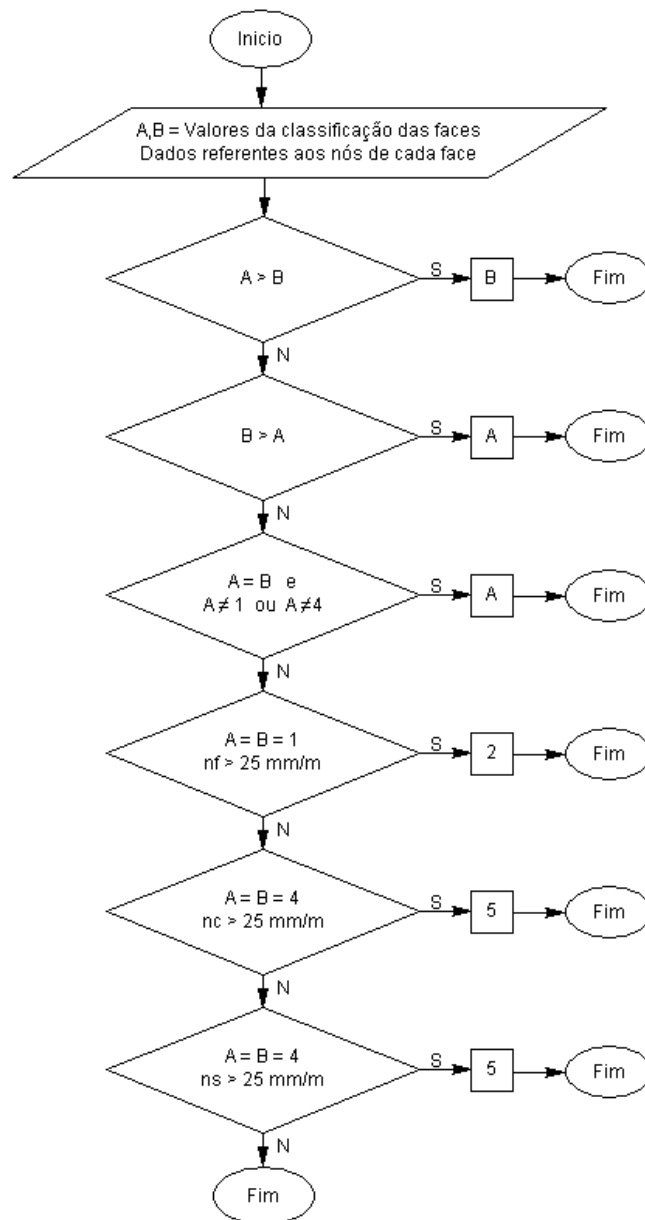


Figura 6 - Fluxograma de classificação de uma face da tábua (Norma ABNT).  
 Fonte: KHOURY JUNIOR, (2004).



**Figura 7 – Fluxograma de classificação da tábua:**  
*A, B = classificação das faces (1 a 6); nf = somatório nós firmes;  
 nc = somatório nós cariados; ns = somatório nós soltos.*  
 Fonte: KHOURY JUNIOR, (2004).

### **3.10 Avaliação do desempenho do protótipo**

O algoritmo de segmentação e reconhecimento dos objetos foi implementado no software do protótipo, juntamente com o algoritmo do classificador, para serem usados no teste de desempenho do protótipo.

Uma vez ajustada a velocidade da esteira do protótipo por intermédio da regulagem do inversor de frequência, esta ficava em velocidade constante e o protótipo aguardava a passagem de uma tábua.

Uma amostra de 42 tábuas, não usadas em nenhuma das etapas anteriores, foi submetida ao protótipo, para posterior comparação de resultados com a finalidade de se obter, em porcentagem, a exatidão global do sistema, através da matriz de confusão para ambas as normas, com e sem o uso de reclassificação. A partir dessa matriz foi possível observar a eficiência na solução do problema de confusão entre defeitos.

## 4 RESULTADOS E DISCUSSÃO

### 4.1 Interface do usuário

Uma interface de usuário foi desenvolvida para integrar o protótipo. Esta interface foi estruturada de modo a apresentar ao operador o mínimo possível de janelas. As janelas seguiam a seqüência lógica da atividade de operação dos classificadores humanos na serraria objetivando reproduzir o fluxo natural do trabalho. Nesta interface a primeira tela apresentada foi a de seleção do modo de classificação (Figura 8):

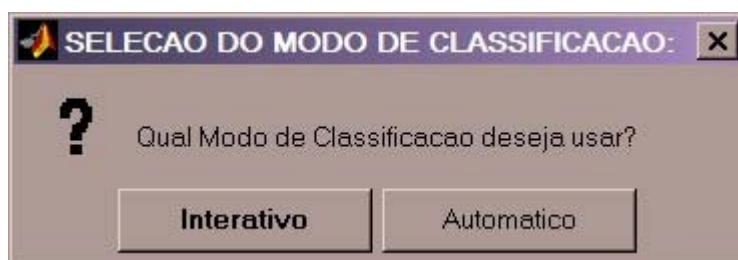


Figura 8 – Caixa de diálogo de escolha do modo de operação.

Uma vez escolhido o modo de operação, o usuário tinha que escolher qual de classificação desejava usar (Figura 9).

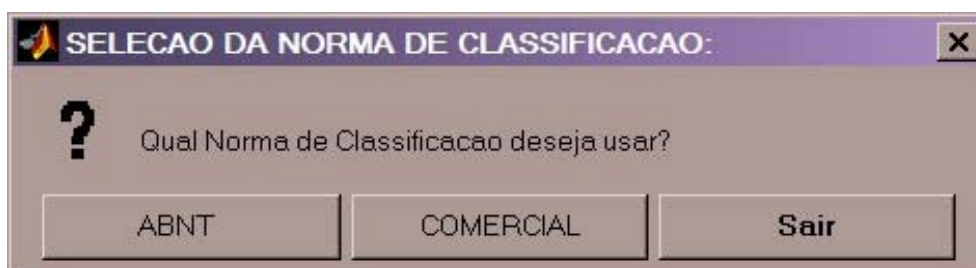


Figura 9 – Caixa de diálogo para seleção da de classificação.

Após esta escolha, o sistema entrava em estado de espera (Figura 10).

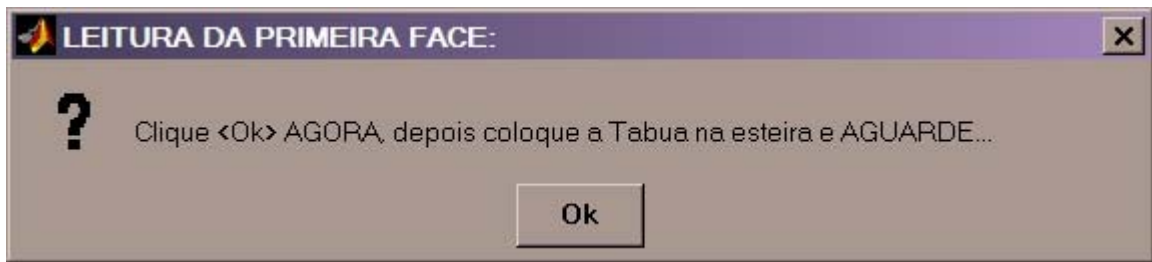


Figura 10 – Caixa de diálogo informando o estado de espera por tábua.

O limiar de soma de pixels de cada coluna da imagem começava a ser usado então para detectar o início e o fim da tábua passando pela esteira. Quando não havia tábua na esteira, a soma dos pixels das colunas da imagem resultava um valor pequeno pois apenas a imagem da esteira preta estava sendo captada. Quando uma tábua começava a passar, essa mesma soma aumentava de valor, pois parte do preto da esteira era substituído pela imagem da tábua, que era mais clara, indicando assim que havia tábua na esteira. Esse mesmo cálculo de limiar foi também usado, com raciocínio inverso, para determinar quando uma tábua terminava de passar e portanto encerrava-se o processo de captura de imagens para aquela face de tábua.

Após a passagem da primeira face da tábua, o usuário via a imagem montada da mesma, e era solicitado ao operador que virasse a tábua na esteira para que fosse adquirida a imagem da outra face (Figura 11).



Figura 11 – Caixa de diálogo com o resultado da leitura parcial da tábua.

Após montadas as duas imagens dos dois lados da mesma tábua, era iniciado o processo de separação do “fundo” da “imagem da tábua”. Ao fim deste processo restou apenas a imagem da tábua que era submetida a segmentação, detecção dos defeitos e classificação pelos algoritmos desenvolvidos no Matlab. O resultado final da classificação aparecia na tela do microcomputador (Figura 12).



Figura 12 – Caixa de diálogo com o resultado da classificação final da tábua.

#### 4.2 Parâmetros do classificador estatístico

Um dos parâmetros que o classificador estatístico usou foi a “inversa da matriz de covariância comum”. Ao se tentar calcular a inversa desta matriz com o comando “inv” do Matlab, obteve-se como resposta que a matriz era mal condicionada. Uma matriz mal condicionada é aquela em que o determinante é um número tendendo a zero. Segundo SEARLE (1982), uma maneira de resolver este problema é calculando-se a matriz que mais se aproximaria de uma possível inversa, ou seja, não possui as propriedades matemáticas das matrizes inversas, mas possui quatro outras propriedades. Dada uma matriz A, a inversa M de Moore-Penrose desta matriz é tal que:

- 1)  $A * M * A = A$ ;
- 2)  $M * A * M = M$ ;
- 3)  $A * M$  é simétrica;
- 4)  $M * A$  é simétrica.

Portanto, o comando “pinv” do Matlab foi utilizado para o cálculo da inversa da matriz de covariância, ao invés do comando “inv”. Esta matriz é chamada de inversa generalizada (SEARLE, 1982), pseudo-inversa (SAS, 1999) ou inversa de Moore-Penrose (MATLAB, 2000).

Outros parâmetros utilizados pelo classificador são as probabilidades a *priori* de cada agrupamento ( $P(C_j)$ ). Estes valores foram usados no cálculo dos valores das funções discriminantes para cada um dos agrupamentos, dado um vetor  $v$  de entrada. As probabilidades a *priori* foram obtidas após a retirada de todos os blocos de defeito e madeira limpa das 11 tábuas e realização dos respectivos agrupamentos. Os valores de ( $P(C_j)$ ) determinados foram: 95,00% para agrupamento 1; 1,06% para agrupamento 2; 2,36% para agrupamento 3; 0,39% para agrupamento 4; 0,30% para agrupamento 5; e 0,89% para agrupamento 6.

### **4.3 Eliminação de variáveis originais por Análise de Componentes Principais**

A matriz de dados das variáveis originais foi utilizada para condução da Análise de Componentes Principais (ACP). Esta matriz, com 45 colunas (variáveis originais, ou características) e 9333 linhas (observações), teve suas variáveis padronizadas e depois feita a ACP usando o MatLab<sup>6</sup>. A padronização foi necessária devido a grande diferença na escala de variação das características.

As características 1 a 12 foram os percentis da banda vermelho; as características 13 a 24 foram os percentis da banda verde; as características

---

<sup>6</sup> Comando “princomp”, que retorna os autovetores e os autovalores da matriz de covariância.

25 a 36 foram os percentis da banda azul. As características 37 a 45 foram, respectivamente a média, mediana e desvio padrão dos valores de pixel para cada uma das bandas.

A ACP retornou a seguinte ordem crescente de importância para as características a serem descartadas: 24, 12, 25, 19, 11, 3, 6, 4, 10, 31, 1, 22, 23, 14, 21, 42, 30, 32, 33, 26, 34, 27, 28, 41, 44, 45, 37, 38, 35. As características 2, 5, 7, 8, 9, 13, 15, 16, 17, 18, 20, 29, 36, 39, 40 e 43 não figuraram como tendo maior correlação com nenhum componente principal de menor autovalor, portanto foram as mais importantes para discriminação entre os agrupamentos.

A seguir, foi realizada uma análise no descarte de características proposto pela ACP. Cada característica proposta no descarte foi eliminada isoladamente, os parâmetros do classificador estatístico foram recalculados considerando-se as características restantes e a classificação das tábuas foi novamente realizada utilizando-se a norma comercial.

As características 24, 12 e 25 foram eliminadas sem que houvesse diminuição do acerto global (80,95%) obtido usando-se as 45 características originais, ou seja, não houve alteração na matriz de confusão do resultado da classificação das tábuas. Mas, após a eliminação da característica 19 o acerto global caiu para 78,57%, permanecendo assim também após a eliminação da característica 11. A seguir, pôde-se eliminar as características 3, 6, 4, 10, 31, 1, 22, 23, 14, 21 e 42 com o acerto global mantendo-se em 76,19%. Por fim o acerto global caiu para 73,81% eliminando-se as características 30, 32, 33, 26, 34, 27, 28, 41, 44, 45, 37, 38, 35, 29 e 36.

Observou-se que ao optar pela eliminação de todas as características ditas menos importantes pela ACP (29 no total), o acerto global cairia para 73,81% e restariam apenas três características da banda azul (características de número 25, 36 e 43). No entanto, as bandas verde e vermelha estariam representadas em sete e seis características, respectivamente. Portanto a banda azul é a que carregava menos variância dentre as três sendo a menos importante para o processo de classificação.

Observou-se também que as características média, mediana e desvio padrão foram importantes. Numa análise menos atenta poderia se pensar que estas características tinham seus valores representados nos percentis de valores de pixel, o que não foi verdade. Para todas as bandas, no mínimo uma delas mostrou-se necessária mesmo se houvesse eliminação das 29 características sugeridas pela ACP.

Neste trabalho, optou-se por eliminar apenas as três características (24, 12 e 25) que não afetaram o acerto global. Esta escolha se deu porque apesar da perda, em porcentagem, no acerto global ter sido pequena eliminado-se também as características 19 e 11 (foi de 80,95% para 78,57%), não houve ganho no desempenho do sistema em termos de velocidade de processamento.

Estes resultados mostraram que as tarefas que deveriam ser otimizadas são a localização inicial dos defeitos nas imagens das tábuas, que é a mais dispendiosa, e a atividade de limiarização/filtragem, que é a segunda mais dispendiosa. Portanto, nas análises posteriores foram utilizadas 42 das 45 características iniciais.

#### ***4.4 Resultado da classificação de blocos de imagens***

No Quadro 1 é apresentada a matriz de confusão gerada para os blocos de imagens usados no teste final do classificador estatístico.

**Quadro 1** – Matriz de confusão para blocos de imagens, por agrupamento de defeitos, da amostra de teste<sup>7</sup>.

<i>Agrupamentos</i>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<i>Total</i>	<i>Erro Inclusão(%)</i>
<b>1</b>	<b>7768</b>	4	87	2	13	3	7877	1,38%
<b>2</b>	0	<b>182</b>	33	3	0	33	251	27,49%
<b>3</b>	1	45	<b>563</b>	0	0	15	624	9,78%
<b>4</b>	8	57	0	<b>147</b>	6	45	263	44,11%
<b>5</b>	3	5	1	17	<b>11</b>	4	41	73,17%
<b>6</b>	1	75	13	6	4	<b>178</b>	277	35,74%
<i>Total</i>	<b>7781</b>	<b>368</b>	<b>697</b>	<b>175</b>	<b>34</b>	<b>278</b>	<b>Total</b>	<b>Acerto Global</b>
<i>Erro Omissão(%)</i>	<b>0,17%</b>	<b>50,54%</b>	<b>19,23%</b>	<b>16,00%</b>	<b>67,65%</b>	<b>35,97%</b>	<b>9333</b>	<b>94,81%</b>

Pode-se observar também que o agrupamento que teve o pior desempenho na classificação foi o de número 5. O fator que contribuiu para que o agrupamento tivesse alto erro de omissão (67,65%) foi a pequena ocorrência destes defeitos nas tábuas (poucas amostras disponíveis para o desenvolvimento do classificador) o que causou dificuldade ao classificador em reconhecer tais blocos. O alto valor do erro de inclusão para este mesmo agrupamento (73,17%) se deveu a dois fatores: primeiro, ele possuía poucas instâncias amostradas, portanto poucas inclusões neste agrupamento aumentavam significativamente o erro de inclusão; segundo, este agrupamento compreendeu o defeito nó sadio que foi confundido com madeira limpa (problema este também encontrado por KHOURY JUNIOR (2004)), então muitas instâncias do agrupamento 1 foram inclusas neste agrupamento aumentando assim este erro. Como algumas normas comerciais de classificação permitem esse defeito em todas as suas classes de qualidade este erro só influenciou a classificação via norma oficial, com efeito pouco significativo pois só afetou as tábuas da classe 1.

Entre os agrupamentos que representam defeitos (2 a 5), o melhor resultado observado foi para o agrupamento 4 que teve um erro de omissão de 16%. Mas, teve um erro de inclusão significativo: 44,11%. Este valor se deveu ao fato de algumas partes dos defeitos *bolsa de goma* e *podridão* (do

<sup>7</sup> **Agrupamento 1** – Madeiras Limpas, **Agrupamento 2** – bolsa de goma, nó cariado, nó solto encaixado, nó solto oco e podridão, **Agrupamento 3** – Esmoados, **Agrupamento 4** – Fendilhados, **Agrupamento 5** – Nó sadio e nó sadio corrompido, **Agrupamento 6** – Rachas.

agrupamento 2), e *racha* (do agrupamento 6) ao serem transformados em blocos, se assemelharem a um *fendilhado* o que tornou difícil a discriminação entre eles. Como tanto *fendilhado* como *bolsa de goma* e *podridão* podem ocorrer em qualquer parte de uma tábua, as regras de posição usadas na reclassificação, durante a classificação de tábuas, não podem oferecer possibilidade de diferenciação entre os mesmos.

O agrupamento 2 teve o segundo maior erro de inclusão (50,54%). Ele recebeu muitas instâncias do agrupamento 3 que inclui o defeito *esmoado*. Este defeito, quando ocupa todo o bloco, em alguns casos, possui características semelhantes a *bolsa de goma* (do agrupamento 2) tornando-os pouco diferenciáveis. Como tanto *esmoado* como *bolsa de goma* podem ocorrer nas extremidades de uma tábua, as regras de posição usadas na reclassificação não teriam possibilidade de contribuir para diferenciação entre os mesmos.

Um baixo erro de inclusão nos agrupamentos 1 e 2 foi importante para o processo de classificação de tábuas pois muitas inclusões nestes agrupamentos causariam muitas classificações em classes abaixo das suas verdadeiras classes de qualidade. Um alto erro de inclusão no agrupamento 1 por exemplo (*madeira limpa*) causaria a classificação de muitas tábuas em classes de qualidade inferiores. Inclusões no agrupamento 2 causaria muitas classificações de tábuas em classes inferiores também pois nele está incluído o defeito *bolsa de goma*. A presença deste defeito por si só, segundo a norma oficial, já jogaria a classificação das tábuas para classes mais inferiores.

Em trabalho semelhante, KHOURY JUNIOR (2004) obteve Kappa de 0,77 e acerto global de 81%. O acerto global na classificação de blocos neste trabalho foi de 94,81% com a estimativa do coeficiente Kappa de 0,82, valor este considerado eficiente segundo a escala de LANDIS e KOCH<sup>8</sup> (1977) para este coeficiente.

---

<sup>8</sup> A referida escala associa critérios qualitativos de aceitação com intervalos de valores de Kappa em problemas da área médica. Ela está sendo usada aqui pela falta de uma escala semelhante, mas para uso na classificação de madeira.

#### 4.5 Classificação de tábuas

O número total de blocos processados para a classificação das 42 tábuas foi de 31672. Destes, 2929 foram considerados defeitos. Dos 2929, 118 eram agrupamento 5, portanto não sofreram tentativa de reclassificação. Apenas os blocos considerados defeitos e pertencentes aos agrupamentos 2, 3, 4 e 6 foram submetidos ao processo de reclassificação. Um total de 1270 tiveram sua classificação inicial alterada.

Uma parte dos blocos reclassificados, aproximadamente 71%, implicou em correção, ou seja, haviam recebido classificação errada, e o problema foi detectado e ajustado. Mas, outra parte dos reclassificados, 29%, houve o problema inverso, haviam recebido classificação correta e sofreram alteração para uma classificação errada. Observou-se então que o número de alterações corretas foi maior que o de alterações para classificações incorretas justificando assim o uso das regras de reclassificação.

A separação inicial entre defeitos e *madeira limpa* foi feita através de limiarização e filtragem. Esta escolha se deu para que houvesse uma redução no número de falsos positivos, ou seja, este processo permitiu também que o classificador, em média, processasse apenas algo em torno de 10% dos blocos da tábua já que *madeira limpa* representava a maior parte de todas as tábuas. Houve ganho em tempo de processamento, pois reduziu-se o tempo destinado a manipulação de falsos positivos.

Observou-se ainda que, com alguns *esmoados* que percorrem toda a extensão de comprimento da tábua, este pode ser confundido com esteira pelo algoritmo. Quando isto acontece o *esmoado* vai sendo retirado da tábua até que se alcance uma situação de limiar aceitável. Porém, o *esmoado*, tanto em largura quanto em comprimento, afeta a classificação da tábua. Nestas situações ocorreram classificações incorretas.

A seguir são apresentadas as matrizes de confusão resultantes do teste final do protótipo utilizando 42 tábuas. A partir delas pode-se fazer a análise da classificação utilizando-se ou não a reclassificação bem como comparar a

classificação para ambas as normas.

#### **4.6 Resultados da classificação**

Devido a ausência de exemplares de tábuas para certas classes, não foi usado o coeficiente Kappa na avaliação das matrizes de confusão, nem na comparação entre as mesmas. A ausência de representantes para uma ou mais classes causa a presença de muitas células com valor zero na matriz, na coluna da referida classe. A presença destes valores nulos causa imprecisão no cálculo do coeficiente pois perde parte de sua capacidade de expressar a confusão entre as classes. Isto ocorre porque a ausência de elementos em uma classe faz que com a matriz de confusão não apresente os elementos daquela classe que foram incorretamente classificados como pertencentes a outras classes (erro de omissão). O erro de inclusão (elementos de outras classes classificados como sendo daquela classe) também fica comprometido pois este passa a ter o valor máximo sempre, para qualquer que seja o número de elementos inclusos na classe sem representantes. Portanto, a avaliação individual de cada matriz e a comparação entre elas foi feita utilizando-se o acerto global e os erros de omissão e de inclusão para as classes.

Nos Quadros 2 e 3, são apresentados os resultados de classificação para as tábuas utilizando-se a norma da ABNT, sem e com a reclassificação, respectivamente.

**Quadro 2** – Matriz de confusão para imagens das tábuas classificadas pela norma da ABNT, sem o uso da reclassificação dos blocos.

<i>Classes ABNT</i>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>Total</b>	<i>Erro Inclusão(%)</i>
<b>1</b>	<b>0</b>	0	0	0	0	0	<b>0</b>	<i>0,0%</i>
<b>2</b>	0	<b>8</b>	0	0	2	0	<b>10</b>	<i>20,0%</i>
<b>3</b>	0	0	<b>0</b>	0	2	0	<b>2</b>	<i>100,0%</i>
<b>4</b>	0	1	0	<b>3</b>	5	0	<b>9</b>	<i>66,7%</i>
<b>5</b>	0	1	0	2	<b>12</b>	0	<b>15</b>	<i>20,0%</i>
<b>6</b>	0	0	0	0	3	<b>3</b>	<b>6</b>	<i>50,0%</i>
<b>Total</b>	<b>0</b>	<b>10</b>	<b>0</b>	<b>5</b>	<b>24</b>	<b>3</b>	<b>Total</b>	<b><i>Acerto Global</i></b>
<i>Erro Omissão(%)</i>	<i>0,0%</i>	<i>20,0%</i>	<i>0,0%</i>	<i>40,0%</i>	<i>50,0%</i>	<i>0,0%</i>	<b>42</b>	<b><i>61,90%</i></b>

A classe 5 teve o maior erro de omissão (50%), sendo que parte das suas amostras foram atribuídas a classes superiores. Esta classificação foi devido a presença de *furo de inseto* em algumas delas. O *furo de inseto* é um defeito de dimensões muito pequenas que foi eliminado nas filtragens. Mas a presença dele em uma tábua, segundo a norma da ABNT, faz com que a mesma só possa ser classificada nas classes mais inferiores, 5 ou 6. Com a retirada do defeito na etapa de filtragem, as tábuas foram classificadas em classes superiores.

Algumas tábuas classificadas como classe 5, mas que eram classe 6, receberam esta designação devido ao tamanho do *esmoado* nelas presente. Este ocupava mais que 70% de sua extensão. Porém, durante a retirada da esteira da imagem da tábua consumia-se um pequeno pedaço deste devido a semelhança do mesmo, em sua região de fronteira, com a esteira. Restava então uma tábua cuja largura e comprimento, em pixels, eram valores não múltiplos de 64. Fazia-se depois a retirada de tantas colunas e linhas quanto fossem necessárias para que a imagem fosse transformada, na largura e comprimento, em um valor múltiplo de 64, que foi o tamanho do bloco adotado neste trabalho. Consumia-se assim mais uma pequena parte do *esmoado*. Deste modo o defeito sofria redução e a tábua subia de classificação.

O erro de inclusão da classe 3 foi grande (100%) porque não haviam amostras para esta classe, portanto, qualquer que fosse o valor da inclusão

nesta classe teria este mesmo efeito.

De maneira geral, o desempenho do classificador não foi tendencioso para atribuições a classes superiores apenas ou a classes inferiores apenas. Pode-se observar isto comparando-se o número de tábuas atribuídas a classes superiores (9, que é a soma das células acima da diagonal principal) com o número tábuas atribuídas a classes inferiores (7, que é a soma das células abaixo da diagonal principal). Em termos comerciais este comportamento tem um efeito prático de não causar prejuízo recorrente apenas para o comprador ou apenas para o vendedor com os erros de classificação.

**Quadro 3** – Matriz de confusão para imagens das tábuas classificadas pela norma da ABNT, com o uso da reclassificação dos blocos.

<i>Classes ABNT</i>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>Total</b>	<i>Erro Inclusão(%)</i>
<b>1</b>	<b>0</b>	0	0	0	0	0	<b>0</b>	0,0%
<b>2</b>	0	<b>9</b>	0	0	2	0	<b>11</b>	18,2%
<b>3</b>	0	0	<b>0</b>	1	5	1	<b>7</b>	100,0%
<b>4</b>	0	0	0	<b>3</b>	3	0	<b>6</b>	50,0%
<b>5</b>	0	1	0	1	<b>13</b>	0	<b>15</b>	13,3%
<b>6</b>	0	0	0	0	1	<b>2</b>	<b>3</b>	33,3%
<b>Total</b>	<b>0</b>	<b>10</b>	<b>0</b>	<b>5</b>	<b>24</b>	<b>3</b>	<b>Total</b>	<b>Acerto Global</b>
<i>Erro Omissão(%)</i>	0,0%	10,0%	0,0%	40,0%	45,8%	33,3%	<b>42</b>	<b>64,29%</b>

Comparando-se os Quadros 2 e 3, no que diz respeito ao acerto global, podemos ver que reclassificação melhorou a taxa de acerto: foi de 61,90% para 64,29%.

Houve uma tendência de classificação em classes superiores no processo de classificação. Pode-se observar isto comparando-se o número de tábuas atribuídas a classes superiores (12) com o número de atribuições a classes inferiores (3). Este comportamento tem o efeito prático de causar prejuízo recorrente para o comprador, em benefício do vendedor.

Nos Quadros 4 e 5 são apresentados os resultados de classificação para as tábuas utilizando-se a norma comercial, sem e com a reclassificação,

respectivamente.

**Quadro 4** – Matriz de confusão para imagens das tábuas classificadas pela norma comercial, sem o uso da reclassificação dos blocos.

<i>Classes ARACRUZ</i>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Total</b>	<i>Erro de Inclusão(%)</i>
<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<i>0,0%</i>
<b>2</b>	<b>0</b>	<b>17</b>	<b>0</b>	<b>5</b>	<b>0</b>	<b>22</b>	<i>22,7%</i>
<b>3</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>2</b>	<b>0</b>	<b>2</b>	<i>100,0%</i>
<b>4</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>10</b>	<b>2</b>	<b>13</b>	<i>23,1%</i>
<b>5</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>2</b>	<b>3</b>	<b>5</b>	<i>40,0%</i>
<b>Total</b>	<b>0</b>	<b>18</b>	<b>0</b>	<b>19</b>	<b>5</b>	<b>Total</b>	<i>Acerto Global</i>
<i>Erro de Omissão(%)</i>	<i>0,0%</i>	<i>5,6%</i>	<i>0,0%</i>	<i>47,4%</i>	<i>40,0%</i>	<b>42</b>	<i>71,43%</i>

No Quadro 4 pode-se observar o mesmo problema ocorrido no Quadro 2 que é causado pela amostra. O erro de inclusão da classe 3 foi grande (100%) porque não haviam amostras para esta classe.

Outro problema com a amostra pode ser visto aqui devido a ausência de tábuas classe 1. Não houve inclusão nesta classe pois todas as tábuas utilizadas eram menores que o comprimento mínimo (2,44 m) para se enquadrar nela pela norma comercial. Assim, o algoritmo de classificação descartava logo a entrada de alguma tábua nesta classe, sem nem mesmo se utilizar da quantificação dos defeitos nela existentes.

Esta norma de classificação aceita tábuas com *furo de inseto* em todas as suas classes, mas estes precisam ter diâmetro menor que 0,635 mm. Tábuas com *furo de inseto* acima desse diâmetro recebem classificação 5. Como estes foram eliminados na filtragem haviam tábuas classe 5 que foram classificadas em classes superiores.

O mesmo problema de classificação em classes superiores foi observado na norma da ABNT, com elementos da classe 6 indo para classe 5, aqui observa-se agora entre as classe 5 e 4, com as mesmas tábuas, por causa da retirada em excesso do *esmoado*.

Olhando-se o acerto global, o resultado da classificação foi considerado razoável, exceto pelo problema da tendência de classificação em classes

superiores que é considerada uma característica ruim para um classificador de tábuas, do ponto de vista do comprador.

**Quadro 5** – Matriz de confusão para imagens das tábuas classificadas pela norma comercial, com o uso da reclassificação dos blocos.

<i>Classes ARACRUZ</i>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Total</b>	<i>Erro de Inclusão(%)</i>
<b>1</b>	<b>0</b>	0	0	0	0	<b>0</b>	<i>0,0%</i>
<b>2</b>	0	<b>14</b>	0	1	1	<b>16</b>	<i>12,5%</i>
<b>3</b>	0	2	<b>0</b>	0	0	<b>2</b>	<i>100,0%</i>
<b>4</b>	0	2	0	<b>17</b>	1	<b>20</b>	<i>15,0%</i>
<b>5</b>	0	0	0	1	<b>3</b>	<b>4</b>	<i>25,0%</i>
<b>Total</b>	<b>0</b>	<b>18</b>	<b>0</b>	<b>19</b>	<b>5</b>	<b>Total</b>	<b><i>Acerto Global</i></b>
<i>Erro de Omissão(%)</i>	<i>0,0%</i>	<i>22,2%</i>	<i>0,0%</i>	<i>10,5%</i>	<i>40,0%</i>	<b>42</b>	<b><i>80,95%</i></b>

O índice de acerto (exatidão global) na classificação de blocos foi alto (94,81%), o que não se refletiu na classificação das tábuas (64,29% e 80,95% para as normas ABNT e comercial). Isto ocorre porque segundo a norma da ABNT se uma tábua possui *bolsa de goma* ela só pode ser classificada como sendo das classes 4, 5 ou 6. Como o número de tábuas utilizadas no teste do protótipo (42) foi bem menor que o número de blocos utilizados no teste do classificador (9333), um erro deste tipo foi representativo no processo de classificação de tábuas, mas o mesmo não pode ser dito no processo de classificação de blocos.

Na comparação entre os quadros 2 e 3, com os resultados de classificação pela norma da ABNT, podemos ver que a reclassificação fez o acerto global subir de 61,90% para 64,29%. Já comparando-se os quadros 4 e 5, que contêm os resultados de classificação pela norma comercial, podemos ver que a reclassificação fez o acerto global subir de 71,43% para 80,95%.

A reclassificação foi considerada útil, uma vez que teve influência no índice final de acerto de classificação das tábuas. Conclui-se assim a eficiência desta técnica no processo de classificação de madeira serrada de eucalipto.

Ainda comparando-se os Quadros 2 e 3 com os Quadros 4 e 5 observa-

se que a simplicidade da norma de classificação teve efeito na taxa de acerto de classificação. Em ambos os casos, ou seja, com e sem reclassificação, o classificador que utilizou-se da norma comercial obteve melhores resultados que o classificador que utilizou a norma da ABNT.

#### **4.7 O comportamento funcional do protótipo**

O protótipo apresentou vibração na esteira quando em velocidades maiores que  $0,61 \text{ m s}^{-1}$ . Em velocidades mais elevadas, o contato da esteira com as laterais do protótipo fez com que a vibração se tornasse um fator limitante. Com relação a câmera, esta velocidade poderia aumentar, bastando ajustar o tempo de leitura do CCD (*shutter speed*).

O protótipo utilizou uma “*AreaScan Camera*”. Devido a problemas de sincronismo causados pelo acesso simultâneo ao disco rígido do microcomputador por alguns serviços do Sistema Operacional, durante o processo de aquisição das imagens, houve sobreposição de pequenas partes de algumas delas, além do desejado, na montagem de partes das tábuas. Já em outras partes da imagem da mesma tábua não houve sobreposição. O sistema foi montado assim porque foi preferível que houvesse pequenas repetições em certas partes da imagem do que faltar partes da tábua na imagem. A câmera adequada para tarefas do tipo “*inspeção em esteiras rolantes*” é a “*LineScan Camera*” (HAJIMOWLANA et al., 1999).

Observou-se, durante o processo de classificação, que o alinhamento das tábuas na esteira tem efeito sobre o resultado da classificação. Isto ocorre porque durante a retirada da esteira das imagens obtidas, um pequeno pedaço de tábua é retirado junto devido a deformidades da tábua. Quanto mais alinhada a imagem da tábua estiver menos tábua se perde. O desalinhamento ocorre devido a falta de um mecanismo para endireitar as tábuas antes da sua passagem sob a câmera. Pode-se fazer o alinhamento da imagem via software, mas esta é uma tarefa exigente computacionalmente devido as dimensões das imagens. Como o processo de classificação também

tem um alto custo computacional, a produtividade do sistema ficaria comprometida. Por isso a sugestão é que o processo seja mecânico e não via software. Neste trabalho, o problema foi resolvido no momento da passagem das tábuas. Estas já foram introduzidas manualmente para que ficassem bem alinhadas na esteira.

Desconsiderando o tempo para se virar a tábua e considerando uma tábua média como sendo de 0,03x0,12x1,83 metros de altura, largura e comprimento, respectivamente, e a velocidade de operação de 0,61 m s<sup>-1</sup>, o protótipo teve uma produtividade aproximada de 7,9 m<sup>3</sup> h<sup>-1</sup>.

## 5 CONCLUSÕES

O desempenho do classificador bayesiano mostrou-se eficiente na classificação de blocos de imagens de madeira serrada, segundo a escala de LANDIS e KOCH para o coeficiente Kappa. O índice Kappa foi de 0,82 com o acerto global de 94,81%.

A reclassificação de blocos com base em características morfológicas e de posição trouxe resultados positivos ao processo de classificação de tábuas. Para a norma da ABNT, o resultado na classificação das tábuas sem o uso de reclassificação foi de 61,90% e com o uso de reclassificação foi de 64,29%, representando um ganho de 3,86% no índice de acerto. Para a norma comercial o resultado da classificação das tábuas sem o uso da reclassificação foi de 71,43% e com o uso da reclassificação foi de 80,95%, proporcionando um ganho de 13,33% de acerto na classificação.

Normas menos complexas que a da ABNT, como as normas comerciais criadas por serrarias, facilitaram o processo de classificação e, portanto, melhoraram a taxa de acerto de classificação.

O protótipo desenvolvido teve uma produtividade considerada satisfatória ( $7,9 \text{ m}^3 \text{ h}^{-1}$ ) para ser utilizado na linha de produção de uma serraria comercial.

## 6 SUGESTÕES PARA TRABALHOS FUTUROS

O protótipo deve receber um dispositivo mecânico para alinhar as tábuas em relação a câmera para diminuir o tempo de pré-processamento da imagem antes da classificação.

O protótipo deve receber uma segunda câmera, da mesma marca e modelo da primeira, colocada embaixo do mesmo, assim as duas faces da tábua seriam adquiridas simultaneamente.

O uso de tamanho variável de bloco deve ser testado para evitar a retirada (parcial ou total) de defeitos que afetam a classificação das tábuas.

Usar Análise de Componentes Principais para a determinação de quais dos 100 possíveis percentis são os que carregam maior capacidade de discriminação para cada classe de defeito.

Avaliar a utilização de uma matriz de covariância para cada classe ao invés de uma matriz de covariância comum, o que pode melhorar a discriminação entre as classes de defeitos.

Testar o sistema com um maior número de tábuas para se obter observações de todas as classes. Assim, poder-se-ia utilizar um teste estatístico para se obter o contraste entre os coeficientes Kappa na comparação entre as matrizes de confusão.

Avaliar a utilização de limiarização automática, ao invés de limiarização manual, para comparação de qual método proporciona melhor resultado.

Testar a utilização de informações de textura no processo de reclassificação no intuito de melhorar a distinção entre os defeitos confundíveis pelo classificador estatístico.

## 7 REFERÊNCIAS BIBLIOGRÁFICAS

ALVES, R.M. **Caracterização genética de populações de cupuaçuzeiro, Theobroma grandiflorum (Willd. ex. Spreng.) Schum., por marcadores microssatélites e características botânico-agronômicos.** 2002. 146p. Tese (Doutorado em Agronomia) - Escola Superior de Agricultura Luiz de Queiroz, Piracicaba.

ARACRUZ PRODUTOS DE MADEIRA S.A. Descrição do produto – Madeira Serrada de Eucalipto. Disponível em: <<http://www.lyptus.com.br>>. Acesso em: 09 ago. 2004.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS – ABNT. **NBR 12.551:** Madeira serrada – Terminologia. Rio de Janeiro-RJ, 2002A.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS – ABNT. **NBR 14.806:** Madeira serrada de eucalipto – Requisitos. Rio de Janeiro-RJ, 2002B.

BHANDARKAR, S.M.; FAUST, T.D.; TANG, M. Design and prototype development of a computer vision-based lumber production planning system. **Image and vision computing**, n.20, p.167-189, 2002.

CARREIRA, M.R. **Crítérios para classificação visual de peças estruturais de Pinus Sp.** 2003. 164p. Dissertação (Mestrado em Engenharia de Estruturas) - Escola de Engenharia de São Carlos – Universidade de São

Paulo, São Carlos.

CARVALHO, B.P.R. O estado da arte em métodos para reconhecimento de padrões: Support vector machine. In: Congresso Nacional de Tecnologia da Informação e Comunicação (SUCESU 2005). **Anais...** Belo Horizonte, MG. 7p. 2005.

CENAPAD. **Introdução ao SAS**. Campinas: UNICAMP, 2002. 144p. Apostila.

CERA, M.C. **Uso de redes neurais para o reconhecimento de padrões**. 2005. 12p. Trabalho de Graduação (Graduação em Ciência da Computação) - Escola de Engenharia de São Carlos – Universidade de São Paulo, USP, São Carlos.

CHO, T.H.; CONNERS, R.W.; ARAMAN, P.A. A computer vision system for automated grading of rough hardwood lumber using a knowledge-based approach. In: IEEE International Conference on Systems, Man, and Cybernetics. **Annals...** California/EUA, p.345-350, 1990.

CONGALTON, R.G. A review of assessing the accuracy of classifications of remotely sensed data. **Remote sensing thematic accuracy assessment: a compendium**. Las vegas: Ed. Lynn K. Fenstermaker, 1991, p. 3-14.

CONGALTON, R.G.; MEAD, R.A. A quantitative method to test for consistency and correctness in photointerpretation. **Remote sensing thematic accuracy assessment: a compendium**. Las vegas: Ed. Lynn K. Fenstermaker, 1991, p. 260-265.

CONNERS, R.W.; CHO, T.; ARAMAN, P.A. Automated grading of rough hardwood lumber. In: 3RD International Conference on Scanning Technology in Saw-milling. **Annals...** San Francisco, Califórnia/EUA, p.xvi1-xvi15, 1989A.

CONNERS, R.W.; CHO, T.H.; ARAMAN, P.A. **Lumber grading with a computer vision system**. In: Seventeenth Annual Hardwood Symposium. Merrimac, **Annals...** Wisconsin/EUA, p.182-193, 1989B.

CONNERS, R.W.; MCMILLIN, C.W.; LIN, K.; VASQUEZ-ESPINOSA, R.E. Identifying and locating surface defects in wood: part of an automated lumber processing system. **IEEE Transactions on Pattern Analysis and Machine Intelligence**. Vol. PAMI-5, n.6, p.573-583, 1983.

COUTO, H.T.Z. Manejo de florestas e sua utilização em serraria. In: Seminário Internacional de Utilização da Madeira de Eucalipto para Serraria. **Anais...** São Paulo/Brasil, p.21-30, 1995.

FERREIRA, D.F. **Análise multivariada**. Lavras: ESAL, 1996, 400p. Apostila.

FUNCK, J.W.; ZHONG, Y.; BUTLER, D.A.; BRUNNER, C.C.; FORRER, J.B. Image segmentation algorithms applied to wood defect detection. **Computers and Electronics in Agriculture**, n.41, p.157-179, 2003.

GABOARDI, C. **Utilização de imagem de coerência SAR para classificação do uso da terra: Floresta Nacional do Tapajós**. 2003. 137p. Dissertação (Mestrado em Snsoriamento Remoto) - Instituto Nacional de Pesquisas Espaciais, São José dos Campos.

GONZALEZ, R.C.; WOODS, R.E. **Processamento de imagens digitais**. São Paulo: Editora Edgard Blücher Ltda, 2000, 509p.

HAJIMOWLANA, S.H.; MUSCEDERE, R.; JULLIEN, G.A.; ROBERTS, J.W. An In-Camera data stream processing system for defect detection in eb inspection tasks. *Real-Time Imaging*, n.5, p.23-34, 1999.

HORTON'S, I. **Beginning MFC programming**. Wrox Press Ltd., 1997, 613p.

HORTON'S, I. **Beginning Visual C++ 6**. Wrox Press Ltd., 1998, 1181p.

IIVARINEN, J.; VISA, A. **An adaptive texture and shape based defect classification**. In: Proceedings of 14<sup>th</sup> International Conference on Pattern Recognition, **Annals...** Brisbane, Australia. v.1, p.117-122, 1998.

KAUPPINEN, H. **Development of a color machine vision method for wood surface inspection**. 1999. 133p. Dissertation - Department of Electrical Engineering and Infotech Oulu - University of Oulu, Finlandia.

KHOURY JUNIOR, J.K. **Desenvolvimento e avaliação de um sistema de visão artificial para classificação de madeira serrada de eucalipto**. 2004. 115p. Tese (Doutorado em Engenharia Agrícola) - Departamento de Engenharia Agrícola da Universidade Federal de Viçosa, Viçosa.

KLINE, D.E.; SURAK, C.; ARAMAN, P.A. Automated hardwood lumber grading utilizing a multiple sensor machine vision technology. **Computers and electronics in agriculture**, n.41, p.139-155, 2003.

KLINE, D.E.; HOU, Y.J.; CONNERS, R.W.; SCHMOLDT, D.L.; ARAMAN, P.A. **Lumber scanning system for surface defect detection**. In: ASAE International Winter Meeting. **Annals...** Tennessee/EUA. p.1-14, 1992.

KLINKHACHORN, P.; MOODY, J.; ARAMA, P. **Automated lumber processyng system: grading the hardwood lumber?**, In: 33th Annual Hardwood Simposium Proceedings, p.77-83, 1995.

KOIVO, A.J.; KIM, C.W. Automatic classification of surface defects on red oak

boards. **Forest Products Journal**, v.39, n.9, p. 22-30, 1989.

LANDIS, J.R., KOCH, C.H. The measurement of observer agreement for categorical data. **Biometrics**, v.33, n.3, p.159-174, 1977.

LEE, S.; ABBOTT, A.L.; SCHMOLDT, D.L. Surface shape analysis of rough lumber for wane detection. **Computers and electronics in agriculture**, n.41, p.121-137, 2003.

LU, Q. **A real-time system for color sorting edge-glued panel parts**. 1997. 118p. Dissertation - Faculty of the Virginia Polytechnic Institute and State University. Blacksburg, Virginia.

MACIEL, M.W. **A educação ambiental como instrumento na busca de soluções para os problemas socioambientais na Ilha dos Marinheiros**. Departamento de Divulgação Técnico-Científica e Educação Ambiental do IBAMA. Série Meio Ambiente em Debate. Brasília, v.28, 104p., 1998.

MATLAB User's Guide, The Math Works Inc., 2000.

MÁXIMO, O.A.; FERNANDES, D. Classificação supervisionada de imagens SAR do SIVAM pré-filtradas. In: XII Simpósio Brasileiro de Sensoriamento Remoto. **Anais...** Goiânia, p.4139-4146., 2005.

OJA, J.; WALLBACKS L.; GRUNBERG, S.; HAGERDAL, E.; GRONLUND, A. Automatic grading os Scot pine (*Pinus silvestris L.*) sawlogs using an industrial X-ray log scanner. **Computers and electronics in agriculture**, n.41, p.67-75, 2003.

PONCE, R.H. Madeira serrada de eucalipto: desafios e perspectivas. In: Seminário Internacional de Utilização da Madeira de Eucalipto para Serraria.

**Anais...** São Paulo, p.50-58, 1995.

PULNix America, 2005. **Specifications of the Camera Link Interface Standard for Digital Cameras and Frame Grabbers**. PULNix America and other companies. Disponível em:  
<<http://www.compumodules.com/pdf/CameraLinkOfficial.pdf>>. Acesso em: 17 fev. 2005.

REGAZZI, A.J. **Análise multivariada**, notas de aula INF 766. Departamento de Informática da Universidade Federal de Viçosa, vol. 2, 2000.

ROCHA, M.P. **Eucalyptus grandis Hill ex Maiden e Eucalyptus dunnii Maiden como Fontes de Matéria Prima para Serrarias**. 2000. 185p. Tese (Doutorado em Ciências Florestais) - Departamento de Engenharia Florestal da Universidade Federal do Paraná, Curitiba.

ROSENFELD, G.H.; FITZPATRICK-LINS, K. A coefficient of agreement as a measure of thematic classification accuracy. **Remote sensing thematic accuracy assessment: a compendium**. Las vegas: Ed. Lynn K. Fenstermaker, 1991, p. 268-272.

SARIGUL, E.; ABBOTT, A.L.; SCHMOLDT, D.L. Rule-driven defect detection in CT images of hardwood logs. **Computers and electronics in agriculture**, n.41, p.101-119, 2003.

SAS Institute, 1999. **SAS Online Doc**, Version 8, SAS Institute Inc., Cary, NC, USA.

SEARLE, S.R. **Matrix algebra useful for statistics**. New York: John Wiley, 1982, 438p.

SILVA, J. C. Influência dos tratamentos silviculturais na qualidade da madeira. **Revista da Madeira**, v.8, n.75, p.3-10, 2003.

SILVA, J.C. Qualidade da madeira serrada de eucalipto. **Revista da Madeira**, v.9, n.50, p.44-46, 2000.

SIMÕES, J.W.; COELHO, A.S.R.; MELLO, H.A.; COUTO, H.T.Z. **Crescimento e produção de madeira de eucalipto**. Instituto de Pesquisa e Estudos Florestais - IPEF, Piracicaba, n.20, p.77-97, 1980.

STORY, M. e CONGALTON, R.G. Accuracy assessment: a user's perspective. **Remote sensing thematic accuracy assessment: a compendium**. Las Vegas: Ed. Lynn K. Fenstermaker, 1991, p. 257-259.

TOU, J.T. e GONZALES, R.C. **Pattern recognition principles**. Addison-Wesley Publishing Company, Reading, 1974, 377p.

VIEIRA, C.A.O. **Accuracy of remotely sensing classification of agricultural crops: a comparative study**. 2000. 352p. Ph.D. Thesis - School of Geography – University of Nottingham, Nottingham, England.

VITAL, B.R. e DELLA LUCIA, R.M. Propriedades físicas e mecânicas da madeira de eucalipto. **Informe Agropecuário**, Belo Horizonte, ano 12, n.141, 1986. p.71-74.

YANG, X.; BEYENAL, H.; GARY H.; LEWANDOWSKI, Z. **Evaluation of biofilm image thresholding methods**. Water Research, London, v.35, n.5, p. 1149-1158, 2001.

## **8 APÊNDICES**

**Código-fonte do algoritmo implementado no Visual C++ para acessar a câmera, obter uma imagem e salva-la na memória principal para posterior acesso pelo Matlab.**

```
// Modulo usado para a inicializacao da camera
// e para a obtencao de uma imagem no formato desejado.
// Apos a obtencao da imagem sai do modulo liberando memoria,
// com a imagem desejada ja gravada no drive virtual
// na memoria principal do microcomputador, pronta para ser
// acessada por qualquer aplicativo com um comando simples de leitura.

#include "stdafx.h"
#include "seqsnap.h"
#include "MainFrm.h"
#include "seqsnapD.h"
#include "seqsnapV.h"
#include <ifc.h>
#include <windowsx.h>

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

BEGIN_MESSAGE_MAP(CSeqsnapApp, CWinApp)
//{{AFX_MSG_MAP(CSeqsnapApp)
ON_COMMAND(ID_APP_ABOUT, OnAppAbout)
// NOTE - the ClassWizard will add and remove mapping macros here.
// DO NOT EDIT what you see in these blocks of generated code!
//}}AFX_MSG_MAP
// Standard file based document commands
ON_COMMAND(ID_FILE_NEW, CWinApp::OnFileNew)
ON_COMMAND(ID_FILE_OPEN, CWinApp::OnFileOpen)
END_MESSAGE_MAP()

void CSeqsnapApp::InitIFC(void)
{
    m_iActiveBoard=0;
    if (m_lpCmdLine && m_lpCmdLine[0])
        sscanf(m_lpCmdLine, "%d ", &m_iActiveBoard);
    if (!(m_iCap=IfxCreateCaptureModule("LNK",m_iActiveBoard,"lnktest.txt")))
    { if (!(m_iCap=IfxCreateCaptureModule("LNK",m_iActiveBoard))
        { ::MessageBox(NULL,"Specified Image Capture Module not detected.", "No
Module",MB_OK); exit(0); }
        m_iCap->ProcessCameraFilesInDir("camdb",TRUE);
    }
    m_cam = m_iCap->GetCam(0); CAM_ATTR attr; // Get the camera's basic attributes
    m_cam->GetAttr(&attr); m_srcBitspp = attr.dwBitsPerPixel;
}

CSeqsnapApp::CSeqsnapApp() // CSeqsnapApp construction
{}

CSeqsnapApp::~CSeqsnapApp()
{ if (m_iCap) delete m_iCap; }
```

```

CSeqsnapApp theApp; // The one and only CSeqsnapApp object

BOOL CSeqsnapApp::InitInstance() // CSeqsnapApp initialization
{
    AfxEnableControlContainer();
#ifdef _AFXDLL
    Enable3dControls(); // Call this when using MFC in a shared DLL
#else
    Enable3dControlsStatic(); // Call this when linking to MFC statically
#endif
    SetRegistryKey(_T("Local AppWizard-Generated Applications"));
    // LoadStdProfileSettings(); // Load standard INI file options (including MRU)
    // CSingleDocTemplate* pDocTemplate; // main SDI frame window
    // pDocTemplate = new CSingleDocTemplate(IDR_MAINFRAME,RUNTIME_CLASS(CSeqsnapDoc),
    //
    RUNTIME_CLASS(CMainFrame),RUNTIME_CLASS(CSeqsnapView));
    // AddDocTemplate(pDocTemplate);
    // CCommandLineInfo cmdInfo; // Parse command line for standard shell commands, DDE, file open
    // ParseCommandLine(cmdInfo);
    InitIFC();

    // -----
    // Vindo de SeqSnapV.cpp do Procedimento void CSeqsnapView::OnCmdsWritetiff()
    CSeqsnapApp *theapp=(CSeqsnapApp*)AfxGetApp();
    DWORD m_areaDx, m_areaDy; // DX and DY of the image area being processed
    CAM_ATTR attr; BYTE *m_imgBuf; // Memory space which is the destination of the ITX grab operation
    CString fname="Z:Imagem.bmp"; // dgf.GetPathName();
    theapp->m_cam->GetAttr(&attr); // Get the current acquisition dimensions
    m_areaDx = attr.dwWidth; m_areaDy = attr.dwHeight;
    IFC_BAYER_FILTER_TYPE bayerFilter=(IFC_BAYER_FILTER_TYPE)(DWORD)theapp->m_cam-
>GetAcqParam(P_BAYER_FILTER);
    m_imgBuf = (BYTE*)GlobalAllocPtr(GMEM_FIXED,m_areaDx * m_areaDy * attr.dwBytesPerPixel); //
    Allocate space for the image
    theapp->m_cam->Snap(m_imgBuf); // Snap a single image from the camera into memory
    DWORD bayOptions=0;
    if ((DWORD)theapp->m_cam->GetAcqParam(P_PIXEL_SIZE) == 10)
        bayOptions |= IFC_BAYOPT_10BIT_SRC;
    BYTE *fileBuf=(BYTE*)malloc( m_areaDx * m_areaDy *4);
    IfxBayerFilterConvert(bayOptions,bayerFilter, fileBuf, m_imgBuf,m_areaDx,
m_areaDy,IFC_RGB_PACK24,NULL);
    IfxWriteImgFile(fname.GetBuffer(80),fileBuf,m_areaDx,m_areaDy,24,IFFCL_RGB); free(fileBuf);
    // -----
    // Termina a vinda aqui
    exit(0);
    // Sai liberando memoria que foi usada
    // Imagem ja gravada no drive virtual z: na memoria principal do microcomputador
    // -----

    // if (!ProcessShellCommand(cmdInfo)) // Dispatch commands specified on the command line
    // return FALSE;
    // m_pMainWnd->ShowWindow(SW_SHOW); m_pMainWnd->UpdateWindow(); // The one and only
    window has been initialized, so show and update it.

    return TRUE;
}

class CAboutDlg : public CDialog
{
public:

```

```

    CAboutDlg();

// Dialog Data
//{{AFX_DATA(CAboutDlg)
enum { IDD = IDD_ABOUTBOX };
//}}AFX_DATA

// ClassWizard generated virtual function overrides
//{{AFX_VIRTUAL(CAboutDlg)
protected:
virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV support
//}}AFX_VIRTUAL

// Implementation
protected:
//{{AFX_MSG(CAboutDlg)
// No message handlers
//}}AFX_MSG
DECLARE_MESSAGE_MAP()
};

CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)
{
//{{AFX_DATA_INIT(CAboutDlg)
//}}AFX_DATA_INIT
}

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    {{{AFX_DATA_MAP(CAboutDlg)
    }}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CAboutDlg, CDialog)
    {{{AFX_MSG_MAP(CAboutDlg)
    // No message handlers
    }}}AFX_MSG_MAP
END_MESSAGE_MAP()

// App command to run the dialog
void CSeqsnapApp::OnAppAbout()
{
    CAboutDlg aboutDlg;
    aboutDlg.DoModal();
}

```

**Código-fonte do algoritmo implementado no SAS para validar os parâmetros calculados para o classificador estatístico.**

```
PROC IMPORT OUT= WORK.MatC45VO
  DATAFILE= "C:\Documents and Settings\Usuario\Meus documentos \My SAS Files\V8\MatCars45VO -
SAS.xls"
  DBMS=EXCEL2000 REPLACE;
  GETNAMES=YES;
RUN;

proc discrim data=MatC45VO outstat=Saida method=normal pool=yes crossvalidate;
class Classe;
var P02R P05R P10R P20R P30R P40R P50R P60R P70R P80R P90R P95R
    P02G P05G P10G P20G P30G P40G P50G P60G P70G P80G P90G P95G
    P02B P05B P10B P20B P30B P40B P50B P60B P70B P80B P90B P95B
    MednR MednG DesvPR MedG DesvPG MedB MednB DesvPB;
priors '1'=0.95 '2'=0.0106 '3'=0.0236 '4'=0.0039 '5'=0.003 '6'=0.0089;
run; quit;
```

**Código-fonte do algoritmo implementado no MatLab para geração dos parâmetros do classificador estatístico e para classificação das tábuas.**

```
% MODULO PRINCIPAL - Chama todos os outros, na sequencia correta
clear; clc;

global RegraClassif ModoClassif NunVOs VOsaUsar ClassifFinalTab ClasseTab ResultClassif Mensagem Cam;
% RegraClassif: 1 = ABNT, 2 = ARACRUZ
global LTAAtual LTAAtualAux LTAAtualLF NumLinsTab NumColsTab;
global ProbsPosClassesNVO VetMediasCarsNVO InvMatCovComumNVO;
global ContBG ContEsm ContFen ContNoS ContRac;
global NumTotalBlocosClassifs NumAltersClassifsBlocos NumBlocosML;

InicializaVOsaUsar;
RegraClassif = Dialogos(1);
ModoClassif = Dialogos(2); % ModoClassif: 1 = Interativa, 2 = Automatica
NumTotalBlocosClassifs = 0; NumAltersClassifsBlocos = 0; NumBlocosML = 0;
NumTabuas = 2; Cam = 'D:\Tabuas\Jai\SemEsteira\JpT'; ResultClassif = uint8(ones(1,NumTabuas));
load ProbsPosClassesNVO.mat; load VetMediasCarsNVO.mat; load InvMatCovComumNVO.mat;
ProbsPosClassesNVO = ProbsPosClassesNVO; VetMediasCarsNVO = VetMediasCarsNVO;
InvMatCovComumNVO = InvMatCovComumNVO;

if RegraClassif == 'ABNT '
    for TNum = 1:NumTabuas
        if ModoClassif == 'Interativo' Dialogos(3); end % Interativa
        Face = 1; ContBG = 0; ContEsm = 0; ContFen = 0; ContNoS = 0; ContRac = 0;
        ContEsmF1 = 0; ContEsmF2 = 0; MaiorLargEsmF1 = 0; MaiorLargEsmF2 = 0; MaiorLargEsm = 0;

        LTAAtual = MontaLadoTabInteiras(TNum,Face); LTAAtualAux = LTAAtual;

        [NumLinsTab,NumColsTab,Nada] = size(LTAAtual);
        NumTotalBlocos = (NumLinsTab/64)*(NumColsTab/64); Larg = NumColsTab/42; Compr =
        NumLinsTab/42;
        ObtemDadosFace;
        ContEsmF1 = ContEsm; MaiorLargEsmF1 = MaiorLargEsm;
        if (ContEsmF1 > 5) & (ContEsmF1 < 25) MaiorLargEsmF1 = 64/42; elseif ContEsmF1 > 25
        MaiorLargEsmF1 = 128/42; end
        ClasseFA = 0; ContNoSFA = ContNoS; ContBGFA = ContBG;
        [ClasseFA,Mensagem] =
        ClassificaFaceTabABNT(Larg,Compr,MaiorLargEsm,ContEsmF1,ContBG,ContFen,ContRac,ContNoS);
        Dialogos(4);
        Face = 2; ContBG = 0; ContEsm = 0; ContFen = 0; ContNoS = 0; ContRac = 0;
        LTAAtual = MontaLadoTabInteiras(TNum,Face); LTAAtualAux = LTAAtual;

        [NumLinsTab,NumColsTab,Nada] = size(LTAAtual);
        NumTotalBlocos = (NumColsTab/64)*(NumLinsTab/64); Larg = NumColsTab/42; Compr =
        NumLinsTab/42;
        ObtemDadosFace;
        ContEsmF2 = ContEsm; MaiorLargEsmF2 = MaiorLargEsm;
        if (ContEsmF2 > 5) & (ContEsmF2 < 25) MaiorLargEsmF2 = 64/42; elseif ContEsmF2 > 25
        MaiorLargEsmF2 = 128/42; end
        ClasseFB = 0; ContNoSFB = ContNoS; ContBGFB = ContBG;
        if MaiorLargEsmF1 > MaiorLargEsmF2 MaiorLargEsm = MaiorLargEsmF1; else MaiorLargEsm =
        MaiorLargEsmF2; end

        [ClasseFB,Mensagem] =
```

```

ClassificaFaceTabABNT(Larg,Compr,MaiorLargEsm,ContEsmF2,ContBG,ContFen,ContRac,ContNoS);
  ClassifFinalTab =
ClassificaTabABNT(ClasseFA,ClasseFB,ContNoSFA+ContNoSFB,ContBGFA+ContBGFB,Compr);
  disp(['Tab' num2str(TNum) ' = ' num2str(ClassifFinalTab), ' -- ',F1=',num2str(ClasseFA),'
F2=',num2str(ClasseFB)]);
  ResultClassif (TNum) = ClassifFinalTab;
  Dialogos(5);
end
%
#####
####
elseif RegraClassif == 'ARACRUZ'
  for TNum = 1:NumTabuas
    if ModoClassif == 'Interativo' Dialogos(3); end % Interativa
    Face = 1; ContBG = 0; ContEsm = 0; ContFen = 0; ContNoS = 0; ContRac = 0;
    ContEsmF1 = 0; ContEsmF2 = 0; MaiorLargEsm = 0; MaiorLargEsmF1 = 0; MaiorLargEsmF2 = 0;
    RendMeF = 0; RendPiF = 0;
    LTAAtual = MontaLadoTabInteiras(TNum,Face); LTAAtualAux = LTAAtual;

    [NumLinsTab,NumColsTab,Nada] = size(LTAAtual);
    NumTotalBlocos = (NumColsTab/64)*(NumLinsTab/64); Larg = NumColsTab/42; Compr =
    NumLinsTab/42;
    ObtemDadosFace;
    ContRacF1 = ContRac; ContBGF1 = ContBG; ContEsmF1 = ContEsm; ContNoSF1 = ContNoS;
    ContFenF1 = ContFen;
    SomaDefsF1 = ContBG+ContEsmF1+ContFen+ContNoS+ContRac;
    if (ContEsmF1 > 5) & (ContEsmF1 < 25) MaiorLargEsmF1 = 64/42; elseif ContEsmF1 > 25
    MaiorLargEsmF1 = 128/42; end
    RendMeF = 0.9 * (NumTotalBlocos -
    (ContBG+ContEsmF1+ContFen+ContNoS+ContRac))/NumTotalBlocos;
    Dialogos(4);
    Face = 2;
    LTAAtual = MontaLadoTabInteiras(TNum,Face); LTAAtualAux = LTAAtual;

    [NumLinsTab,NumColsTab,Nada] = size(LTAAtual);
    NumTotalBlocos = (NumColsTab/64) * (NumLinsTab/64); Larg = NumColsTab/42; Compr =
    NumLinsTab/42;
    ObtemDadosFace;
    ContRacF2 = ContRac - ContRacF1; ContBGF2 = ContBG - ContBGF1;
    ContFenF2 = ContFen - ContFenF1; ContNoSF2 = ContNoS - ContNoSF1; ContEsmF2 = ContEsm -
    ContEsmF1;
    if (ContEsmF2 > 5) & (ContEsmF2 < 25) MaiorLargEsmF2 = 64/42; elseif ContEsmF2 > 25
    MaiorLargEsmF2 = 128/42; end
    RendPiF = 0.9 * (NumTotalBlocos -
    (ContBG+ContEsmF2+ContFen+ContNoS+ContRac))/NumTotalBlocos;
    if RendMeF < RendPiF [RendMeF,RendPiF] = TrocaValores(RendMeF,RendPiF); end
    if MaiorLargEsmF1 > MaiorLargEsmF2 MaiorLargEsm = MaiorLargEsmF1; else MaiorLargEsm =
    MaiorLargEsmF2; end

    [ClasseTab,Mensagem] = ClassificaTabARACRUZ...
    (Larg,Compr,RendMeF,RendPiF,ContRacF1+ContRacF2,ContBGF1+ContBGF2,MaiorLargEsm,ContEs
    mF1+ContEsmF2);
    ResultClassif (TNum) = ClasseTab; disp([num2str(TNum) ' - ' Mensagem]);
    Dialogos(6);
  end
else disp('Saindo...'); end

MatConf = GeraMatrizConfusao(NumTabuas)
if RegraClassif == 'ABNT '

```

```

    PA =
    (MatConf(1,1)+MatConf(2,2)+MatConf(3,3)+MatConf(4,4)+MatConf(5,5)+MatConf(6,6))/sum(sum(MatConf))
;
else PA = (MatConf(1,1)+MatConf(2,2)+MatConf(3,3)+MatConf(4,4)+MatConf(5,5))/sum(sum(MatConf));
end
disp(['Acerto(%)= ',num2str(PA),' - Kappa= ',num2str(CalculaIndiceKappa(MatConf))]);
disp(['Num. blocos classificados= ',num2str(NumTotalBlocosClassifs),...
      ' - Defeitos= ',num2str(NumTotalBlocosClassifs - NumBlocosML),' - Alteracoes=
      ',num2str(NumAltersClassifsBlocos)]);

```

---

```

% Gera a parametros para o classificador estatistico

```

```

disp('Gerando "Vetor de Probabilidades das Caracteristicas":');
NBsML = 500; NBsBG = 80; NBsEsm = 80; NBsFen = 60; NBsNoS = 14; NBsRac = 80; %
Treinamento
ProbsPosClassesNVO = [0.9500 0.0106 0.0236 0.0039 0.0030 0.0089];
save ProbsPosClassesNVO.mat ProbsPosClassesNVO; disp(ProbsPosClassesNVO);
disp(sum(ProbsPosClassesNVO));

disp('Gerando "MatrizCaracteristicas"- "VetorMediasCaracteristicas"- "InversaMatrizCovarianciaComum":');
MatCovComumNVO = double(zeros(NumVOS,NumVOS)); VetMediasCarsNVO = double(zeros(NumVOS,6));

load CarsML.mat; MatCarsTemp = zeros(NBsML,NumVOS); CarsTemp = zeros(1,NumVOS);
for Cont = 1:NBsML
    CarsTemp = RetornaCarsAUsar(CarsML(Cont,:));
    VetMediasCarsNVO(1:NumVOS,1) = VetMediasCarsNVO(1:NumVOS,1) + CarsTemp';
    MatCarsTemp(Cont,1:NumVOS) = CarsTemp;
end; %disp('Madeira limpa - Ok. ');
VetMediasCarsNVO(1:NumVOS,1) = VetMediasCarsNVO(1:NumVOS,1)/NBsML;
MatCovComumNVO = (NBsML - 1) * cov(MatCarsTemp);
MatCarsNVO(1:NBsML,:) = MatCarsTemp;

load CarsBG.mat; MatCarsTemp = zeros(NBsBG,NumVOS);
for Cont = 1:NBsBG
    CarsTemp = RetornaCarsAUsar(CarsBG(Cont,:));
    VetMediasCarsNVO(1:NumVOS,2) = VetMediasCarsNVO(1:NumVOS,2) + CarsTemp';
    MatCarsTemp(Cont,1:NumVOS) = CarsTemp;
end; %disp('Bolsa goma ---- Ok. ');
VetMediasCarsNVO(1:NumVOS,2) = VetMediasCarsNVO(1:NumVOS,2)/NBsBG;
MatCovComumNVO = MatCovComumNVO + (NBsBG - 1) * cov(MatCarsTemp);
MatCarsNVO(NBsML+1:NBsML+NBsBG,:) = MatCarsTemp;

load CarsEsm.mat; MatCarsTemp = zeros(NBsEsm,NumVOS);
for Cont = 1:NBsEsm
    CarsTemp = RetornaCarsAUsar(CarsEsm(Cont,:));
    VetMediasCarsNVO(1:NumVOS,3) = VetMediasCarsNVO(1:NumVOS,3) + CarsTemp';
    MatCarsTemp(Cont,1:NumVOS) = CarsTemp;
end; %disp('Esmoadado ----- Ok. ');
VetMediasCarsNVO(1:NumVOS,3) = VetMediasCarsNVO(1:NumVOS,3)/NBsEsm;
MatCovComumNVO = MatCovComumNVO + (NBsEsm - 1) * cov(MatCarsTemp);
MatCarsNVO(NBsML+NBsBG+1:NBsML+NBsBG+NBsEsm,:) = MatCarsTemp;

load CarsFen.mat; MatCarsTemp = zeros(NBsFen,NumVOS);
for Cont = 1:NBsFen
    CarsTemp = RetornaCarsAUsar(CarsFen(Cont,:));
    VetMediasCarsNVO(1:NumVOS,4) = VetMediasCarsNVO(1:NumVOS,4) + CarsTemp';
    MatCarsTemp(Cont,1:NumVOS) = CarsTemp;

```

```

end; %disp('Fendilhado ---- Ok. ');
VetMediasCarsNVO(1:NumVOs,4) = VetMediasCarsNVO(1:NumVOs,4)/NBsFen;
MatCovComumNVO = MatCovComumNVO + (NBsFen - 1) * cov(MatCarsTemp);
MatCarsNVO(NBsML+NBsBG+NBsEsm+1:NBsML+NBsBG+NBsEsm+NBsFen,:) = MatCarsTemp;

load CarsNoS.mat; MatCarsTemp = zeros(NBsNoS,NumVOs);
for Cont = 1:NBsNoS
    CarsTemp = RetornaCarsAUsar(CarsNoS(Cont,:));
    VetMediasCarsNVO(1:NumVOs,5) = VetMediasCarsNVO(1:NumVOs,5) + CarsTemp';
    MatCarsTemp(Cont,1:NumVOs) = CarsTemp;
end; %disp('No sadio ----- Ok. ');
VetMediasCarsNVO(1:NumVOs,5) = VetMediasCarsNVO(1:NumVOs,5)/NBsNoS;
MatCovComumNVO = MatCovComumNVO + (NBsNoS - 1) * cov(MatCarsTemp);
MatCarsNVO(NBsML+NBsBG+NBsEsm+NBsFen+1:NBsML+NBsBG+NBsEsm+NBsFen+NBsNoS,:) =
MatCarsTemp;

load CarsRac.mat; MatCarsTemp = zeros(NBsRac,NumVOs);
for Cont = 1:NBsRac
    CarsTemp = RetornaCarsAUsar(CarsRac(Cont,:));
    VetMediasCarsNVO(1:NumVOs,6) = VetMediasCarsNVO(1:NumVOs,6) + CarsTemp';
    MatCarsTemp(Cont,1:NumVOs) = CarsTemp;
end; %disp('Racha ----- Ok. ');
VetMediasCarsNVO(1:NumVOs,6) = VetMediasCarsNVO(1:NumVOs,6)/NBsRac;
MatCovComumNVO = MatCovComumNVO + (NBsRac - 1) * cov(MatCarsTemp);
MatCarsNVO(NBsML+NBsBG+NBsEsm+NBsFen+NBsNoS+1:NBsML+NBsBG+NBsEsm+NBsFen+NBsNoS
+NBsRac,:) = MatCarsTemp;

MatCovComumNVO = MatCovComumNVO/(NBsML + NBsBG + NBsEsm + NBsFen + NBsNoS + NBsRac -
6);
InvMatCovComumNVO = pinv(MatCovComumNVO); % 6 = Numero de classes

save VetMediasCarsNVO.mat VetMediasCarsNVO
save InvMatCovComumNVO.mat InvMatCovComumNVO

disp('Geracao finalizada...');

```

---

```

function RetClassificaTabABNT = ClassificaTabABNT(CFA,CFB,SomaNoSFaces,SomaBGFaces,ComprT)
% Classifica a tabua, baseado na classificacao das duas faces

SomaNoSFaces = (SomaNoSFaces*64/42)/ComprT;
SomaBGFaces = (SomaBGFaces*64/42)/ComprT;

if CFA == CFB RetClassificaTabABNT = CFA; end

if CFA > CFB
    RetClassificaTabABNT = CFA;
elseif CFB > CFA
    RetClassificaTabABNT = CFB;
elseif (CFA == CFB) & (CFA ~ 1) & (CFA ~ 4)
    RetClassificaTabABNT = CFA;
elseif (CFA == 1) & (CFB == 1)
    if SomaNoSFaces > 0.025 RetClassificaTabABNT = 2; end
elseif (CFA == 4) & (CFB == 4)
    if SomaBGFaces > 0.8 RetClassificaTabABNT = 5; end % disp('Mudei...');
end

```



```

Y = 1;
while Y+63 < NumColsTab
    Y = AchaColProxDefeito (X,Y);
    if sum(sum(~LTAtualLF(X:X+63,Y:Y+63))) > 4
        [NX,NY] = CentralizaDefeito(X,Y);
        ClasseBloco = FuncaoDiscrimNVO(RetornaCarsNVO(ObtemBloco(NX,NY)));
        RegrasPosicaoDefeito (NX,NY,ClasseBloco); LTAtualLF(NX:NX+63,NY:NY+63) = 1;
    else
        LTAtualAux(X:X+63,Y:Y+63,:) = 255; LTAtualLF(X:X+63,Y:Y+63) = 1; Y = NumColsTab;
    end
end
X = AchaLinProxDefeito (X);
end;

```

---

```

function RegrasPosicaoDefeito = RegrasPosicaoDefeito (Li,Ci,NumDef)
% Preenche o entorno do defeito com a sua cor associada
% Faz ajustes do tipo do defeito de acordo com suas características morfológicas e informações de posição

```

```

global ContBG ContEsm ContFen ContNoS ContRac;
global LTAtual LTAtualAux LTAtualLF NumLinsTab NumColsTab;
global NumTotalBlocosClassifs NumAltersClassifsBlocos NumBlocosML;

```

```

Lf = Li+63; Cf = Ci+63; NumDefInicial = NumDef;
NumTotalBlocosClassifs = NumTotalBlocosClassifs + 1;
[AltDef,LargDef] = AltLargDefeito(LTAtualLF(Li:Lf,Ci:Cf));
switch NumDef
    case 2 % BOLSA DE GOMA,
        if (Ci == 1) | (Cf == NumColsTab) % mas esta na beirada,
            if AltDef > 57 % e altura do defeito eh significativa,
                NumDef = 3; end; end % entao passa a ser Esmoadado...
            if (Li < 64) | (Lf > (NumLinsTab-64)) % mas se estiver no topo,
                NumDef = 6; end % passa a ser Racha...
        end
    case 3 % ESMOADO,
        if (Ci > 70) & (Cf < (NumColsTab-70)) % mas nao esta na beirada,
            NumDef = 2; end % entao eh uma Bolsa de Goma...
            if AltDef < 36 % mas altura do defeito nao eh significativa,
                NumDef = 2; end % entao passa a ser Bolsa de Goma...
        end
    case 4 % FENDILHADO,
        if (Li < 300) | (Lf > (NumLinsTab-300)) % mas esta na extremidade,
            NumDef = 6; end % entao passa a ser Racha...
            if AltDef < 36 % mas nao eh alto,
                NumDef = 2; end % entao eh uma Bolsa de Goma...
            if (Ci == 1) | (Cf == NumColsTab) % mas esta na beirada,
                NumDef = 3; end % entao pasa a ser Esmoadado...
        end
    case 5 % NO SADIO,
        if (Ci == 1) | (Cf == NumColsTab) % mas esta na beirada,
            if (AltDef > 57) % & (sum(sum(~LTAtualLF(Li:Lf,Ci:Cf))) > 350)
                NumDef = 3; end; end % entao passa a ser Esmoadado...
        end
    case 6 % RACHA,
        if (Li > 300) & (Lf < (NumLinsTab-300)) % mas nao esta na extremidade,
            NumDef = 4; end % entao passa a ser Fendilhado...
            if AltDef < 36 % mas nao eh esticado,
                NumDef = 2; end % entao eh uma Bolsa de Goma...
            if (Ci == 1) | (Cf == NumColsTab) % mas esta na beirada,
                NumDef = 3; end % entao pasa a ser Esmoadado...
        end
end

```

LTAtualAux(Li:Lf,Ci:Cf,:) = 255;

switch NumDef

case 1 % MADEIRA LIMPA = Verde

NumBlocosML = NumBlocosML + 1;

% LTAtual(Li:Li+63,Ci,2) = 255; LTAtual(Li:Li+63,Ci+63,2) = 255;

% LTAtual(Li,Ci:Li+63,2) = 255; LTAtual(Li+63,Ci:Li+63,2) = 255;

% LTAtual(Li:Li+63,Ci,1) = 0; LTAtual(Li:Li+63,Ci+63,1) = 0;

% LTAtual(Li,Ci:Li+63,1) = 0; LTAtual(Li+63,Ci:Li+63,1) = 0;

% LTAtual(Li:Li+63,Ci,3) = 0; LTAtual(Li:Li+63,Ci+63,3) = 0;

% LTAtual(Li,Ci:Li+63,3) = 0; LTAtual(Li+63,Ci:Li+63,3) = 0;

case 2 % BOLSA DE GOMA = Preto

% LTAtual(Li:Li+63,Ci,:) = 0; LTAtual(Li:Li+63,Ci+63,:) = 0;

% LTAtual(Li,Ci:Li+63,:) = 0; LTAtual(Li+63,Ci:Li+63,:) = 0;

% LTAtual(Li+04:Li+12,Ci+04,:) = 255; LTAtual(Li+05:Li+11,Ci+07,:) = 255;

% LTAtual(Li+04,Ci+04:Li+06,:) = 255; LTAtual(Li+08,Ci+04:Li+06,:) = 255;

% LTAtual(Li+12,Ci+04:Li+06,:) = 255;

ContBG = ContBG + 1;

case 3 % ESMOADO = Cinza

% LTAtual(Li:Li+63,Ci,:) = 127; LTAtual(Li:Li+63,Ci+63,:) = 127;

% LTAtual(Li,Ci:Li+63,:) = 127; LTAtual(Li+63,Ci:Li+63,:) = 127;

% LTAtual(Li+04:Li+12,Ci+04,:) = 127; LTAtual(Li+04,Ci+04:Li+08,:) = 127;

% LTAtual(Li+08,Ci+04:Li+06,:) = 127; LTAtual(Li+12,Ci+04:Li+08,:) = 127;

ContEsm = ContEsm + 1;

case 4 % FENDILHADO - Amarelo

% LTAtual(Li:Li+63,Ci,:) = 255; LTAtual(Li:Li+63,Ci+63,:) = 255;

% LTAtual(Li,Ci:Li+63,:) = 255; LTAtual(Li+63,Ci:Li+63,:) = 255;

% LTAtual(Li:Li+63,Ci,3) = 0; LTAtual(Li:Li+63,Ci+63,3) = 0;

% LTAtual(Li,Ci:Li+63,3) = 0; LTAtual(Li+63,Ci:Li+63,3) = 0;

% LTAtual(Li+04:Li+12,Ci+04,:) = 255; LTAtual(Li+04,Ci+04:Li+08,:) = 255;

% LTAtual(Li+08,Ci+04:Li+06,:) = 255;

% LTAtual(Li+04:Li+12,Ci+04,3) = 0; LTAtual(Li+04,Ci+04:Li+08,3) = 0;

% LTAtual(Li+08,Ci+04:Li+06,3) = 0;

ContFen = ContFen + 1;

case 5 % NO SADIO = Branco

% LTAtual(Li:Li+63,Ci,:) = 255; LTAtual(Li:Li+63,Ci+63,:) = 255;

% LTAtual(Li,Ci:Li+63,:) = 255; LTAtual(Li+63,Ci:Li+63,:) = 255;

% LTAtual(Li+04:Li+12,Ci+04,:) = 255; LTAtual(Li+04:Li+12,Ci+08,:) = 255;

% LTAtual(Li+06,Ci+05,:) = 255; LTAtual(Li+08,Ci+06,:) = 255;

% LTAtual(Li+10,Ci+07,:) = 255;

ContNoS = ContNoS + 1;

case 6 % RACHA = Vermelho

% LTAtual(Li:Li+63,Ci,1) = 255; LTAtual(Li:Li+63,Ci+63,1) = 255;

% LTAtual(Li,Ci:Li+63,1) = 255; LTAtual(Li+63,Ci:Li+63,1) = 255;

% LTAtual(Li+04:Li+12,Ci+04,1) = 255; LTAtual(Li+05:Li+07,Ci+07,1) = 255;

% LTAtual(Li+04,Ci+04:Li+06,1) = 255; LTAtual(Li+08,Ci+04:Li+06,1) = 255;

% LTAtual(Li+09,Ci+05,1) = 255; LTAtual(Li+10,Ci+06,1) = 255;

% LTAtual(Li+11,Ci+07,1) = 255; LTAtual(Li+12,Ci+08,1) = 255;

ContRac = ContRac + 1;

end

if NumDef ~= NumDefInicial

NumAltersClassifsBlocos = NumAltersClassifsBlocos + 1;

% switch NumDefInicial

% case 2 % BOLSA DE GOMA = B

% LTAtual(Li+04:Li+12,Ci+52,:) = 255; LTAtual(Li+05:Li+11,Ci+55,:) = 255;

% LTAtual(Li+04,Ci+52:Li+06,:) = 255; LTAtual(Li+08,Ci+52:Li+06,:) = 255;

% LTAtual(Li+12,Ci+52:Li+06,:) = 255;

% case 3 % ESMOADO = E

```

%   LTAtual(Li+04:Li+12,Ci+52,:)= 255; LTAtual(Li+04,Ci+52:Li+12,Ci+56,:)= 255;
%   LTAtual(Li+08,Ci+52:Li+12,Ci+54,:)= 255; LTAtual(Li+12,Ci+52:Li+12,Ci+56,:)= 255;
% case 4 % FENDILHADO - F
%   LTAtual(Li+04:Li+12,Ci+52,:)= 255; LTAtual(Li+04,Ci+52:Li+12,Ci+56,:)= 255;
%   LTAtual(Li+08,Ci+52:Li+12,Ci+54,:)= 255;
% case 5 % NO SADIO = N
%   LTAtual(Li+04:Li+12,Ci+52,:)= 255; LTAtual(Li+04:Li+12,Ci+56,:)= 255;
%   LTAtual(Li+06,Ci+53,:)= 255; LTAtual(Li+08,Ci+54,:)= 255;
%   LTAtual(Li+10,Ci+55,:)= 255;
% case 6 % RACHA = R
%   LTAtual(Li+04:Li+12,Ci+52,:)= 255; LTAtual(Li+05:Li+07,Ci+55,:)= 255;
%   LTAtual(Li+04,Ci+52:Li+12,Ci+54,:)= 255; LTAtual(Li+08,Ci+52:Li+12,Ci+54,:)= 255;
%   LTAtual(Li+09,Ci+53,:)= 255; LTAtual(Li+10,Ci+54,:)= 255;
%   LTAtual(Li+11,Ci+55,:)= 255; LTAtual(Li+12,Ci+56,:)= 255;
% end
end

```

---

```

function RetCapturaUmaImagem = CapturaUmaImagem
!Z:\Snap
RetCapturaUmaImagem = imread('z:\Imagem.BMP');

```

---

```
clear; clc;
```

```

global ProbsPosClassesNVO VetMediasCarsNVO InvMatCovComumNVO;
global NBsML NBsBG NBsEsm NBsFen NBsNoS NBsRac;
global NunVOs VOsaUsar;
global RegraClassif ResultClassif;

```

```

InicializaVOsaUsar;
RegraClassif = 3;
ResultClassif = ones(1,NBsML+NBsBG+NBsEsm+NBsFen+NBsNoS+NBsRac);

```

```

GeraInvMatCovsVetMedsCarsNVO;
% load ProbsPosClassesNVO.mat; load VetMediasCarsNVO.mat; load InvMatCovComumNVO.mat;

```

```

load CarsML; TestaClasseNVO (CarsML,1,NBsML);
load CarsBG; TestaClasseNVO (CarsBG,NBsML+1,NBsML+NBsBG);
load CarsEsm; TestaClasseNVO (CarsEsm,NBsML+NBsBG+1,NBsML+NBsBG+NBsEsm);
load CarsFen; TestaClasseNVO (CarsFen,NBsML+NBsBG+NBsEsm+1,NBsML+NBsBG+NBsEsm+NBsFen);
load CarsNoS; TestaClasseNVO
(CarsNoS,NBsML+NBsBG+NBsEsm+NBsFen+1,NBsML+NBsBG+NBsEsm+NBsFen+NBsNoS);
load CarsRac; TestaClasseNVO
(CarsRac,NBsML+NBsBG+NBsEsm+NBsFen+NBsNoS+1,NBsML+NBsBG+NBsEsm+NBsFen+NBsNoS+NB
sRac);

```

```
MatResp = GeraMatrizConfusao(NBsML+NBsBG+NBsEsm+NBsFen+NBsNoS+NBsRac)
```

```

PA =
(MatResp(1,1)+MatResp(2,2)+MatResp(3,3)+MatResp(4,4)+MatResp(5,5)+MatResp(6,6))/sum(sum(MatResp))
;
Kappa = CalculaIndiceKappa(MatResp);
disp(['Porcentagem de acerto = ',num2str(PA),' -- Indice Kappa = ',num2str(Kappa)]);

```

---

```

function ClassifCorretaABNT_ARACRUZ = ClassifCorretaABNT_ARACRUZ(Regra)
% Define a correta classificacao de cada tabua

global NBsML NBsBG NBsEsm NBsFen NBsNoS NBsRac;

if Regra < 3 ClassCor = zeros(1,42);
else ClassCor = zeros(1,NBsML+NBsBG+NBsEsm+NBsFen+NBsNoS+NBsRac); end

if Regra == 1 % ABNT

    ClassCor(01) = 5; ClassCor(11) = 5; ClassCor(21) = 5;
    ClassCor(02) = 2; ClassCor(12) = 4; ClassCor(22) = 5;
    ClassCor(03) = 2; ClassCor(13) = 5; ClassCor(23) = 6;
    ClassCor(04) = 2; ClassCor(14) = 5; ClassCor(24) = 4;
    ClassCor(05) = 2; ClassCor(15) = 5; ClassCor(25) = 5;
    ClassCor(06) = 2; ClassCor(16) = 2; ClassCor(26) = 5;
    ClassCor(07) = 4; ClassCor(17) = 2; ClassCor(27) = 5;
    ClassCor(08) = 2; ClassCor(18) = 4; ClassCor(28) = 5;
    ClassCor(09) = 2; ClassCor(19) = 5; ClassCor(29) = 5;
    ClassCor(10) = 2; ClassCor(20) = 5; ClassCor(30) = 5;

    ClassCor(31) = 6; ClassCor(41) = 5;
    ClassCor(32) = 5; ClassCor(42) = 5;
    ClassCor(33) = 5;
    ClassCor(34) = 5;
    ClassCor(35) = 5;
    ClassCor(36) = 5;
    ClassCor(37) = 4;
    ClassCor(38) = 6;
    ClassCor(39) = 5;
    ClassCor(40) = 5;

elseif Regra == 2 % ARACRUZ
    for i = 1:18 ClassCor(i) = 2; end
    for i = 19:37 ClassCor(i) = 4; end
    for i = 38:42 ClassCor(i) = 5; end

else
% Atribui classificacao correta para calculo de Matriz de Confusao e Kappa para blocos
ClassCor(1:NBsML) = 1;
ClassCor(NBsML+1:NBsML+NBsBG) = 2;
ClassCor(NBsML+NBsBG+1:NBsML+NBsBG+NBsEsm) = 3;
ClassCor(NBsML+NBsBG+NBsEsm+1:NBsML+NBsBG+NBsEsm+NBsFen) = 4;
ClassCor(NBsML+NBsBG+NBsEsm+NBsFen+1:NBsML+NBsBG+NBsEsm+NBsFen+NBsNoS) = 5;
ClassCor(NBsML+NBsBG+NBsEsm+NBsFen+NBsNoS+1:NBsML+NBsBG+NBsEsm+NBsFen+NBsNoS+
NBsRac) = 6;
end

ClassifCorretaABNT_ARACRUZ = ClassCor;

```

---

```

function [ResClasseFaceTabABNT,Mensagem] = ClassificaFaceTabABNT...
(Larg,Compr,MaiorLargEsm1F,ComprEsm1F,ContBG1F,ComprFen1F,ComprRac,ComprNoS1F)
% Usa os dados da face da tabua, tudo em centimetros ou centimetros quadrados

ComprEsm1F = ComprEsm1F*64/42; ComprFen1F = ComprFen1F*64/42;
ComprRac = ComprRac*64/42; ComprNoS1F = ComprNoS1F*64/42;
Area = Larg * Compr; ComprBG1F = ContBG1F*64/42; AreaBG1F = ((64/42)^2)*ContBG1F;

```

```

ClassifParcial = 6;

if Compr >= 180
if Larg >= 7.5
if MaiorLargEsm1F <= 0.3*Larg
if ComprEsm1F <= 0.52*Compr
if AreaBG1F < 0.05*Area  ClassifParcial = 5;
% =====
if ComprBG1F/Compr <= 0.06 %0.025
if MaiorLargEsm1F <= 0.2*Larg
if ComprEsm1F <= 0.5*Compr
if ComprFen1F/Compr <= 0.5
if AreaBG1F <= 0.3*Area
if ComprRac/Compr < 0.15  ClassifParcial = 4;
%
% =====
if Larg >= 9
if AreaBG1F < 9.3  % == 0
if MaiorLargEsm1F <= 0.2*Larg
if ComprFen1F/Compr <= 0.3
if ComprRac/Compr < 0.1  ClassifParcial = 3;
% -----
if ComprEsm1F <= 0.2*Compr
if ComprFen1F/Compr <= 0.1
if ComprNoS1F/Compr <= 0.039
if ComprRac/Compr < 0.05  ClassifParcial = 2;
% .....
if MaiorLargEsm1F == 0
if ComprEsm1F == 0
if ComprFen1F == 0
if ComprNoS1F == 0

if ComprRac/Compr < 0.025  ClassifParcial = 1;  Mensagem = '1-Nenhum problema. Classe 1...';

else Mensagem = '2-Comprimento de Racha > 25mm/metro. Classe 2...'; end
else Mensagem = '2-Possui No Sadio/Corrompido. Classe 2...'; end
else Mensagem = '2-Possui Fendilhado. Classe 2...'; end
else Mensagem = '2-Possui Esmoadado. Classe 2...'; end
else Mensagem = '2-Possui Esmoadado. Classe 2...'; end
% .....
else Mensagem = '3-Comprimento de Racha > 50mm/metro. Classe 3...'; end
else Mensagem = '3-Comprimento de Nos Sadios/Corrompido > 25mm/metro. Classe 3...'; end
else Mensagem = '3-Comprimento de Fendilhado > 10cm/metro. Classe 3...'; end
else Mensagem = '3-Comprimento de Esmoadado > 20cm/metro. Classe 3...'; end
% -----
else Mensagem = '4-Comprimento de Racha > 100mm/metro. Classe 4...'; end
else Mensagem = '4-Comprimento de Fendilhado > 30cm/metro. Classe 4...'; end
else Mensagem = '4-Largura de Esmoadado > 10% da largura da tabua. Classe 4...'; end
else Mensagem = '4-Possui Bolsa Goma. Classe 4...'; end
else Mensagem = '4-Largura da tabua < 10cm. Classe 4...'; end
%
% =====
else Mensagem = '5-Comprimento da Racha > 150mm/metro. Classe 5...'; end
else Mensagem = '5-Area da Bolsa de Goma > 15% da area da tabua. Classe 5...'; end
else Mensagem = '5-Comprimento do Fendilhado > 50cm/metro. Classe 5...'; end
else Mensagem = '5-Comprimento do Esmoadado > 50% do comprimento da tabua. Classe 5...'; end
else Mensagem = '5-Largura do Esmoadado > 20% da largura da tabua. Classe 5...'; end
else Mensagem = '5-Comprimento da Bolsa de Goma > 25mm/metro. Classe 5...'; end

```

```

% |||||
else Mensagem = '6-Area da Bolsa Goma > 20% da area da tabua. Classe 6...'; end
else Mensagem = '6-Comprimento do Esmoado > 70% do comprimento da tabua. Classe 6...'; end
else Mensagem = '6-Maior largura do Esmoado > 30% da largura da tabua. Classe 6...'; end
else Mensagem = '6-Largura da tabua < 7,5cm. Classe 6...'; end
else Mensagem = '6-Comprimento da tabua < 180cm. Classe 6...';
end

```

```
ResClasseFaceTabABNT = ClassifParcial;
```

---

```

function RetCorrelacao_VarXj_CPYi =
Correlacao_VarXj_CPYi(CPs_AutoVets,Lambdas_AutoVals,VarsMatDados)
% Retorna a correlacao entre as variaveis originais e os CPs

[NumCPs,NADA] = size(CPs_AutoVets);
Correlacao = zeros(NumCPs);

for i = 1:NumCPs
    for j = 1:NumCPs
        Correlacao(i,j) = (CPs_AutoVets(i,j)*sqrt(Lambdas_AutoVals(i,1)))/sqrt(VarsMatDados(1,j));
    end; end;

RetCorrelacao_VarXj_CPYi = Correlacao;

```

---

```

function RetFuncaoDiscrimNVO = FuncaoDiscrimNVO (VetCars)
% Calcula o valor da funcao discriminante para o Vetor de Caracteristicas da entrada

global ProbsPosClassesNVO VetMediasCarsNVO InvMatCovComumNVO;
NumClasses = 6; ValsFDs = double(zeros(NumClasses,1));
% ate 36 percentis - ate 9 caracteristicas de Media, Mediana e Desvio Padrao
% 9 + 36 = 45 máximo de caracteristicas no total

for j = 1:NumClasses % Calcula os valores das Funcoes Discriminantes
    Termo1 = -0.5 * ((VetMediasCarsNVO(:,j))' * InvMatCovComumNVO * VetMediasCarsNVO(:,j));
    Termo2 = VetCars' * InvMatCovComumNVO * VetMediasCarsNVO(:,j);
    ValsFDs(j,1) = Termo1 + Termo2 + log(ProbsPosClassesNVO(j,1));
end

switch max(ValsFDs) % Retorna o tipo do defeito
    case ValsFDs(1,1)
        RetFuncaoDiscrimNVO = 1; % CL1-MadeiraLimpa
    case ValsFDs(2,1)
        RetFuncaoDiscrimNVO = 2; % CL2-BG-NoC-NoSE-NoSO
    case ValsFDs(3,1)
        RetFuncaoDiscrimNVO = 3; % CL3-Esmoado
    case ValsFDs(4,1)
        RetFuncaoDiscrimNVO = 4; % CL4-Fendilhado
    case ValsFDs(5,1)
        RetFuncaoDiscrimNVO = 5; % CL5-NoSad-NoSadC
    case ValsFDs(6,1)
        RetFuncaoDiscrimNVO = 6; % CL6-Racha
end

```

---

```
% Determina quais VOs serao usadas
```

```
global NumVOs VOsAUsar;
```

```
VOsAUsar = ones(1,45);
```

```
% VOs a serem retiradas recebem valor zero
```

```
VOsAUsar(01) = 1; VOsAUsar(13) = 1; VOsAUsar(25) = 0; VOsAUsar(37) = 1;  
VOsAUsar(02) = 1; VOsAUsar(14) = 1; VOsAUsar(26) = 1; VOsAUsar(38) = 1;  
VOsAUsar(03) = 1; VOsAUsar(15) = 1; VOsAUsar(27) = 1; VOsAUsar(39) = 1;  
VOsAUsar(04) = 1; VOsAUsar(16) = 1; VOsAUsar(28) = 1; VOsAUsar(40) = 1;  
VOsAUsar(05) = 1; VOsAUsar(17) = 1; VOsAUsar(29) = 1; VOsAUsar(41) = 1;  
VOsAUsar(06) = 1; VOsAUsar(18) = 1; VOsAUsar(30) = 1; VOsAUsar(42) = 1;  
VOsAUsar(07) = 1; VOsAUsar(19) = 1; VOsAUsar(31) = 1; VOsAUsar(43) = 1;  
VOsAUsar(08) = 1; VOsAUsar(20) = 1; VOsAUsar(32) = 1; VOsAUsar(44) = 1;  
VOsAUsar(09) = 1; VOsAUsar(21) = 1; VOsAUsar(33) = 1; VOsAUsar(45) = 1;  
VOsAUsar(10) = 1; VOsAUsar(22) = 1; VOsAUsar(34) = 1;  
VOsAUsar(11) = 1; VOsAUsar(23) = 1; VOsAUsar(35) = 1;  
VOsAUsar(12) = 0; VOsAUsar(24) = 0; VOsAUsar(36) = 1;
```

```
NumVOs = sum(VOsAUsar);
```

---

```
function RetLimiarizaFiltr = LimiarizaFiltr (ImgTabua)
```

```
% Limiariza e filtra a imagem da tabua, ressaltando os defeitos
```

```
global NumLinsTab NumColsTab;
```

```
DilateParam = 5; ErodeParam = 3; MedFiltParam = 5; LimiarParam = 0.22;
```

```
ImgTemp = uint8(ones(NumLinsTab+10,NumColsTab+10));
```

```
ImgTemp(6:NumLinsTab+5,6:NumColsTab+5) = im2bw(ImgTabua,LimiarParam);
```

```
% ML = NumLinsTab/2; MC = NumColsTab/2;
```

```
% ImgTemp = uint8(ones(NumLinsTab+10,NumColsTab+10));
```

```
% ImgTemp( 6:ML+5, 6:MC+5) = im2bw(ImgTabua( 1:ML, 1:MC,:),LimiarParam);
```

```
% ImgTemp( 6:ML+5, MC+5+1:NumColsTab+5) = im2bw(ImgTabua( 1:ML,
```

```
MC+1:NumColsTab,:),LimiarParam);
```

```
% ImgTemp(ML+5+1:NumLinsTab+5, 6:MC+5) = im2bw(ImgTabua(ML+1:NumLinsTab,
```

```
1:MC,:),LimiarParam);
```

```
% ImgTemp(ML+5+1:NumLinsTab+5, MC+5+1:NumColsTab+5) = im2bw(ImgTabua(ML+1:NumLinsTab,
```

```
MC+1:NumColsTab,:),LimiarParam);
```

```
% imwrite(ImgTemp,['d:\TesteLF.jpg'],'jpg','Quality',100);
```

```
% ImgTemp = dilate(ImgTemp,ones(DilateParam,DilateParam));
```

```
% ImgTemp = erode(ImgTemp,ones(ErodeParam,ErodeParam));
```

```
ImgTemp = dilate(dilate(ImgTemp,ones(1,DilateParam)),ones(DilateParam,1));
```

```
ImgTemp = erode(erode(ImgTemp,ones(1,ErodeParam)),ones(ErodeParam,ErodeParam));
```

```
ImgTemp = medfilt2(ImgTemp,[MedFiltParam MedFiltParam]);
```

```
RetLimiarizaFiltr = ImgTemp(6:NumLinsTab+5,6:NumColsTab+5);
```

---

```
function RetMontaLadoTabInteiras = MontaLadoTabInteiras (NumTabua,Lado)
% Retorna a imagem apenas da tabua imageada a partir de sua imagem completa
```

```
ImgTabua = RetiraEsteiraTabua(ImgTabua);
ImgTabua = RetiraExcessosTabua(ImgTabua); %figure,imshow(ImgTabua);
```

```
RetMontaLadoTabInteiras = ImgTabua;
```

---

```
function RetMontaLadoTabua = MontaLadoTabua
% Monta a imagem da tabua imageada a partir de suas imagens parciais
% Onde tem tabua, a media dos pixels eh entre 68 e 160,
% onde nao tem entre 7 e 19 (com 2 holofotes)
% Configuracao da camera: ISO800
```

```
global Tempo; QuantLins = 10; Limiar = 45; % Limiar a ser usado
ImgP1 = uint8(zeros(1024,1392,3)); ImgP6 = uint8(zeros(1024,1392,3));
ImgP2 = uint8(zeros(1024,1392,3)); ImgP7 = uint8(zeros(1024,1392,3));
ImgP3 = uint8(zeros(1024,1392,3)); ImgP8 = uint8(zeros(1024,1392,3));
ImgP4 = uint8(zeros(1024,1392,3)); ImgP9 = uint8(zeros(1024,1392,3));
ImgP5 = uint8(zeros(1024,1392,3));
```

```
ImgP1 = CapturaUmaImagem; [Lins,Cols,Prof] = size(ImgP1);
```

```
while sum(sum(sum(ImgP1(Lins-QuantLins:Lins,1:Cols,:)))/(QuantLins*Cols*Prof) < Limiar
    disp('Checando...'); ImgP1 = CapturaUmaImagem; end
pause(0.05);
```

```
disp('Pegando2. '); pause(Tempo); ImgP2 = CapturaUmaImagem;
disp('Pegando3. '); pause(Tempo); ImgP3 = CapturaUmaImagem;
disp('Pegando4. '); pause(Tempo); ImgP4 = CapturaUmaImagem;
disp('Pegando5. '); pause(Tempo); ImgP5 = CapturaUmaImagem;
disp('Pegando6. '); pause(Tempo); ImgP6 = CapturaUmaImagem;
disp('Pegando7. '); pause(Tempo); ImgP7 = CapturaUmaImagem;
disp('Pegando8. '); pause(Tempo); ImgP8 = CapturaUmaImagem;
disp('Pegando9. '); pause(Tempo); ImgP9 = CapturaUmaImagem;
```

```
Ini = 3; Fim = 1022;
ImgTemp = [ImgP1(Ini:Fim,:,:), ImgP2(Ini:Fim,:,:), ImgP3(Ini:Fim,:,:),
           ImgP4(Ini:Fim,:,:), ImgP5(Ini:Fim,:,:), ImgP6(Ini:Fim,:,:),
           ImgP7(Ini:Fim,:,:), ImgP8(Ini:Fim,:,:), ImgP9(Ini:Fim,:,:)];
```

```
RetMontaLadoTabua = ImgTemp;
```

---

```
function RetNomeClasseARACRUZ = NomeClasseARACRUZ(ClasseTabua)
```

```
switch ClasseTabua
    case 1; RetNomeClasseARACRUZ = 'FAS';
    case 2; RetNomeClasseARACRUZ = 'Select';
    case 3; RetNomeClasseARACRUZ = 'N1Common';
    case 4; RetNomeClasseARACRUZ = 'Classe1';
    case 5; RetNomeClasseARACRUZ = 'Exclusao';
```

end

---

```
function RetObtemBloco = ObtemBloco(LinInit,ColInit);
% Obtem as informacoes do bloco a partir da tabua

global LTAtual;

RetObtemBloco = LTAtual(LinInit:LinInit+63,ColInit:ColInit+63,:);
```

---

```
function [ImgSemEsteira] = RetiraEsteiraParte (ICE)
% Retorna a imagem da tabua sem a esteira de fundo
0

[Lins,Cols,Prof] = size(ICE); M = 15; % Limiar a ser usado

Li = 1; % Li = primeira linha contendo M > limiar
while sum(sum(ICE(Li,1:Cols,:)))/(3*Cols) < M
    Li = Li + 1; end
Lf = Lins; % Lf = ultima linha contendo M > limiar
while sum(sum(ICE(Lf,1:Cols,:)))/(3*Cols) < M
    Lf = Lf - 1; end

% Mudar para 1 se necessario
Ci = 200; % Ci = primeira coluna contendo M > limiar
while (sum(sum(ICE(Li:Li+fix((Lf-Li)/2),Ci,:)))/(3*(Lf-Li)/2) < M) | (sum(sum(ICE(Li+fix((Lf-Li)/2):Lf,Ci,:)))/(3*(Lf-Li)/2) < M)
    Ci = Ci + 1; end
% Mudar para 1 Colse necessario
Cf = 1250; % Cf = ultima coluna contendo M > limiar
while (sum(sum(ICE(Li:Li+fix((Lf-Li)/2),Cf,:)))/(3*(Lf-Li)/2) < M) | (sum(sum(ICE(Li+fix((Lf-Li)/2):Lf,Cf,:)))/(3*(Lf-Li)/2) < M)
    Cf = Cf - 1; end

% Corta a esteira da imagem e retorna-a como parametro
ImgSemEsteira = imcrop(ICE,[Ci,Li,Cf-Ci,Lf-Li]);
```

---

```
function [ImgTabuaSemEsteira] = RetiraEsteiraTabuaJai (ImgComEsteira)
% Retorna a imagem da tabua JAI sem a esteira de fundo
% Na esteira a media da soma dos pixels fica em torno de 18
% Na tabua a media da soma dos pixels fica entre 43 e 68
% Sera usado o limiar no valor de 35 por causa
% da ocorrencia de esmoado nas beiradas

global Limiar;

[Lins,Cols,Prof] = size(ImgComEsteira);
% Limiar a ser usado para imagem de tabua da camera Jai

Li = 25; % Li = primeira linha de busca
while sum(sum(ImgComEsteira(Li,:)))/(3*Cols) < Limiar Li = Li + 1; end
Lf = Lins - 5; % Lf = ultima linha de busca
while sum(sum(ImgComEsteira(Lf,:)))/(3*Cols) < Limiar Lf = Lf - 1; end
```

```

Lmt = 500; Ci = 5; % Ci = primeira coluna
MeioInf = round((Lf/2) - Lmt); MeioSup = round((Lf/2) + Lmt);

while (sum(sum(ImgComEsteira(Li:Li+Lmt,Ci,:)))/(3*Lmt) < Limiar) |...
    (sum(sum(ImgComEsteira(MeioInf:MeioSup,Ci,:)))/(3*Lmt*2) < Limiar) |...
    (sum(sum(ImgComEsteira(Lf-Lmt:Lf,Ci,:)))/(3*Lmt) < Limiar)
    Ci = Ci + 1; end

Cf = Cols - 5; % Cf = ultima coluna
while (sum(sum(ImgComEsteira(Li:Li+Lmt,Cf,:)))/(3*Lmt) < Limiar) |...
    (sum(sum(ImgComEsteira(MeioInf:MeioSup,Cf,:)))/(3*Lmt*2) < Limiar) |...
    (sum(sum(ImgComEsteira(Lf-Lmt:Lf,Cf,:)))/(3*Lmt) < Limiar)
    Cf = Cf - 1; end

ImgTabuaSemEsteira = ImgComEsteira(Li:Lf,Ci:Cf,:);

```

---

```

function [ImgSemExcessos] = RetiraExcessosTabua (ImgComExcessos)
% Retorna a imagem da tabua multipla de blocos de 64 pixels
% Corta o Exc igualmente das duas bordas

Li = 1; Ci = 1; [Lf,Cf,Prof] = size(ImgComExcessos);

if mod(Lf,64) == 0 Li = Li + 32; Lf = Lf - 32;
else
    if mod(Lf,2) == 1 Lf = Lf - 1; end;
    if mod(Lf,64) == 0 Li = Li + 32; Lf = Lf - 32;
    else % Excesso no comprimento a ser cortado dos 2 lados
        Excesso = mod(Lf,64)/2; Li = Li + Excesso; Lf = Lf - Excesso; % Corta igual
    end
end

if mod(Cf,64) == 0 Ci = Ci + 32; Cf = Cf - 32;
else
    if mod(Cf,2) == 1 Cf = Cf - 1; end;
    if mod(Cf,64) == 0 Ci = Ci + 32; Cf = Cf - 32;
    else
        % Excesso = mod(Cf,64); % Excesso na largura a ser cortado dos 2 lados
        % if sum(sum(ImgComExcessos(:,Ci,:)) < sum(sum(ImgComExcessos(:,Cf,:)))
        % Ci = Ci + Excesso;
        % else Cf = Cf - Excesso; end % Corta inteligente
        Excesso = mod(Cf,64)/2; Ci = Ci + Excesso; Cf = Cf - Excesso; % Corta igual
    end
end

% Corta os excessos da imagem e retorna-a como parametro
ImgSemExcessos = ImgComExcessos(Li:Lf,Ci:Cf,:);

```

---

```

function RetornoRetornaCarsAUsar = RetornaCarsAUsar (VetCarsOriginal)
% Retorna as Caracteristicas a serem usadas

global NumVOs VOsaUsar;
VetCarsTemp = zeros(1,NumVOs); Cont = 1;

```

```

for i = 1:45
    if VOUsAUsar(i) == 1
        VetCarsTemp(Cont) = VetCarsOriginal(i); Cont = Cont + 1;
    end;
end

```

```
RetornoRetornaCarsAUsar = VetCarsTemp;
```

---

```
function RetornoRetornaCarsNVO = RetornaCarsNVO (Bloco)
```

```
% Retorna o vetor de caracteristicas do Bloco
```

```
global NumVOs VOUsAUsar;
```

```
CarsTemp = double(zeros(1,45));
```

```
BandaR = double(Bloco(:,1)); BandaR = BandaR(:); % Torna bandas um vetor
```

```
BandaG = double(Bloco(:,2)); BandaG = BandaG(:);
```

```
BandaB = double(Bloco(:,3)); BandaB = BandaB(:);
```

```
% Calcula as Caracteristicas para BandaR
```

```
CarsTemp(01) = prctile(BandaR,[02]); CarsTemp(07) = prctile(BandaR,[50]);
```

```
CarsTemp(02) = prctile(BandaR,[05]); CarsTemp(08) = prctile(BandaR,[60]);
```

```
CarsTemp(03) = prctile(BandaR,[10]); CarsTemp(09) = prctile(BandaR,[70]);
```

```
CarsTemp(04) = prctile(BandaR,[20]); CarsTemp(10) = prctile(BandaR,[80]);
```

```
CarsTemp(05) = prctile(BandaR,[30]); CarsTemp(11) = prctile(BandaR,[90]);
```

```
CarsTemp(06) = prctile(BandaR,[40]); CarsTemp(12) = prctile(BandaR,[95]);
```

```
% Calcula as Caracteristicas para BandaG
```

```
CarsTemp(13) = prctile(BandaG,[02]); CarsTemp(19) = prctile(BandaG,[50]);
```

```
CarsTemp(14) = prctile(BandaG,[05]); CarsTemp(20) = prctile(BandaG,[60]);
```

```
CarsTemp(15) = prctile(BandaG,[10]); CarsTemp(21) = prctile(BandaG,[70]);
```

```
CarsTemp(16) = prctile(BandaG,[20]); CarsTemp(22) = prctile(BandaG,[80]);
```

```
CarsTemp(17) = prctile(BandaG,[30]); CarsTemp(23) = prctile(BandaG,[90]);
```

```
CarsTemp(18) = prctile(BandaG,[40]); CarsTemp(24) = prctile(BandaG,[95]);
```

```
% Calcula as Caracteristicas para BandaB
```

```
CarsTemp(25) = prctile(BandaB,[02]); CarsTemp(31) = prctile(BandaB,[50]);
```

```
CarsTemp(26) = prctile(BandaB,[05]); CarsTemp(32) = prctile(BandaB,[60]);
```

```
CarsTemp(27) = prctile(BandaB,[10]); CarsTemp(33) = prctile(BandaB,[70]);
```

```
CarsTemp(28) = prctile(BandaB,[20]); CarsTemp(34) = prctile(BandaB,[80]);
```

```
CarsTemp(29) = prctile(BandaB,[30]); CarsTemp(35) = prctile(BandaB,[90]);
```

```
CarsTemp(30) = prctile(BandaB,[40]); CarsTemp(36) = prctile(BandaB,[95]);
```

```
% Realiza as Normalizacoes
```

```
if (CarsTemp(12)-CarsTemp(01)) ~= 0
```

```
    CarsTemp(01:12) = CarsTemp(01:12)/(CarsTemp(12)-CarsTemp(01)); end;
```

```
if (CarsTemp(24)-CarsTemp(13)) ~= 0
```

```
    CarsTemp(13:24) = CarsTemp(13:24)/(CarsTemp(24)-CarsTemp(13)); end;
```

```
if (CarsTemp(36)-CarsTemp(25)) ~= 0
```

```
    CarsTemp(25:36) = CarsTemp(25:36)/(CarsTemp(36)-CarsTemp(25)); end;
```

```
[Lins,Cols,Prof] = size(Bloco);
```

```
Temp = Bloco(:,1); Temp = Temp(:); % Calcula os valores para banda R
```

```
CarsTemp(37) = sum(sum(Bloco(:,1)))/(Lins*Cols);
```

```
CarsTemp(38) = median(double(Temp));
```

```
CarsTemp(39) = std(double(Temp));
```

```

Temp = Bloco(:,2); Temp = Temp(:); % Calcula os valores para banda G
CarsTemp(40) = sum(sum(Bloco(:,2)))/(Lins*Cols);
CarsTemp(41) = median(double(Temp));
CarsTemp(42) = std(double(Temp));

Temp = Bloco(:,3); Temp = Temp(:); % Calcula os valores para banda B
CarsTemp(43) = sum(sum(Bloco(:,3)))/(Lins*Cols);
CarsTemp(44) = median(double(Temp));
CarsTemp(45) = std(double(Temp));

Cars = double(zeros(1,NumVOs)); ContNumVOs = 1;
for Cont45 = 1:45
    if VOUsar(Cont45) == 1
        Cars(ContNumVOs) = CarsTemp(Cont45); ContNumVOs = ContNumVOs + 1;
    end
end

% Retorna os Percentis normalizados por banda, e as caracteristicas adicionais
RetornaCarsNVO = Cars;

```

---

```

function RetVetorDeVOUsar = VetorDeVOUsar;
% Retorna apenas as VOs que serao usadas dentre as 45

global NumVOs VOUsar DadosBlocosTabuas;

VetorTemp = double(zeros(31672,NumVOs+6)); ContTemp = 5;
VetorTemp(:,1:4) = DadosBlocosTabuas(:,1:4); VetorTemp(:,NumVOs+5:NumVOs+6) =
DadosBlocosTabuas(:,50:51);
for Cont45 = 1:45
    if VOUsar(Cont45) == 1
        VetorTemp(:,ContTemp) = DadosBlocosTabuas(:,Cont45+4);
        ContTemp = ContTemp + 1;
    end
end
RetVetorDeVOUsar = VetorTemp;

```

---

```

function RetCalculaIndiceKappa = CalculaIndiceKappa (MatrizConfusao)
% Calcula o Indice Kappa da matriz de confusao: MatrizConfusao.

% Obtencao de informacoes da matriz de confusao
NumTotalAmostras = sum(sum(MatrizConfusao));
[NumTotalClasses, Numerador1] = size(MatrizConfusao);

% Determina os componentes do numerador e denominador para calculo do Kappa
Numerador1 = NumTotalAmostras * sum(diag(MatrizConfusao));
Numerador2 = sum(MatrizConfusao) * (sum(MatrizConfusao'));
Denominador = NumTotalAmostras^2 - Numerador2;

% Calcula o Kappa e o retorna
RetCalculaIndiceKappa = (Numerador1 - Numerador2)/Denominador;

```

---

```

function GeraMatrizConfusao = GeraMatrizConfusao(NumItens)
% Gera a matriz de confusão

global RegraClassif ResultClassif;
ClasCorr = zeros(1,NumItens);

if RegraClassif == 'ABNT '
    ClasCorr = ClassifCorretaABNT_ARACRUZ(1); MatConf = zeros(6,6);
elseif RegraClassif == 'ARACRUZ'
    ClasCorr = ClassifCorretaABNT_ARACRUZ(2); MatConf = zeros(5,5);
else % Blocos
    ClasCorr = ClassifCorretaABNT_ARACRUZ(3); MatConf = zeros(6,6);
end

for Cont = 1:NumItens
    if ResultClassif(Cont) == ClasCorr(Cont)
        MatConf(Cont,Cont) = MatConf(Cont,Cont) + 1;
    else
        MatConf(ResultClassif(Cont),ClasCorr(Cont)) = MatConf(ResultClassif(Cont),ClasCorr(Cont)) + 1;
    end
end

GeraMatrizConfusao = MatConf;

```

---

```

function [NovoLi,NovoCi] = CentralizaDefeito (Li,Ci)
% Centraliza o defeito no bloco para a cercagem com cor ficar bem vistosa

global LTAtualLF;

Lf = Li+63; Cf = Ci+63;

if sum(sum(~LTAtualLF(Li:Lf,Ci:Cf))) > 0
    if Ci ~= 1 % Centralizando em relacao as colunas
        ContColsFolga = 0; CfTemp = Cf;
        while (sum(~LTAtualLF(Li:Lf,CfTemp)) == 0) & (CfTemp > 64)
            ContColsFolga = ContColsFolga + 1; CfTemp = CfTemp - 1; end
        if ContColsFolga > 1
            ContColsFolga = floor(ContColsFolga/2);
            Ci = Ci - ContColsFolga; Cf = Cf - ContColsFolga; end
        end
    end

    if Li ~= 1 % Centralizando em relacao as linhas
        ContLinsFolga = 0; LfTemp = Lf;
        while (sum(~LTAtualLF(LfTemp,Ci:Cf)) == 0) & (LfTemp > 64)
            ContLinsFolga = ContLinsFolga + 1; LfTemp = LfTemp - 1; end
        if ContLinsFolga > 1
            ContLinsFolga = floor(ContLinsFolga/2);
            Li = Li - ContLinsFolga; Lf = Lf - ContLinsFolga; end
        end
    end
end

NovoLi = Li; NovoCi = Ci;

```

---

```

function RetornoDialogo = Dialogo(NumDialogo);

```

```

% Mostra a mensagem correta no momento adequado

global LTAAtual LTAAtualAux Mensagem ClassifFinalTab ClasseTab ModoClassif;

switch NumDialogo
case 1
    RetornoDialogo = questdlg('Qual Norma de Classificacao deseja usar?','SELECAO DA NORMA DE
CLASSIFICACAO:','ABNT ','COMERCIAL','Sair','Sair');
case 2
    RetornoDialogo = questdlg('Qual Modo de Classificacao deseja usar?','SELECAO DO MODO DE
CLASSIFICACAO:','Interativo','Automatico','Interativo');
case 3
    questdlg('Clique <Ok> AGORA, depois coloque a Tabua na esteira e AGUARDE...','LEITURA DA
PRIMEIRA FACE:','Ok','Ok');
case 4
    subplot(2,1,1), subimage(imrotate(LTAAtual,90)); set(gcf,'Position',[5 40 1017 688]);
    title(['Primeira face da Tabua:']);
    set(gcf,'Resize','Off'); set(gcf,'MenuBar','None'); set(gcf,'NumberTitle','Off'); set(gcf,'Name','Leitura Parcial
da Tabua:');
    if ModoClassif == 'Interativo'
        questdlg('Face 1 escaneada. Clique <Ok> AGORA, depois vire a Tabua na esteira e
AGUARDE...','LEITURA DA SEGUNDA FACE:','Ok','Ok');
    end
case 5
    subplot(2,1,2), subimage(imrotate(LTAAtual,90));
    title(['Segunda face da Tabua:']); xlabel(Mensagem);
    set(gcf,'Name',['CLASSIFICACAO FINAL DA TABUA = <<<< Classe ',num2str(ClassifFinalTab),' >>>>']);
case 6
    subplot(2,1,2), subimage(imrotate(LTAAtual,90));
    title(['Segunda face da Tabua:']); xlabel(Mensagem);
    set(gcf,'Name',['CLASSIFICACAO FINAL DA TABUA = <<<< ',NomeClasseARACRUZ(ClasseTab),'
>>>>']);
end

```

---

```

function RetPadronizacao = Padronizacao(MatDados)
% Retorna padronizadas as variaveis contidas em MatDados
% Padronizadas --> desvio padrao = 1

[NumLinsMD,NumColsMD] = size(MatDados);
VarsPadronizadas = zeros(size(MatDados));
VarsMatDados = var(MatDados);

for i = 1:NumLinsMD
    for j = 1:NumColsMD
        VarsPadronizadas(i,j) = MatDados(i,j)/sqrt(VarsMatDados(j));
    end; end;

RetPadronizacao = VarsPadronizadas;

```

---

```

function RetornoAchaColProxDefeito = AchaColProxDefeito(LinInit,ColInit);
% Acha a proxima coluna, a partir de LinInit na tabua limiarizada, que contem um pixel preto

global LTAAtualLF NumColsTab;

```

```
while (sum(~LTAtualLF(LinInit:LinInit+63,ColInit)) == 0) & (ColInit < NumColsTab)
    ColInit = ColInit + 1; end
```

```
if ColInit+63 > NumColsTab
    ColInit = NumColsTab - 63; end
```

```
RetornoAchaColProxDefeito = ColInit;
```

---

```
function RetornoAchaLinProxDefeito = AchaLinProxDefeito(LinInit);
% Acha a proxima linha, a partir de LinInit na tabua limiarizada, que contem um pixel preto
```

```
global LTAtualLF NumLinsTab;
```

```
while (sum(~LTAtualLF(LinInit,:)) == 0) & (LinInit+63 < NumLinsTab)
    LinInit = LinInit + 1; end;
```

```
if LinInit+63 > NumLinsTab
    LinInit = NumLinsTab - 63; end
```

```
RetornoAchaLinProxDefeito = LinInit;
```

---

```
function RetEmPes = EmPes (Valor)
% Transforma um Valor em centimetros para pes e o retorna
```

```
RetEmPes = Valor/30.48;
```

---

```
function RetEmPolegadas = EmPolegadas (Valor)
% Transforma um Valor em centimetros para polegadas e o retorna
```

```
RetEmPolegadas = Valor/2.54;
```

---

```
function [NovoVar1,NovoVar2] = TrocaValores (Var1,Var2)
% Troca o valor de duas variaveis. Retorna os valores em NovoVar1 e NovoVar2
```

```
NovoVar1 = Var2; NovoVar2 = Var1;
```