

**UNIVERSIDADE NOVE DE JULHO - UNINOVE**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO**

**TÉCNICAS DE VISÃO COMPUTACIONAL APLICADAS NA AUTOMAÇÃO DE  
PROCESSOS METROLÓGICOS**

PETERSON ADRIANO BELAN

**SÃO PAULO**

**2012**

**PETERSON ADRIANO BELAN**

**TÉCNICAS DE VISÃO COMPUTACIONAL APLICADAS NA AUTOMAÇÃO DE  
PROCESSOS METROLÓGICOS**

Dissertação apresentada ao Programa de Mestrado em Engenharia de Produção da Universidade Nove de Julho - Uninove, como requisito parcial para a obtenção do grau de Mestre em Engenharia de Produção.

Prof. André Felipe Henriques Librantz, Dr. – Orientador, UNINOVE

**SÃO PAULO**

**2012**

# **TÉCNICAS DE VISÃO COMPUTACIONAL APLICADAS NA AUTOMAÇÃO DE PROCESSOS METROLÓGICOS**

Por

**PETERSON ADRIANO BELAN**

Dissertação apresentada ao Programa de Mestrado em Engenharia de Produção da Universidade Nove de Julho - Uninove, como requisito parcial para a obtenção do grau de Mestre em Engenharia de Produção, pela Banca Examinadora, formada por:

---

Presidente: Prof. André Felipe Henriques Librantz, Dr. - Orientador, UNINOVE

---

Membro: Prof. Sérgio Vicente Denser Pamboukian, Dr, Mackenzie

---

Membro: Prof. Cléber Gustavo Dias, Dr., UNINOVE

São Paulo, 13 de agosto de 2012

Dedico este trabalho ao meu filho  
Guilherme pelos momentos ausentes nesta  
fase tão importante de sua vida.

# AGRADECIMENTOS

Gostaria de agradecer inicialmente a Deus, que me guiou até este momento.

Ao Prof. Dr. André Felipe Henriques Librantz, que me orientou neste projeto de minha vida, com muita paciência e compreensão, principalmente nos momentos de maior dificuldade.

Ao Prof. Dr. Sidnei Alves de Araújo, meu co-orientador e uma das pessoas responsáveis pela minha realização deste mestrado, me convidando, apoiando e ajudando em tudo que foi possível, além de ser um grande amigo o qual gosto e respeito muito.

À minha esposa Lygia e meu filho Guilherme pela paciência que tiveram neste ano e meio, pelos finais de semana perdidos e principalmente pelos momentos que tive de me ausentar nesta fase tão importante de nossas vidas.

Aos meus pais, Danilo e Sônia, por todo o incentivo prestado em todos os momentos.

A todos os professores do Programa de Engenharia de Produção da UNINOVE em especial os quais tive o prazer de assistir as disciplinas, passando-me valiosos conhecimentos.

A todos os amigos que adquiri neste período de estudo, que me ajudaram e apoiaram nos momentos em que se fez necessário.

Ao Prof. Cleber Gustavo Dias e ao Prof. Sérgio Vicente Denser Pamboukian que participaram de minha banca fornecendo-me valiosas informações, ajudando a refinar esse trabalho e que estiveram sempre prontos aos chamados.

E finalmente a CAPES que me concedeu uma bolsa de estudos, a qual foi de grande valia para que pudesse ter toda a dedicação prestada.

## RESUMO

Um problema comum encontrado nos laboratórios de calibração é a confiabilidade dos resultados obtidos na calibração de instrumentos de medição, especialmente quando eles não possuem interface de comunicação. Neste caso o consumo de tempo e os erros humanos cometidos são problemas enfrentados por vários desses laboratórios. Neste contexto, muitas abordagens baseadas em visão de computador têm sido propostas na literatura para automatizar processos de calibração. Desta forma, neste trabalho foram propostas quatro abordagens de visão computacional para a área de metrologia, sendo três delas visando a calibração de instrumentos e uma visando a inspeção da qualidade de moldes para injeção de plástico. O primeiro algoritmo é baseado em correlação cruzada normalizada para leitura de dígitos do display de manômetro digital, o segundo utiliza projeções radiais juntamente com algoritmo de Bresenham para determinar a posição do ponteiro em relógios comparadores analógicos, o terceiro efetua a segmentação da imagem para determinar o tamanho dos furos de peneiras grano-volumétrica. Por fim o quarto algoritmo envolve a automatização do processo de inspeção de moldes para injeção de plástico por meio de um equipamento laser para ajudar a detectar problemas, tais como excesso de polimento ("casca de laranja") ou riscos. Os resultados experimentais mostraram que os quatro algoritmos propostos apresentaram boa precisão e bom desempenho de processamento, tornando-se assim viáveis para as tarefas metrológicas sugeridas.

**Palavras-chave:** Visão computacional, calibração, instrumentos de medição, correlação cruzada normalizada, projeção radial, Gabor, molde de injeção.

## ABSTRACT

A common problem found in the calibration laboratories is the reliability of the results obtained in calibration of instruments, especially when they do not have a built-in communication interface. In this case, the spending time is increased significantly and the calibration can be subject to human error. In this context, many approaches based on computer vision have been proposed in the literature for automating calibration processes. Thus, four approaches of computer vision for application to the area of metrology were proposed in this work, three of them aiming at the calibration of instruments and one at the evaluation of quality injection mould surface. The first algorithm is based on normalized cross-correlation for reading digits of the display of a digital manometer; the second uses radial projections with Bresenham algorithm to determine the position of the pointer in analogical dial gages; the third makes a segmentation of the image to determine the holes size of screens grano-volumetric; and finally the fourth algorithm involves the automation of the evaluation of injection mould surface in which the proposed algorithm applied to images generated by means of laser device could help to detect problems such as over-polishing ("orange peel") or risks. Results showed that the proposed algorithms presented high accuracy and performance and can be used in real time systems applied to calibration processes.

**Key-words:** Computer vision, calibration, measurement instruments, normalized cross-correlation, radial projection, Gabor, injection mould.

# LISTA DE ILUSTRAÇÕES

Figura 1 – Diagrama dos processos metrológicos.....	15
Figura 2 – Exemplo de um molde para injeção de plástico.....	19
Figura 3 – Passos fundamentais em processamento de imagens digitais. ....	25
Figura 4 – Resultado de NCC entre duas imagens. ....	28
Figura 5 – NCC entre duas imagens. ....	29
Figura 6 – Exemplo de conectividade e vizinhança de pixels.....	32
Figura 7 – Exemplos de funções de Gabor. ....	32
Figura 8 – Imagem sintética com valores de intensidade variando senoidalmente na direção horizontal. ....	33
Figura 9 – Imagens de intensidade das partes real e imaginária de um determinado Kernel do filtro de Gabor.....	33
Figura 10 – Parte de uma impressão digital real com orientação aproximada de 135º.....	34
Figura 11 – Parte de uma impressão digital real com orientação aproximada de 135º, mas com o valor de frequência errado.....	34
Figura 12 – Imagem após a aplicação do Filtro de Gabor usando os valores $\theta = 3\pi/4$ e $f = 4$ .....	36
Figura 13 – Exemplo de componente principal. Linha vermelha mostra a distribuição principal dos dados e a linha azul mostra a componente secundária.....	38
Figura 14 – Exemplo de aplicação de ACP.....	42
Figura 15 – Modelo gráfico de um Neurônio Artificial.....	44
Figura 16 – RNA alimentada adiante sem camada oculta. ....	47
Figura 17 – RNA alimentada adiante com uma camada oculta. ....	47
Figura 18 – Representação de uma realimentação. ....	47
Figura 19 – Representação de uma rede recorrente. ....	48
Figura 20 – Representação de um modelo baseado na correção de erro. ....	50
Figura 21 – Representação de um modelo baseado em Memória. ....	50
Figura 22 – Representação de um modelo baseado em competição.....	51
Figura 23 – Representação de um modelo Percéptron de Camada Única. ....	53
Figura 24 – Representação de um modelo Percéptron de Múltiplas Camadas.....	54
Figura 25 – Manômetro digital Zurich.....	59
Figura 26 – Relógios comparadores. ....	60
Figura 27 – Peneira grão-volumétrica.....	60
Figura 28 – Exemplo do display utilizado para reconhecimento dos dígitos.....	63
Figura 29 – Exemplo de um quadro dos vídeos capturados para os experimentos. ....	64
Figura 30 – Exemplos de imagens adquiridas de peneiras grão-volumétrica.....	65
Figura 31 – Régua graduada padrão. ....	65
Figura 32 – Sub-amostragem das imagens adquiridas dos moldes. ....	67
Figura 33 – Exemplo de geração de <i>speckle</i> .....	66
Figura 34 – Protótipo calibrador de relógio comparador. ....	68
Figura 35 – Protótipo calibrador de peneira grão-volumétrica. ....	69
Figura 36 – Projeto da caixa de proteção com dispositivos.....	70
Figura 37 – Mesa de deslocamento XY. ....	71

Figura 38 – Dispositivo em funcionamento. ....	71
Figura 39 – Diagrama do processo de calibração para manômetro digital. ....	72
Figura 40 – Processo de determinação da RI. ....	73
Figura 41 – Detecção das sub-regiões. ....	74
Figura 42 – Modelo dos dígitos utilizados para o reconhecimento. ....	75
Figura 43 – Diagrama do processo de calibração para relógio comparador analógico. ....	77
Figura 44 – Exemplo da metodologia proposta utilizando três valores de $\alpha$ ( $0^0$ , $90^0$ e $257^0$ ), $\lambda=150$ , $x_c=175$ e $y_c=175$ . O tamanho da imagem é $350 \times 350$ pixels. ....	78
Figura 45 – Diagrama do processo de calibração para relógio comparador analógico. ....	79
Figura 46 – Etapas do pré-processamento da imagem adquirida da trama da peneira grano-volumétrica. ....	80
Figura 47 – Diagrama da proposta para avaliação de moldes para injeção de plástico. ....	82
Figura 48 – Exemplo de segmentação por análise de textura. ....	83
Figura 49 – Exemplo da determinação da RI. ....	84
Figura 50 – Exemplos de imagens adquiridas em regiões diferentes da superfície do molde. ....	86
Figura 51 – Exemplo da metodologia proposta com a abordagem de visão computacional com a medida de circularidade proposta. ....	87
Figura 52 – Exemplo de imagens afetadas. ....	88
Figura 53 – Resultado do algoritmo de detecção de ponteiro. ....	90
Figura 54 – Histograma de frequência dos valores lidos ....	91
Figura 55 – Tela com resultado de saída do sistema ....	92
Figura 56 – Representação da superfície do molde 1 gerado pela metodologia proposta. O eixo X e Y estão representados em mm e o eixo Y é a medida de circularidade. ....	94
Figura 57 – Representação da superfície do molde 2 gerada pela metodologia proposta. O eixo X e Y estão representados em mm e o eixo Y é a medida de circularidade. ....	95
Figura 58 – Representação da superfície do molde (exemplo 3) gerada pela metodologia proposta. O eixo X e Y estão representados em mm e o eixo Y é a medida de circularidade. ....	95
Figura 59 – Exemplo de classificação da circularidade proposta no molde 2. ....	96

## LISTA DE TABELAS

Tabela 1 – Rugosidade superficial típica dos diversos processos de fabricação .....	20
Tabela 2 – Características dos moldes para injeção de plástico.....	61
Tabela 3 – Distribuição dos dígitos no conjunto de imagens .....	63
Tabela 4 – Matriz P calculada no reconhecimento do display da Figura 19. Os valores de correlação variam de -1 a 1, porém foram normalizados para o intervalo de 0 a 255 para visualização .....	76
Tabela 5 – Amostragem da medida da circularidade proposta (CP) na superfície dos cinco moldes. .	93

## LISTA DE ABREVIATURAS E SIGLAS

ACP – Análise de Componentes Principais

CMYK– Cyan, Magenta, Yellow e Black

EASM – Equipamento de Análise de Superfície de Molde

FPS – Frames Por Segundos

INMETRO – Instituto Nacional de Metrologia, Normalização e Qualidade Industrial

MLP – Multi-Layer Perceptron

NCC –*Normalized Cross Correlation* (Correlação Cruzada Normalizada)

Ra - Média aritmética de rugosidade

RGB–Red, Green and Blue

RI – Região de Interesse

RNA – Rede Neural Artificial

# SUMÁRIO

1	INTRODUÇÃO .....	14
1.1	CONSIDERAÇÕES INICIAIS .....	14
1.2	REVISÃO DA LITERATURA.....	15
1.3	FUNDAMENTAÇÃO TEÓRICA .....	18
1.3.1	Metrologia .....	18
1.4	JUSTIFICATIVA .....	20
1.5	OBJETIVOS .....	21
1.6	MÉTODOS DE TÉCNICAS DE PESQUISA .....	22
1.7	PUBLICAÇÕES RESULTANTES DA DISSERTAÇÃO .....	22
1.8	ESTRUTURA DO TRABALHO .....	23
2	VISÃO COMPUTACIONAL.....	24
2.1	CASAMENTO DE PADRÃO COM CORRELAÇÃO CRUZADA NORMALIZADA .....	27
2.2	ALGORITMO DE DESENHO DE LINHA DE BRESENHAM .....	30
2.3	LIMIARIZAÇÃO.....	30
2.4	COMPONENTES CONEXOS.....	31
2.5	FILTROS DE GABOR .....	32
2.5.1	Descrição do filtro de Gabor .....	34
2.6	ANÁLISE DE COMPONENTES PRINCIPAIS (ACP) .....	38
2.6.1	Componentes principais .....	38
2.6.2	Matriz de Covariância .....	39
2.6.3	Autoespaços, autovetores e autovalores.....	41
2.6.4	Análise dos componentes principais .....	42
2.7	REDES NEURAIS ARTIFICIAIS.....	43
2.7.1	Modelo lógico de um neurônio.....	44
2.7.2	Modelo matemático de um neurônio .....	45
2.7.3	Forma de propagação do sinal em RNA .....	46
2.7.4	Treinamento de Redes Neurais Artificiais .....	48
2.7.5	Aprendizagem .....	49
2.7.6	Modelos de redes neurais.....	52
2.7.7	Algoritmo Back-Propagation .....	54
2.8	CIRCULARIDADE.....	57
3	MATERIAIS E MÉTODOS .....	59
3.1	EQUIPAMENTOS .....	59
3.2	MÉTODOS.....	62

3.3	BANCOS DE IMAGENS UTILIZADOS.....	63
3.3.1	Calibração de manômetro digital .....	63
3.3.2	Calibração de relógio comparador analógico .....	63
3.3.3	Calibração de peneira grão-volumétrica.....	64
3.3.4	Análise de superfície de moldes para injeção de plástico.....	65
3.4	PROTÓTIPOS DESENVOLVIDOS.....	67
3.4.1	Calibração de relógio comparador.....	67
3.4.2	Calibração de peneira grão-volumétrica.....	69
3.4.3	Aparato Laser .....	70
4	METODOLOGIAS PROPOSTAS .....	72
4.1	CALIBRAÇÃO DE MANÔMETRO DIGITAL.....	72
4.1.1	Determinação da RI .....	72
4.1.2	Reconhecimento dos dígitos.....	75
4.2	CALIBRAÇÃO DE RELÓGIO COMPARADOR ANALÓGICO .....	76
4.2.1	Algoritmo de detecção de ponteiro .....	77
4.3	CALIBRAÇÃO DE PENEIRA GRÃO-VOLUMÉTRICA .....	79
4.4	ANÁLISE DA SUPERFÍCIE DE MOLDES PARA INJEÇÃO DE PLÁSTICO .....	81
4.4.1	Segmentação por Análise de Texturas.....	82
4.4.2	Determinação da RI .....	83
4.4.3	Medida de circularidade proposta.....	84
4.4.4	Reconhecimento e interpretação.....	85
5	RESULTADOS E DISCUSSÃO .....	88
5.1	CALIBRAÇÃO DE MANÔMETRO DIGITAL.....	88
5.2	CALIBRAÇÃO DE RELÓGIO COMPARADOR ANALÓGICO .....	89
5.3	CALIBRAÇÃO DE PENEIRA GRÃO-VOLUMÉTRICA .....	91
5.4	ANÁLISE DA SUPERFÍCIE DE MOLDES PARA INJEÇÃO DE PLÁSTICO .....	92
6	CONCLUSÕES .....	98

# 1 INTRODUÇÃO

## 1.1 CONSIDERAÇÕES INICIAIS

Visão computacional pode ser definida como uma subárea de processamento de imagem que estuda o desenvolvimento de métodos e técnicas que permitam que um sistema de computador possa reconhecer objetos em imagens, imitando assim algumas capacidades da visão humana, dotando a máquina com a capacidade de descrever uma cena contida em uma imagem digital. Desta forma, a inteligência artificial tem um papel muito importante, auxiliando na interpretação dos resultados descritos pelo sistema de visão computacional. (GONZALES; WOODS, 2002; CONCI; AZEVEDO; LETA, 2008; PEDRINI; SCHWARTZ, 2007).

Vários autores vêm propondo sistemas baseados em visão computacional para solucionar problemas na área metrológica a fim de tornar mais rápido e evitar erros humanos (ALEGRIA; SERRA, 2000a; ALEGRIA; SERRA, 2000b; ANDRIA *et al.*, 2009; ARAÚJO ; BELAN; LIBRANTZ, 2011; BELAN; ARAÚJO; LIBRANTZ, 2012; FENG; ZHAO, 2008; HE *et al.*, 2007; HEMMING; LEHTO, 2002; HEMMING; LAGERLUND; LASSILA, 2007; VÁSQUEZ-FERNÁNDEZ *et al.*, 2009).

A calibração de instrumentos nos laboratórios metrológicos atualmente se depara com alguns problemas associados à produtividade em tarefas relacionadas ao processo de calibração e à fidelidade dos resultados obtidos durante a calibração para a transcrição do certificado de calibração.

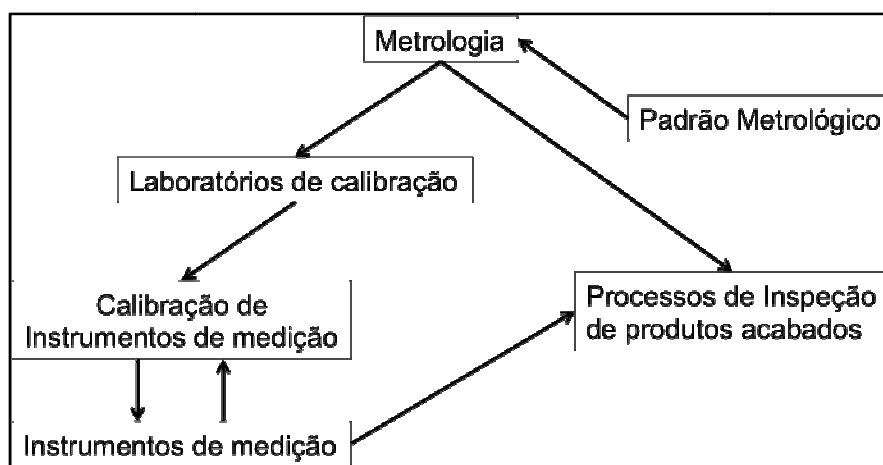
Outro problema frequente é o emprego destes instrumentos metrológicos para verificação da qualidade nos produtos finalizados, como por exemplo, a análise da superfície de moldes para injeção de plástico, problema abordado neste trabalho. Essa tarefa em muitos casos demanda tempo excessivo e devido à experiência requerida do usuário, torna esta parte do processo de fabricação do molde caro em relação ao custo total do produto (EDELSTAHL, 2009).

Neste contexto, este trabalho propõe metodologias que possuam certa aderência nos laboratórios de calibração para a realização da calibração de instrumentos de medição de forma automática. Para isto foram desenvolvidos

protótipos e aplicativos que tornam possível a realização destas tarefas de forma automática, o que difere de grande parte da literatura pesquisada. Também foi proposta uma metodologia para identificar e quantificar problemas causados no processo de polimento no molde e um dispositivo para a análise da qualidade da superfície.

Para explicar o fluxo dos processos metrológicos, foi criado um diagrama conforme mostrado na Figura 1.

Figura 1 – Diagrama dos processos metrológicos.



Fonte: Autor

## 1.2 REVISÃO DA LITERATURA

As calibrações de instrumentos de medição estão suscetíveis a erros principalmente pelo fato de que na maioria dos casos o técnico que executou a calibração não emite o certificado, apenas confere e assina após um digitador preenche-lo. Em alguns casos, o resultado expresso pode ser diferente do obtido durante a calibração do instrumento, ocasionado principalmente por falhas humanas, tais como: anotação errada do valor exibido no display do instrumento, erro de leitura por paralaxe ou cópia errada pelo digitador dos dados preenchidos pelo técnico (FRACASSO; SILVA; SELVATICI, 2010).

Desta forma os sistemas automatizados de calibração para dispositivos que não possuem uma interface de comunicação de dados, desempenham um papel muito importante na calibração de instrumentos de medição. Eles proporcionam

maior precisão, repetibilidade e redução de custos, além da diminuição de trabalhos monótonos e de tarefas complexas (ALEGRIA; SERRA, 2000; FRACASSO; SILVA; SELVATICI, 2010).

Após os equipamentos serem calibrados, os mesmos podem ser utilizados nas indústrias para verificação de qualidade dos produtos fabricados. Como por exemplo, o problema de análise da superfície dos moldes para injeção de plástico. Estes moldes possuem uma vasta diversificação de aplicações variando desde utensílios domésticos à indústria automotiva (MESQUITA; SCHNEIDER, 2010). No processo de fabricação de peças plásticas injetadas, a qualidade do molde é essencial, visto que todos os detalhes presentes no molde são destacados na peça finalizada, assim deve-se prestar atenção nestes detalhes, pois estes podem ter sido gerados propositalmente ou serem falhas no molde (YAN *et al.*, 2007; SANKAR; JAIN; RAMKUMAR, 2010; BEALL, 2000; SHIMIZU; FUJI, 2003).

Com base nestes problemas metrológicos citados, aplicações baseadas em visão computacional tentam melhorar a qualidade do processo. Na prática estas aplicações são normalmente implementadas a fim de reduzir custos, garantir um processo com mais qualidade e principalmente, aumentar a confiabilidade no processo de inspeção mecânica ou visual.

Nos últimos anos observa-se um forte crescimento nas aplicações baseadas em visão computacional. Este aumento tem sido influenciado principalmente pelos baixos custos de desenvolvimento e dos equipamentos para a implementação de sistemas complexos e robustos (ARAÚJO; KIM, 2011). Neste contexto, diversos autores vêm propondo sistemas de calibração automática empregando técnicas de visão computacional.

ALEGRIA *et al.* (2000) desenvolveram um sistema de calibração de instrumentos digitais para determinar de forma automática a leitura dos dígitos no display do instrumento, para esta tarefa utilizou um algoritmo baseado nos quadrantes do dígito, dividindo o mesmo em 15 partes para a análise. Outra aplicação desenvolvida pelos autores é para determinar a posição do ponteiro para instrumentos analógicos. Nessa tarefa foi utilizado um algoritmo baseado em transformadas de *Hough*. De acordo com os resultados apresentados pelos autores as aplicações apresentaram falhas quando testadas em condições de iluminação alteradas, o que pode gerar inconfiabilidade nos dados das calibrações executadas.

ANDRIA *et al.* (2009), desenvolveram um sistema para reconhecimento dos dígitos do display baseado em OCR (*Optical Character Recognition*). Nesse caso, não foram reportadas falhas de reconhecimento, apenas avalia o método quanto a sua funcionalidade.

VÁSQUEZ-FERNANDES *et al.* (2009) desenvolveram sistemas para leitura e conversão dos dados numéricos do display do instrumento de medição utilizando dois tipos de câmeras: uma webcam normal e uma câmera de vídeo CCD com conversor para o computador, o que apresentou uma ligeira diferença no resultado final de processamento, uma vez que as imagens adquiridas por meio da câmera CCD apresentaram melhores resultados para o algoritmo proposto..

FRACASSO, SILVA, SELVATICI (2010) desenvolveram uma bancada inteligente para calibração de hidrômetros onde os dígitos e as posições dos ponteiros são detectadas por um sistema de visão computacional. Nestas tarefas foi empregado o uso de algum algoritmo baseado em correlação, e para a determinação da posição dos ponteiros foi desenvolvido um sistema que detecta o ângulo de suas bordas.

HEMMING e LEHTO (2002) propuseram um sistema de visão para a calibração de instrumentos com ponteiro, para determinação da posição do ponteiro foi utilizada uma técnica para destacar a diferença entre duas imagens. Neste processo de subtração, a imagem resultante é a do ponteiro, visto que a imagem de referência é de um equipamento do mesmo modelo. A partir da imagem extraída é realizado um processo de detecção de bordas para determinar o ângulo do ponteiro.

HEMMING *et al.* (2007) desenvolveram um sistema para calibração automática de micrômetros, também utilizando uma câmera USB. Nesse caso o foco foi o desenvolvimento de um dispositivo que desloca o fuso do micrômetro e posiciona os blocos padrão para realizar as medidas. Com o bloco posicionado e o micrômetro ajustado é realizada a leitura do traço do micrômetro. Pelos resultados apresentados o sistema proposto não apresentou falhas e possui total capacidade para ser implantado em laboratórios de calibração.

HE *et al.* (2007) desenvolveram um sistema para leitura de instrumentos de ponteiro. Neste caso foi utilizada a técnica de transformada de *Hough* para detecção da posição do ponteiro.

Dentre as referências consultadas, apenas em HEMMING *et al.* (2007) foi proposto um sistema com a integração da automação. Nos demais casos apenas as técnicas são apresentadas, normalmente com uso de um aparato para facilitar a execução dos experimentos, porém todo o processo de mudança dos valores para leitura nos equipamentos é realizado de forma manual.

### 1.3 FUNDAMENTAÇÃO TEÓRICA

#### 1.3.1 Metrologia

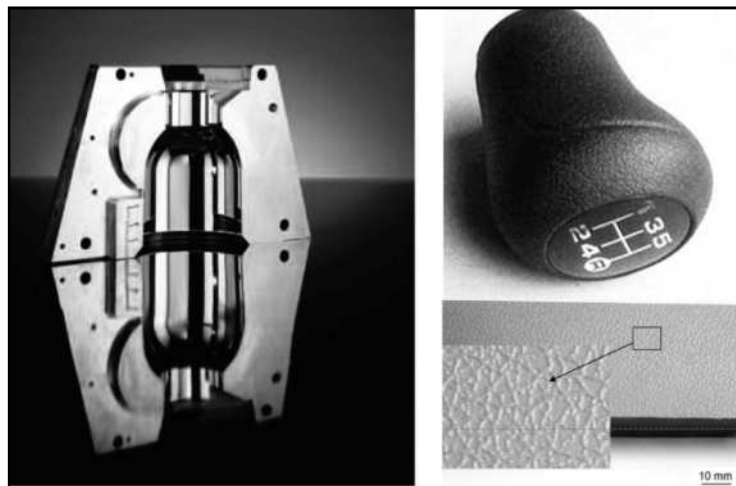
Metrologia pode ser definida como a ciência das medições e contém todos os aspectos teóricos e práticos que garantem a exatidão nos processos produtivos e na transferência de custódia. Por meio dela é possível garantir a qualidade dos produtos e a fidelidade nas transações comerciais, onde sua principal ferramenta é a calibração dos instrumentos de medição associados, sejam estes analógicos ou digitais. Simplificando, a calibração de um instrumento de medição corresponde à determinação das diferenças entre as medições realizadas em um padrão referência e o equipamento a ser calibrado, considerando todo curso de funcionamento. Medir por sua vez é comparar uma grandeza com outra, de mesma natureza, tomada como padrão, portanto, é o conjunto de operações que tem por objetivo determinar o valor de uma grandeza (INMETRO, 2007).

Um padrão metrológico (de qualquer grandeza) reconhecido tem como característica a mais alta qualidade metrológica e cujo valor é aceito sem referência a outro padrão, chamado assim de padrão primário. Um padrão cujo valor é estabelecido pela comparação direta com o padrão primário é chamado padrão secundário, e assim sucessivamente, criando uma cadeia de padrões onde um padrão de maior qualidade metrológica é usado como referência para o de menor qualidade metrológica. Pode-se, por exemplo, a partir de um padrão de trabalho, percorrer toda a cadeia de rastreabilidade desse padrão, chegando ao padrão primário (INMETRO, 2007).

Com a rastreabilidade dos padrões garantida, é possível realizar a calibração dos instrumentos de medição e utilizá-los na inspeção dos produtos produzidos (INMETRO, 2007). Como exemplo de uma aplicação de inspeção pode-se citar o caso dos moldes de injeção de plástico. Para que estes possam ser comercializados, garantindo sua qualidade de fabricação, devem passar por um processo de inspeção, no qual a análise da superfície, rugosidade e suas dimensões são mensuradas. Estas medidas visam evitar problemas ao final do processo de produção.

Os moldes para injeção de plástico normalmente são fabricados com aços especiais (MESQUITA, 2005). Existem moldes lisos e texturizados que necessitam ser analisados, de modo que um risco ou uma falha causada por excesso de polimento, conhecido como “casca de laranja” não sejam confundidos com a textura no momento da análise (BEALL, 2000; SHIMIZU; FUJII, 2003), a Figura 2 mostra o exemplo de um molde para injeção de plástico. Essa análise sobre a qualidade do molde tem como referência a rugosidade da superfície do molde em análise. Além destas características, a indústria do plástico é tão dinâmica, que pode levar a mudanças contínuas nas dimensões de moldes e nos requisitos de produtividade.

Figura 2 – Exemplo de um molde para injeção de plástico.



Fonte: MESQUITA; SCHNEIDER, 2010

O processo de polimento é uma etapa essencial no acabamento do molde. O polimento aplicado sobre a superfície dos moldes é responsável pelo grau de rugosidade superficial do mesmo. A rugosidade é a estrutura de um elemento e está relacionada diretamente a forma de fabricação do produto. A rugosidade representa um grupo de micro-vales e micro-relevos e normalmente estão distribuídos em um

espaço aleatoriamente, como mostrado pela Figura 2. Na Tabela 1 são listados valores de rugosidade superficiais com suas típicas aplicações na indústria (PIRES; TAGIAFERRI; THAUMATURGO, 2002; ASHBY, 1992).

Tabela 1 – Rugosidade superficial típica dos diversos processos de fabricação.

<b>Rugosidade Superficial</b>	<b>Processo de Fabricação</b>	<b>Aplicações Típicas</b>
0,01 $\mu\text{m}$	Micro-polimento	Espelhos
0,1 $\mu\text{m}$	Polimento de alta precisão	Mancais de alta qualidade
0,2 – 0,5 $\mu\text{m}$	Polimento de precisão	Cilindros, pistões e mancais
0,5 – 2,0 $\mu\text{m}$	Usinagem de precisão	Engrenagens e peças de máquina
2,0 – 10,0 $\mu\text{m}$	Usinagem	Mancais pouco carregados e componentes não críticos
3,0 – 50,0 $\mu\text{m}$	Fundição inacabada	Superfícies não solicitadas

Como pode ser observado na Tabela 1 cada tipo de aplicação requer um tipo de rugosidade específica, que em alguns casos pode exigir um valor muito baixo.

O processo de polimento pode apresentar alguns problemas. Dentre os problemas possíveis pode ser citado o excesso polimento, conhecido como “casca de laranja”, este causado pelo tempo excessivo de polimento, causando micro cavidades superficiais. O excesso de polimento na maioria dos casos é gerado por polimento mecanizado.

#### 1.4 JUSTIFICATIVA

Devido ao aumento de fiscalização, enrijecimento da legislação na área metrológica e aumento da necessidade de produtos com maior qualidade dada à concorrência, os processos metrológicos devem auxiliar as empresas a tornarem-se mais produtivas e mais confiáveis na qualidade do produto final entregue. Tais características agregam a necessidade de automação das tarefas quando possível. Entretanto muitos equipamentos metrológicos não possuem interface de comunicação de dados e os que possuem normalmente são muito caros quando comparados aos sem comunicação, o que inviabiliza a compra por muitas indústrias. Por este motivo, os instrumentos mais utilizados são aqueles sem comunicação de dados com o computador, tornando-se necessário o emprego de visão artificial para realização das tarefas que só são possíveis com o uso de visão humana.

Os instrumentos de medição necessitam de calibração para serem comercializados bem como necessitam de calibrações periódicas, sendo este período definido pelas normas de qualidade da empresa que utiliza o instrumento. Estas empresas por sua vez estão sujeitas as normas impostas pelo INMETRO, visando assim assegurar a qualidade de seus produtos. Também se faz necessária a calibração sempre que um instrumento sofrer qualquer tipo de ajuste ou conserto (INMETRO, 2007; VÁSQUEZ-FERNÁNDEZ *et al.*, 2009; SILVA; ALVES, 2004).

Quando nos referimos ao uso de equipamentos para medição na inspeção de produtos fabricados, mais especificamente para inspeção de moldes para injeção de plástico, usualmente realizado com um rugosímetro, praticamente ainda não existem métodos científicos e quantitativos para a análise desta superfície (ZANATTA; GOMES; MESQUITA 2008). Nos dias de hoje esta análise tem como principal referência a experiência de um especialista. Devido tais características, a inspeção dos moldes normalmente consome muito tempo e tem custo elevado, podendo alcançar aproximadamente 1/3 do custo total do molde (BOUJELBENE *et al.*, 2003).

Esses fatores viabilizam a automação de processos metrológicos, visto o grande volume de instrumentos que necessitam calibração no campo industrial, a produtividade dos laboratórios de calibração é precária devido à complexidade dos processos envolvidos nas tarefas.

## 1.5 OBJETIVOS

O objetivo deste trabalho é a automação de processos metrológicos para calibração de instrumentos de medição bem como avaliação e quantificação da qualidade de superfície de moldes para injeção de plástico reduzindo assim a complexidade para as tarefas agregadas a esses processos bem como os custos. Estas técnicas podem ser separadas em dois grupos na linha de pesquisa discutida neste trabalho, sendo a primeira referente à calibração de instrumentos metrológicos e a segunda para inspeção de qualidade para processos metrológicos. Nesta segunda, a inspeção da qualidade de moldes para injeção de plástico é abordada.

## 1.6 MÉTODOS DE TÉCNICAS DE PESQUISA

O método de pesquisa utilizado neste trabalho é o científico com aplicação experimental. Pode ser caracterizado desta forma, pelo fato de que novas metodologias são exploradas em seu todo e discutidas suas eficácias para solução dos problemas propostos.

Quanto a sua classificação pode ser considerada:

- explicativa: os processos que estão sendo estudados serão descritos em sua totalidade, com todas as etapas e processos utilizados na resolução dos problemas;
- aplicada: as metodologias terão aplicações na área metrológica dentro de laboratórios de calibração e na indústria de moldes para injeção de plástico, visando o aumento de produtividade;
- quantitativa: os resultados expressos nas saídas dos sistemas são de forma direta, expressando os erros se existirem de forma a quantificar o problema encontrado.

## 1.7 PUBLICAÇÕES RESULTANTES DA DISSERTAÇÃO

BELAN, P. A.; ARAÚJO, S. A.; LIBRANTZ, A. F. H. Segmentation-free approaches of computer vision for automatic calibration of digital and analog instruments. **MEASUREMENT**, v. 46, n. 01, p. 177-184, 2013.

ARAÚJO, S.A.; ALVES, W. A. L; LIBRANTZ, A. F. H.; Belan, P.A. A comparative study of statistical methods for characterization of materials surfaces by means of texture analysis. **INTERNATIONAL JOURNAL OF COMPUTER APPLICATIONS IN TECHNOLOGY**. (Aceito para publicação 2013)

BELAN, P. A.; ARAÚJO, S. A.; LIBRANTZ, A. F. H. Técnicas de visão computacional aplicadas no processo de calibração de instrumentos de medição com display numérico digital sem interface de comunicação de dados. **EXACTA**, v. 10, n. 1, p. 293-302, 2012.

ARAÚJO, S. A.; BELAN, P. A.; LIBRANTZ, A. F. H. A template matching based algorithm applied to automatic calibration of digital measuring instruments without built-in communication interface. In: **XLIII SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL**, 2011, Ubatuba, São Paulo. p. 1-9

## **1.8 ESTRUTURA DO TRABALHO**

Este trabalho está dividido em seis capítulos, sendo o primeiro, uma introdução sobre os assuntos abordados e a revisão da literatura dos temas abordados; no segundo capítulo, estão descritas as técnicas de visão computacional utilizadas neste trabalho; no terceiro capítulo, são descritos todos os materiais utilizados na pesquisa bem como os métodos abordados; o quarto detalha as metodologias propostas, descrevendo como cada técnica foi desenvolvida e seu funcionamento passo a passo; no quinto capítulo, são apresentados os resultados obtidos pela pesquisa realizada e por fim no sexto capítulo, são apresentadas as conclusões.

## 2 VISÃO COMPUTACIONAL

Visão computacional é considerada como uma subárea de processamento de imagens. Alguns autores definem como uma ciência, a qual faz com que as máquinas possam enxergar tornando possível assim a realização de tarefas como, por exemplo, o reconhecimento de imagens (HUTCHINSON; HAGER; CORKE, 1996). Outros definem como o processo que produz, a partir de imagens do mundo externo, uma descrição que é útil ao usuário e que não é repleta de informações irrelevantes. Deve-se levar em conta a principal característica que um sistema de visão computacional deve possuir, a de ter a possibilidade de reconhecimento de um objeto dentro de uma imagem (GONZALES; WOODS, 2002; LONCARIC, 1998; LONG; ZHANG; FENG, 2005).

Nas últimas décadas pesquisas a respeito de processamento de imagens e visão computacional vêm crescendo muito, em aplicações nas soluções de problemas em diversas áreas, tais quais: como medicina, jogos, inspeção visual, arqueologia, reconhecimento de caracteres (GONZALES; WOODS, 2002; CONCI; AZEVEDO; LETA, 2008; PEDRINI; SCHWARTZ, 2007).

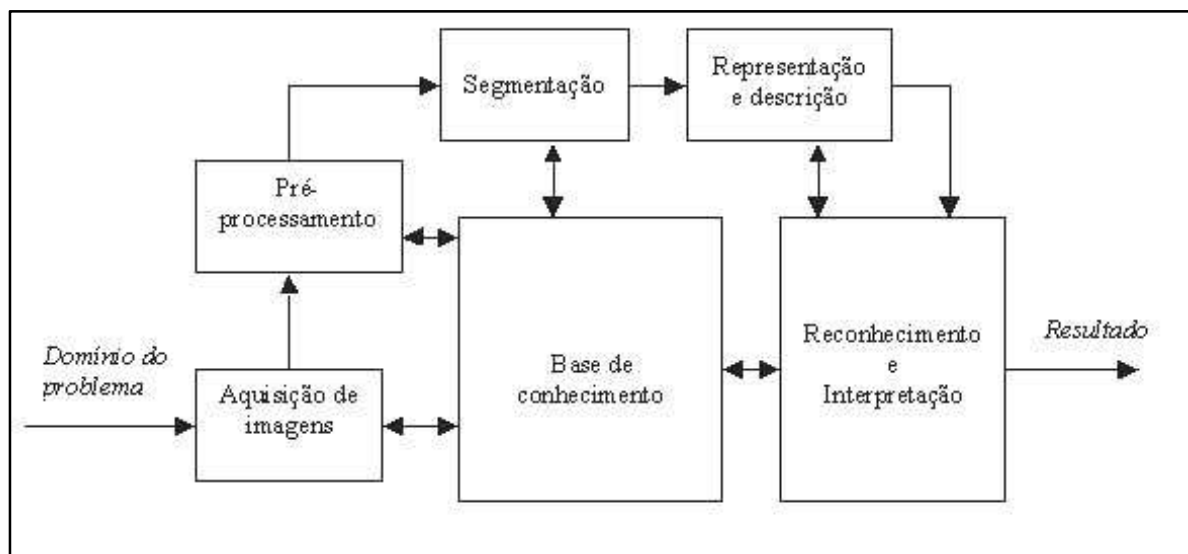
Do ponto de vista matemático uma imagem digital pode ser representada por uma função bidimensional  $f(x, y)$ , com  $x, y \in \mathbb{Z}^2$ , sendo discretizada tanto em coordenadas espaciais (amostragem) quanto na intensidade dos níveis de cinzas (quantização). Uma matriz é capaz de representar matematicamente uma imagem em níveis de cinza ou uma imagem binária, de forma que cada elemento da matriz seja a representação de um pixel. No caso de uma imagem colorida a representação é realizada por  $n$  matrizes, onde  $n$  depende do sistema de cor utilizado, por exemplo, no sistema RGB (*Red, Green and Blue*)  $n=3$  e no CMYK (Cyan, Magenta, Yellow and Black)  $n=4$  (GONZALES; WOODS, 2002; PEDRINI; SCHWARTZ, 2007).

Na representação das imagens em escala de cinza, o valor de  $f$  está compreendido entre 0 a 255, onde 0 é preto e 255 é branco. Para imagens coloridas, cada camada de cor da imagem assume uma função de imagem em níveis de cinza. Desta forma cada camada assumirá o mesmo intervalo de valores de uma imagem em níveis de cinza (GONZALES; WOODS, 2002).

Um sistema de visão computacional deve extrair um conjunto de características da imagem que descreva o objeto a ser localizado, bem como ser um sistema de rápido processamento e que não exija muitos esforços computacionais, viabilizando assim o uso nas indústrias para inspeção visual, visão de robôs, sistemas de vigilância, reconhecimento de caracteres, entre outros (ARAÚJO, 2009).

Basicamente um sistema de visão computacional envolve as etapas de aquisição de imagens, pré-processamento, segmentação, representação e descrição e o reconhecimento (GONZALES; WOODS, 2002; PEDRINI; SCHWARTZ, 2007). A Figura 3 descreve os passos fundamentais deste processo.

Figura 3 – Passos fundamentais em processamento de imagens digitais.



Fonte: GONZALES; WOODS, 2002

Segundo Gonzales (2002), os passos mostrados a Figura 3 podem ser descritos da seguinte forma:

- aquisição de imagens: refere-se à forma em que a imagem é adquirida, seja por meio de uma foto digital, por meio de scanner ou uma câmera trabalhando on-line;
- pré-processamento: tem como objetivo melhorar a qualidade da imagem de forma a aumentar as chances de sucesso na próxima etapa do processamento;
- segmentação: a segmentação é realizada para separar da imagem completa apenas os fragmentos da imagem que são interessantes para a análise, como por exemplo, o reconhecimento de caracteres. Neste caso a

função da segmentação é separar os caracteres para a etapa de reconhecimento;

- representação e descrição: esta etapa determina a forma como os dados adquiridos pelos processos anteriores serão utilizados nas etapas seguintes;
- reconhecimento e interpretação: o reconhecimento é o processo que atribui um rótulo ao objeto, com base em um conjunto de informações previamente fornecidas pelo descritor, e a interpretação é a atribuição de um significado a um conjunto de objetos reconhecidos;
- base de conhecimento: é o conhecimento adquirido e armazenado sobre o problema para consultas futuras.

Dentre a bibliografia consultada, na maioria delas inicialmente é executada binarização nas imagens para depois fazer a extração das características, tornando as imagens mais simples de serem trabalhadas. Desta forma, o maior problema encontrado nessa etapa do processo, é a possibilidade de ocorrer alguma falha na binarização que pode comprometer todo o processo. Dentre as técnicas mais aplicadas em imagens simplificadas aparecem as medidas de similaridades.

A solução ótima para reconhecimento de qualquer objeto em uma imagem é a comparação deste objeto por meio de cálculo de alguma distância em toda a imagem, com todas as escalas, rotações e translações possíveis. Este método também é chamado de força bruta, porém essa é uma técnica de extremo esforço computacional o que inviabiliza seu emprego nos sistemas de visão (TSAI; TSAI, 2000).

A utilização de sistemas de visão computacional na área metrológica deve compartilhar apenas uma característica das descritas acima: ser extremamente eficaz, porém ao contrário da outra característica de custo computacional, deve ser extremamente rápida, com o mínimo esforço e tempo possível, visto que muitas destas aplicações necessitam ser executadas em tempo real.

## 2.1 CASAMENTO DE PADRÕES USANDO CORRELAÇÃO CRUZADA NORMALIZADA

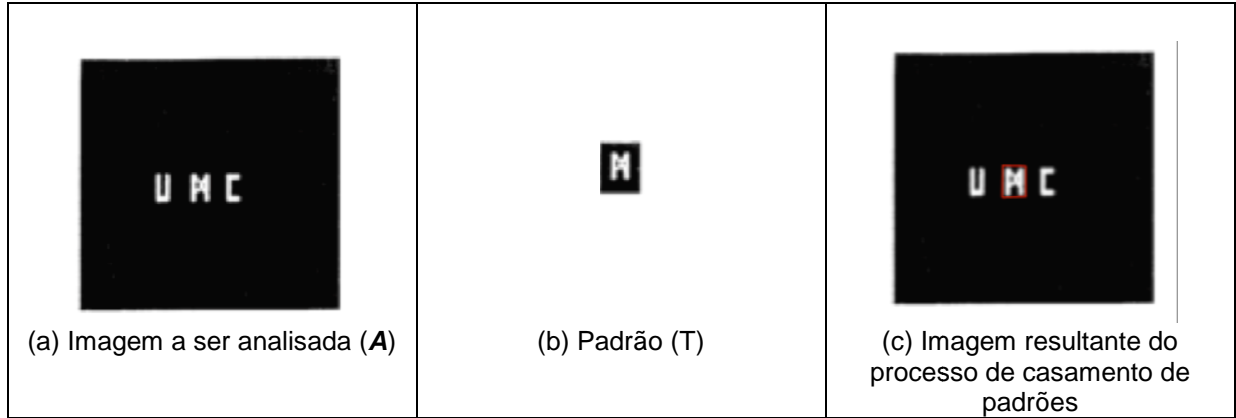
Uma das técnicas mais conhecidas em sistemas de visão computacional é casamento de padrões. São utilizados para encontrar pequenas partes na imagem, estes pedaços são chamados de padrão, em outra imagem que está sendo analisada (KIM; ARAÚJO, 2007; HII *et al.*, 2006).

O casamento de padrões em linhas gerais refere-se à comparação de dois vetores. Existem muitas maneiras de realizar este processo, onde o mais simples deles é a distância euclidiana ou distância mínima. Neste processo são comparados dois vetores de forma a totalizar a menor distância. Caso essa distância seja igual a zero há um casamento perfeito entre os padrões. Este tipo de cálculo de casamento de padrões é útil e rápido quando temos imagens bem comportadas, sem grandes variações de escala e interferências, o que raramente ocorre na prática (GONZALES; WOODS, 2000). Segue uma breve descrição da técnica.

Seja  $A$  uma imagem para ser analisada e  $T$  um padrão. O objetivo do algoritmo de casamento de padrões é encontrar todas as ocorrências de  $T$  em  $A$ , como ilustrado na Figura 4. Casamento de padrões pode ser realizado de duas formas: baseado em característica ou em um padrão. O primeiro método usa as características de  $T$  e  $A$ , como por exemplo, correlação e variância. A segunda forma é chamada de abordagem global e usa todo o *template* para determinar as posições de casamento. Em ambos os casos, alguma medida de similaridade precisa ser empregada. A abordagem baseada em *template* pode ser conduzida utilizando, por exemplo, soma dos erros quadráticos, soma de distância absoluta ou medida de correlação cruzada normalizada (NCC – *normalized cross correlation*) (KIM; ARAÚJO, 2007).

Na maioria dos casos as instâncias de  $T$  em  $A$  podem aparecer rotacionadas, em diferentes escalas, transladadas e com brilhos e contrastes diversos. Por este motivo, as técnicas de casamento de padrões são conhecidas por demandarem um alto custo computacional.

Figura 4 – Resultado de NCC entre duas imagens.



Fonte: GONZALES; WOODS, 2002

A implementação mais simples de casamento de padrões é utilizar a abordagem invariante à rotação e à escala, amplamente conhecida como “força bruta”. Nessas abordagens são executadas uma série de casamento de padrões entre a imagem analisada  $A$  e as instâncias da imagem padrão  $T$ , rotacionado em todos os ângulos possíveis e utilizando para cada rotação todas as possibilidades de escalas possíveis (KIM; ARAÚJO, 2007; ARAÚJO; KIM, 2011).

O algoritmo de força bruta é a solução mais precisa para todos os problemas de reconhecimento de padrões. Entretanto, ele tem um consumo de tempo muito alto, inviabilizando sua implementação em várias aplicações. Todavia, esta técnica pode ser muito eficiente se não existir a necessidade de testar rotação e escalas diferentes.

A técnica de correlação cruzada normalizada (NCC) é amplamente utilizada como medida de similaridade para casamento de padrões e é a forma mais robusta de medida de correlação para determinar a similaridade entre pontos de duas imagens (Hill *et al.*, 2006).

Considerando uma imagem  $A$  de tamanho  $M \times N$  pixels e um padrão  $T$  de tamanho  $I \times J$  pixels, então correlação cruzada entre  $A$  e  $T$ , denotada por  $CC(\mathbf{A}, \mathbf{T})$ , é definida por (Equação 1):

$$CC(\mathbf{A}, \mathbf{T}) = C(m, n) = \sum_{i=-a}^a \sum_{j=-b}^b T(i, j) A(m+i, n+j) \quad (1)$$

onde:  $a=(I-1)/2$ ,  $b=(J-1)/2$ ,  $m=0,..., M-1$  e  $n=0,..., N-1$ ,  $C$  é a matriz resultante, com o mesmo tamanho de  $A$ , na qual os picos indicam as ocorrências de  $T$  em  $A$  (posições de casamento).

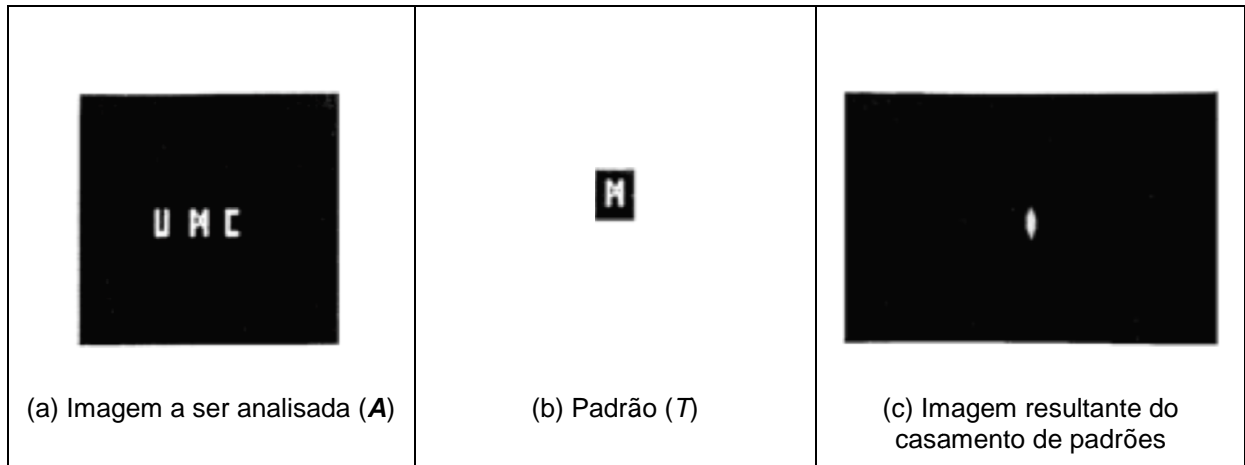
Em várias aplicações o brilho e contraste das imagens podem variar de acordo com as características de iluminação. Nesses casos as imagens podem ser normalizadas primeiramente antes de serem processadas. Desta forma  $NCC(A, T)$  pode ser definida como segue na Equação 2:

$$NCC(A, T) = C(m, n) = \frac{\sum_{i=-a}^a \sum_{j=-b}^b [T(i, j) - \bar{T}(i, j)][A(m+i, n+j) - \bar{A}]}{\sqrt{\sum_{i=-a}^a \sum_{j=-b}^b [T(i, j) - \bar{T}(i, j)]^2 \sum_{i=-a}^a \sum_{j=-b}^b [A(m+i, n+j) - \bar{A}]^2}} \quad (2)$$

O cálculo da  $NCC(A, T)$ , gera uma matriz  $C$  em que os valores variam de -1 a +1. Dependendo da aplicação, pode-se usar tanto o valor absoluto  $|C(m, n)|$  para permitir o casamento em casos negativos de  $T$  em  $A$  ou valor de  $C(m, n)$  com sinal para não permitir o casamento de instâncias negativas de  $T$ .

A Figura 5 mostra o resultado de  $NCC$  entre  $T$  e  $A$ , onde a área destacada na imagem 3c corresponde à ocorrência de  $T$  em  $A$  (GONZALES; WOODS, 2000).

Figura 5 –  $NCC$  entre duas imagens.



Fonte: GONZALES; WOODS, 2002

Na prática, casamento de padrões com  $NCC$  é executado movendo o centro (ou a origem) do padrão  $T$  sobre cada ponto  $(m, n)$  em  $A$  e calculando a soma dos produtos entre os coeficientes em  $A(m, n)$  e  $T(i, j)$  por toda área coberta por  $T$  (GONZALES; WOODS, 2000).

O processo de NCC tem alto custo computacional em relação ao tempo de processamento. No entanto, a implementação desta técnica na maioria das bibliotecas utilizadas em processamento de imagens é otimizada e faz uso de Transformada Rápida de Fourier (FFT), onde os somatórios da correlação são reduzidos para uma matriz dos produtos, acelerando o tempo de processamento (GONZALES; WOODS, 2000). Esse é o caso da biblioteca OpenCV (INTEL, 2012) e ProEikon (KIM, 2012) utilizadas neste trabalho.

## 2.2 ALGORITMO DE DESENHO DE LINHA DE BRESENHAM

O algoritmo de Bresenham é usado para determinar quais pontos devem ser plotados de modo a criar uma linha reta aproximada entre dois pontos. É muito utilizado para desenhar linhas na tela do computador. O principal objetivo deste algoritmo é reduzir o esforço computacional para desenhar uma linha reta, bem como reduzir os erros de arredondamento e de operações de ponto flutuante. Assim, apenas adições e subtrações de inteiros são utilizadas, e estas operações são muito rápidas para a arquitetura de um processador. Foi um dos primeiros algoritmos desenvolvidos no campo da computação gráfica e posteriormente estendido para o desenho de círculos (BRESENHAM, 1965; BRESENHAM, 1977; BRESENHAM, 1982).

## 2.3 LIMIAZIZAÇÃO

Em visão computacional e processamento de imagens, limiarização pode ser definida como o processo de agrupar, utilizando um limiar  $L$ , os pixels que compõem os objetos e os pixels que compõem o fundo da imagem em duas classes distintas. Normalmente este processo consiste na conversão de imagens em escala de cinzas para binárias (Equação 3), onde o valor 1 representa o branco e 0 o preto, podendo ser 1 para representação do fundo e 0 para os objetos ou vice-versa.

$$g(x, y) = \begin{cases} 1, & \text{se } f(x, y) \geq L \\ 0, & \text{se } f(x, y) < L \end{cases} \quad (3)$$

onde  $g$  é a imagem binária de saída e  $f$  é a imagem de entrada em níveis de cinzas.

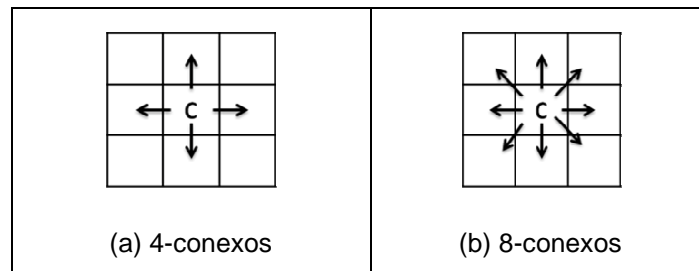
A grande dificuldade em processos de limiarização é estabelecer automaticamente o limiar que classifique os pixels de maneira adequada para uma determinada aplicação (GONZALEZ; WOODS, 2000). Uma técnica amplamente utilizada para limiarização global é o algoritmo de Otsu (OTSU, 1979), o qual permite estabelecer, de forma automática, um limiar  $L$  com base no histograma da imagem.

## 2.4 COMPONENTES CONEXOS

A conectividade entre pixels é um conceito importante usado no estabelecimento de bordas de objetos e componentes de regiões em uma imagem. Para determinar se dois pixels estão conectados, é preciso definir uma noção de vizinhança e se seus níveis de cinza obedecem algum critério de similaridade. Considerando imagens binárias, que contém apenas duas classes (0 e 1), dois pixels vizinhos estão conectados se eles pertencerem a mesma classe, obedecendo assim ao critério de similaridade (GONZALES; WOODS, 2000; PEDRINI; SCHWARTZ, 2007).

Existem duas formas comuns de conectividade para imagens bidimensionais: conectividade-de-4 e conectividade-de-8, conforme os modelos apresentados na Figura 6 (a) e (b). Dois pixels estão conectados de 4 se obedecem ao critério de similaridade e se são vizinhos horizontais ou verticais, e estão conectados de 8, se obedecem ao critério de similaridade e são vizinhos horizontais, verticais ou diagonais. Os vizinhos horizontais de um pixel  $p$  de coordenadas  $(x, y)$  são os pixels de coordenadas  $(x+1, y)$  e  $(x-1, y)$ , os verticais são de coordenadas  $(x, y+1)$  e  $(x, y-1)$  e os diagonais são de coordenadas  $(x+1, y+1)$ ,  $(x+1, y-1)$ ,  $(x-1, y+1)$  e  $(x-1, y-1)$  (GONZALES; WOODS, 2000).

Figura 6 – Exemplo de conectividade e vizinhança de pixels



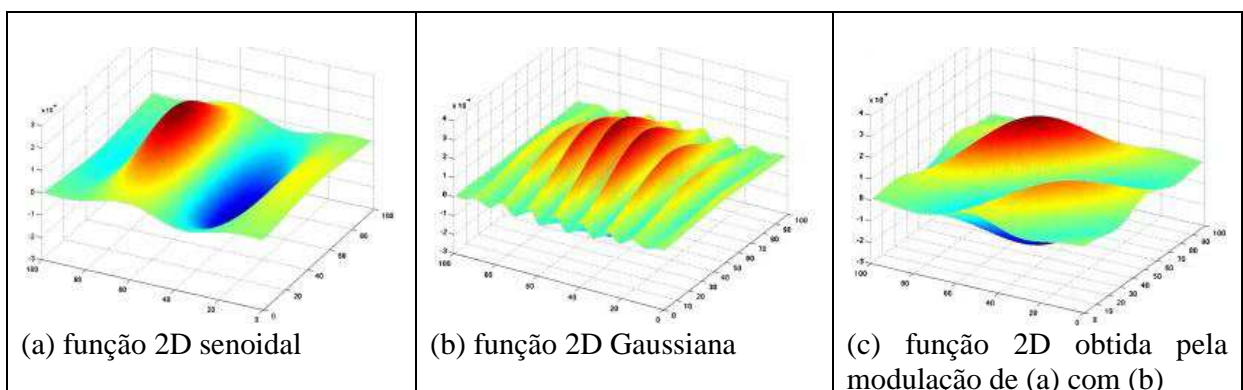
Fonte: BELAN; ARAUJO; LIBRANTZ, 2012

Componente conexo de uma imagem é o conjunto de pixels que estão conectados de 4 ou conectados de 8 entre si (GONZALES; WOODS, 2000).

## 2.5 FILTROS DE GABOR

Os filtros de Gabor pertencem à família dos filtros compostos de Gaussianas e funções harmônicas (seno e cosseno). Este filtro é representado por funções de Gabor senoidal, Gaussianas e modulares entre as duas, conforme mostrado na Figura 7. Este consiste em um filtro linear definido como uma função harmônica multiplicada por uma Gaussiana. A função harmônica pode ser tanto seno como cosseno, ou ainda as duas juntas de forma exponencial complexa. Possui muitas aplicações em processamento de imagens, em especial na classificação de texturas (ZANATA; GOMES; MESQUITA, 2008) e na detecção de estruturas superficiais (PASTERNAK *et al.*, 2009).

Figura 7 – Exemplos de funções de Gabor.

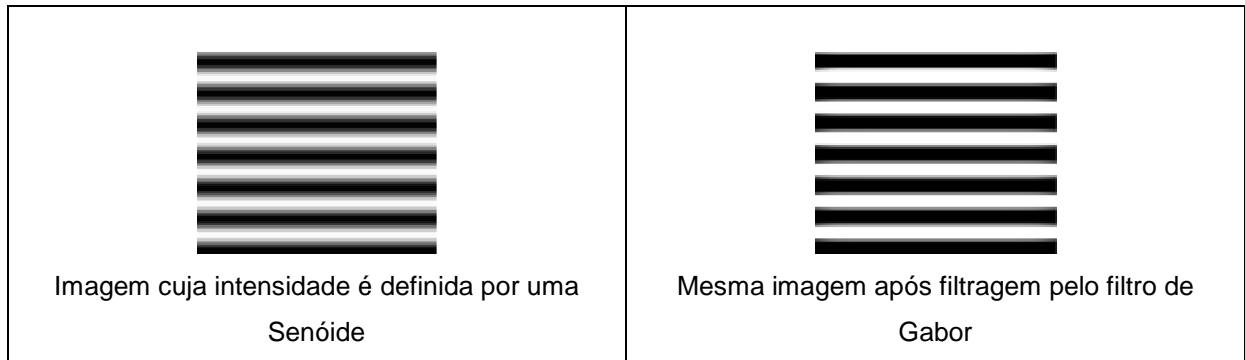


Fonte: ANGELO; HAERTEL, 2001

Devido a suas características, os filtros de Gabor possuem uma grande aderência para imagens com característica senoidal. O filtro de Gabor possui um

conjunto de parâmetros de configuração, entre eles estão frequência e orientação que estão ligadas diretamente ao tamanho do *Kernel* do filtro. Estes parâmetros ajustados de forma correta conseguem uma melhora em imagens como mostrado na Figura 8.

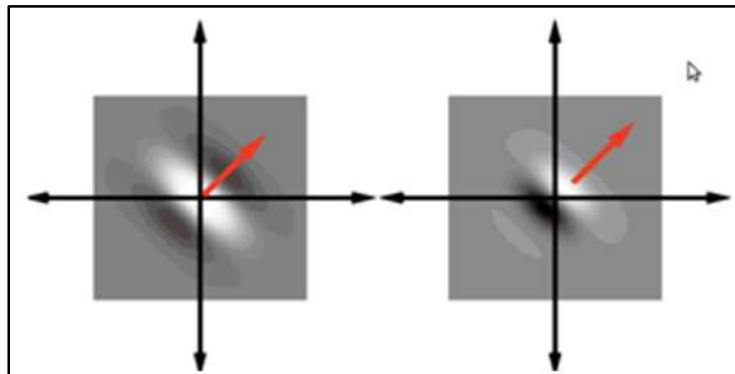
Figura 8 – Imagem sintética com valores de intensidade variando senoidalmente na direção horizontal.



Fonte: LEE; WANG; WU, 2001

A Figura 9 ilustra o papel dos diferentes parâmetros na construção de um Kernel para o filtro de Gabor.

Figura 9 – Imagens de intensidade das partes real e imaginária de um determinado Kernel do filtro de Gabor.

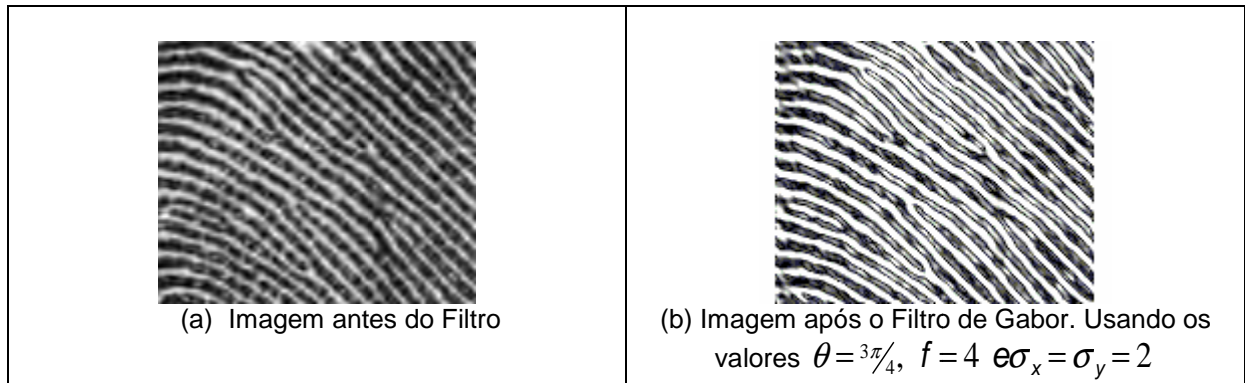


Fonte: FOGEL; SAGI, 1989

O filtro de Gabor bem ajustado é capaz de remover ruídos e preservar as cristas e vales, fornecendo assim as informações contidas nas orientações analisadas. Um filtro muito útil para aplicações de análise de texturas, biomédicas ou naturais, como por exemplo, as biometrias. Vários pesquisadores vêm utilizando extração de atributos de imagens com aplicação deste filtro para classificação de texturas (FOGEL; SAGI, 1989). Como por exemplo, o caso de reconhecimento de impressões digitais, onde neste caso o filtro possui duas tarefas, a remoção de

ruídos da imagem e a reconstrução da mesma, uma vez que estas podem ser vistas com detalhes entre duas cristas paralelas, exemplificado pela Figura 10.

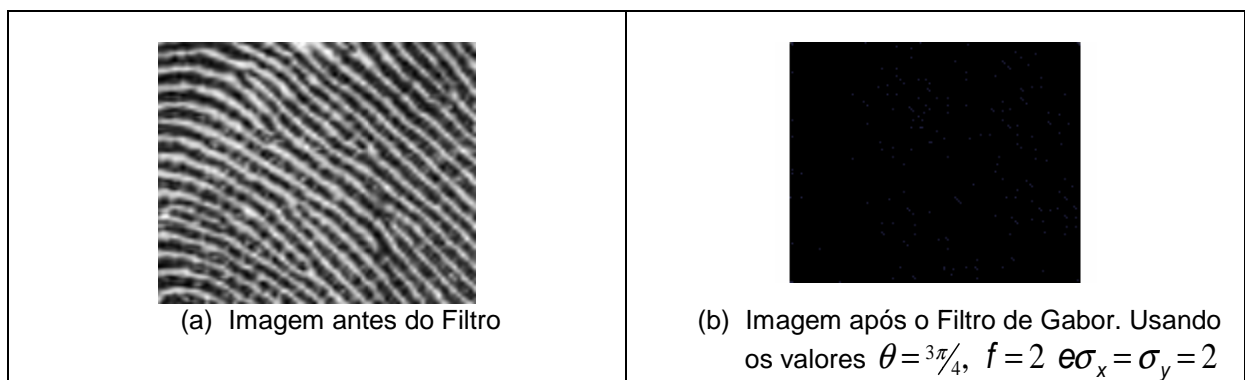
Figura 10 – Parte de uma impressão digital real com orientação aproximada de  $135^\circ$



Fonte: LEE; WANG; WU, 2001

Porém um detalhe que deve ser bem observado é a parametrização deste filtro, pois se este estiver com parâmetros incorretos, ou mal ajustados, a imagem de saída pode ser totalmente o contrário da imagem pretendida, sem informação alguma, conforme mostrado na Figura 11.

Figura 11 – Parte de uma impressão digital real com orientação aproximada de  $135^\circ$ , mas com o valor de frequência errado.



Fonte: LEE; WANG; WU, 2001

### 2.5.1 Descrição do filtro de Gabor

A frequência espacial, em muitos tipos de imagens tem aspectos bem definidos. Por exemplo, em uma imagem de impressões digitais ela é praticamente constante em dada área da imagem. A frequência espacial de impressões digitais é determinada com base na distância média entre as cristas papilares, a qual é praticamente constante em um dactilograma. Cabe ressaltar que esta frequência

varia dependendo do dactilograma e da resolução com a qual o dactilograma foi adquirido.

Sendo  $D_{Ridges}$  a distância entre as cristas (meio das linhas mais escuras) da imagem, a frequência pode ser calculada de acordo com a equação 4:

$$f = \frac{1}{D_{Ridges}} \quad (4)$$

O filtro de Gabor é dado pela equação 5:

$$G(x, y, f, \theta, \sigma) = \exp\left\{\frac{1}{2}\left(\frac{x'^2}{\sigma_x^2} + \frac{y'^2}{\sigma_y^2}\right)\right\} \cdot \exp[2 \cdot \pi \cdot j \cdot f \cdot x] \quad (5)$$

Onde o valor de  $j = \sqrt{-1}$  e os valores de  $x'$  e  $y'$ , necessários para aplicar o filtro de Gabor, são definidos pelas equações 6 e 7 respectivamente:

$$x' = x \cdot \text{sen}(\theta) + y \cdot \cos(\theta) \quad (6)$$

$$y' = x \cdot \cos(\theta) - y \cdot \text{sen}(\theta) \quad (7)$$

O parâmetro de orientação pode ser considerado como um dos principais e é representado pelo  $\theta$ . Esse parâmetro é essencial para o sucesso da resposta do filtro. Todavia não se deve desconsiderar o parâmetro de frequência ( $f$ ), pois se este também estiver mal ajustado as linhas que deveriam ser realçadas não terão a forma esperada, visto o exemplo da impressão digital, se este for um parâmetro muito acima ou abaixo do correto, a imagem não recebe o efeito que deveria ser gerado na resposta.

A equação de Gabor ainda é descrita por duas equações, uma representa a parte real (Equação 8) e a parte imaginária da equação (Equação 9) (FASEL; VARTLETT; MOVELLAN, 2002; JAIN; FARROKHNI, 1991).

$$G_{real}(x, y, f, \theta, \sigma) = \exp\left\{\frac{1}{2}\left(\frac{x'^2}{\sigma_x^2} + \frac{y'^2}{\sigma_y^2}\right)\right\} \cdot \cos(2 \cdot \pi \cdot f \cdot x) , \quad (8)$$

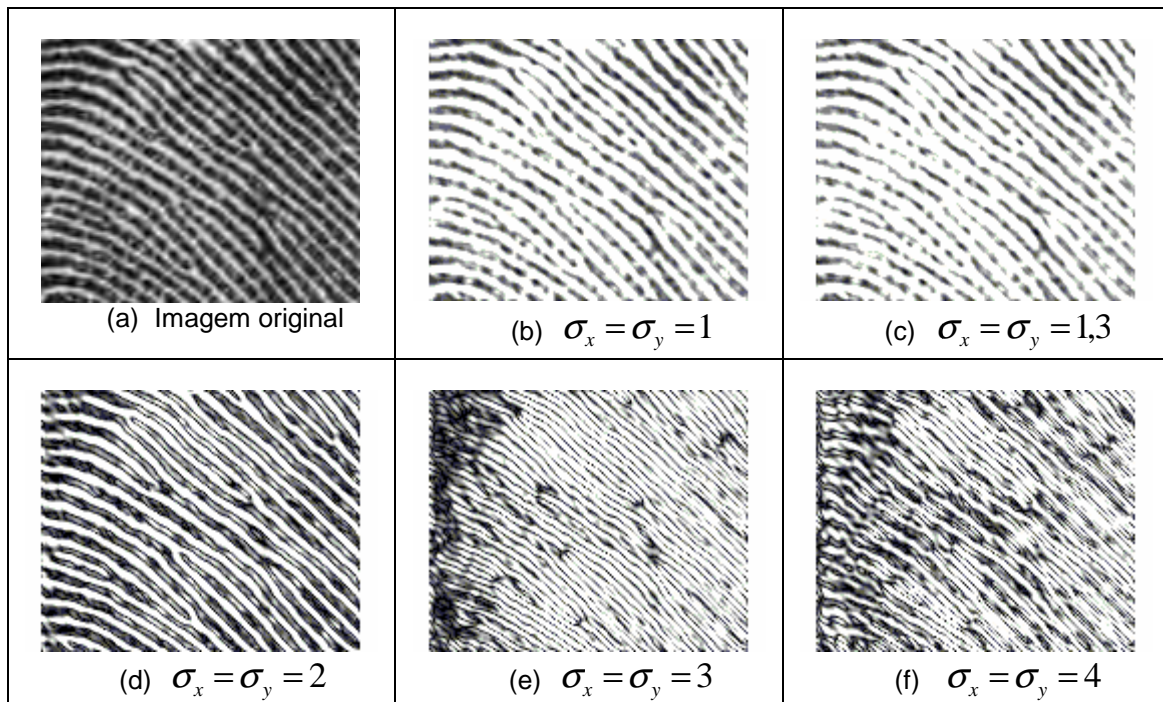
$$G_{\text{imaginario}}(x, y; f, \theta, \sigma = \exp\left\{\frac{1}{2}\left(\frac{x'^2}{\sigma_x^2} + \frac{y'^2}{\sigma_y^2}\right)\right\} \cdot \text{sen}(2\pi \cdot f \cdot x). \quad (9)$$

Onde  $G = G_{\text{real}} + j \cdot G_{\text{imaginario}}$ . Nesse trabalho é utilizado apenas o  $G_{\text{real}}$  da equação. Os componentes  $\sigma_x$  e  $\sigma_y$  da equação de Gabor possuem como origem a equação Gaussiana, e os parâmetros  $\sigma_x$  e  $\sigma_y$  são os desvios padrão que estão relacionados com a largura da equação Gaussiana que modula o filtro de Gabor. Um bom ajuste destes parâmetros resulta em uma boa sensibilidade na remoção de ruídos.

Devido à característica de que as distâncias entre as cristas são praticamente constantes, os componentes  $\sigma_x$  e  $\sigma_y$  também são considerados como constantes na equação.

Vejamos na Figura 12 exemplos de diferentes parametrizações de  $\sigma_x$  e  $\sigma_y$  com diferentes resultados, e como pode ser observado o melhor resultado para esta imagem é  $\sigma_x = \sigma_y = 2$ .

Figura 12 – Imagem após a aplicação do Filtro de Gabor usando os valores  $\theta = \frac{3\pi}{4}$  e  $f = 4$



Fonte: LEE; WANG; WU, 2001

Outro detalhe importante é o fato de que o melhor  $\theta$  para este tipo de imagem está entre  $0^\circ$  e  $180^\circ$ , isso devido às cristas estarem dentro destas inclinações. Desta forma para garantir uma melhor resposta do filtro de Gabor pode-se ainda trabalhar com uma variação de  $\theta$ , aplicados na imagem. A equação (10) a seguir mostra como seria essa tarefa.

$$\theta = i \times \left( \frac{\pi}{k} \right) \text{ onde } 0 \leq i \leq k, \quad (10)$$

onde o valor de  $k$  (*Kernel*) é determinado e corresponde à quantidade de vezes em que o filtro é aplicado sobre a imagem original.

Em classificação de texturas, normalmente um banco de Gabor é montado para extração de características, sendo  $k$  usado para gerar  $k$  imagens complexas de respostas tendo o mesmo tamanho da original (PASTERNAK *et al.*, 2009). Então, cada pixel  $i$  da imagem possui um conjunto de respostas  $k$  com diferentes valores de magnitude para o mesmo pixel  $(x_i, y_i)$  (onde cada valor corresponde à resposta de uma imagem) gerando desta forma um vetor de característica  $v_i$ , usado para alimentar um classificador.

Neste trabalho um banco de 24 *Kernels* foi utilizado na geração das imagens complexas, com as seguintes características: frequências  $f_0 \in \{1/2, 1/3, 1/4, 1/6, 1/8, 1/12\}$ , orientações  $\theta \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$  e  $\sigma_x = \sigma_y = 1/(2f_0)$  para cada Kernel, excluindo os parâmetros  $\phi = 0^\circ$  e  $K=1$  que são constantes. O tamanho de Kernel e seus pesos são definidos por  $2\sigma_x + 1$ .

Como classificador de textura foi utilizada uma rede neural do tipo *Back-Propagation* e devido o tamanho da entrada e quantidade de vezes que a rede deve ser acionada impactar diretamente no tempo de processamento, estes 24 atributos foram reduzidos utilizando Análise de Componentes Principais (ACP). O desenvolvimento de algoritmo de filtro de Gabor foi implementado em C++ e utilizado a biblioteca OpenCV para trabalhar com as imagens e com os vetores complexos, devido a sua fácil implementação dentro desta biblioteca.

## 2.6 ANÁLISE DE COMPONENTES PRINCIPAIS (ACP)

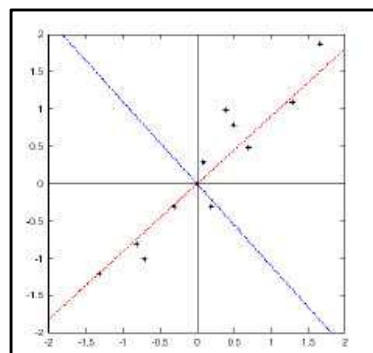
Análise de componentes principais é uma técnica estatística que tem como objetivo transformar um conjunto de dados em outro conjunto de mesma dimensão chamado de componentes principais, estes componentes são a representação linear do conjunto de dados original e são independentes entre si, contendo o máximo de informações em termos de variação do conjunto original. A ideia de funcionamento de ACP está associada à redução de dados, com a menor perda de características possíveis. A operação de ACP redistribui a variação nos eixos originais de forma ortogonal não correlacionado. Esta técnica agrupa os dados de acordo com suas variâncias, ou seja, conforme a população dos dados (JOLIFFE, 2002).

ACP é uma das técnicas mais conhecidas de redução de atributos e um dos métodos estatísticos de múltiplas variáveis mais simples e é a maneira de identificar a relação entre as características extraídas dos dados. Esta técnica é bastante útil quando aplicada a vetores com muitas dimensões, tornando assim uma representação gráfica inviável, mas também pode ser viável a conjunto de dados pequenos.

### 2.6.1 Componentes principais

O componente principal dos dados é o que melhor representa a distribuição conforme mostrado pela linha vermelha, e a componente secundária é perpendicular a principal, representada pela linha azul na Figura 13.

Figura 13 – Exemplo de componente principal. Linha vermelha mostra a distribuição principal dos dados e a linha azul mostra a componente secundária.



Fonte: ANTON; RORRES, 2004

Segundo Anton e Rorres (2004), os passos para realizar o cálculo da componente principal são definidos por:

- obter os dados ou as  $M$  amostras de vetores de dimensão  $n$ ;
- calcular a média ou o vetor médio dos dados;
- subtrair a média de todos os itens do vetor;
- calcular a matriz de covariância usando subtrações. Esta é o resultado da média do produto de cada subtração por ela mesma e terá dimensão  $n \times n$ ;
- calcular os autovalores ( $\lambda$ ) e autovetores da matriz de covariância;
- ajustar a matriz da Transformada de *Hotelling* (cujas linhas são formadas a partir dos autovetores da matriz de covariância arranjados de modo que primeira linha do elemento (0,0) seja o autovetor correspondente ao maior autovalor, e assim sucessivamente até que a ultima linha corresponda ao menor autovalor).

O autovetor com o maior autovalor associado corresponde ao componente principal do conjunto de dados usados. Isso significa que este é o relacionamento mais significativo entre as dimensões dos dados. A Figura 13 ilustra esse ponto no centro da imagem, onde as linhas azul e vermelha se encontram.

### 2.6.2 Matriz de Covariância

As medidas estatísticas possibilitam a análise dos dados das mais diversas formas, como média aritmética, desvio padrão e variância. Essas análises consideram sempre o conjunto de dados classificando e separando por tipo de dados, e a covariância sempre é medida por duas dimensões, sendo calculada a covariância entre a dimensão e seu resultado, que é a variância dos dados. O cálculo de covariância de duas dimensões ( $A, B$ ) é dado pela Equação 11:

$$\text{cov}(A, B) = \frac{\sum_{i=1}^n [(A_i - \bar{A}) \cdot (B_i - \bar{B})]}{n} \quad (11)$$

Na interpretação desta fórmula consideram-se  $\bar{A}$  e  $\bar{B}$  as médias de cada dimensão dos dados e  $A_i$  e  $B_i$  cada elemento de registro dos dados na  $i$ -ésima posição, a variável  $n$  representa o tamanho do conjunto de dados.

Para tratar dados com mais de duas dimensões é necessário calcular a covariância entre cada par de dimensões. No caso de uma matriz de dimensão três ( $A$ ,  $B$  e  $C$ ), a matriz de covariância deve seguir o seguinte formato (Equação 12):

$$matriz\_cov = \begin{pmatrix} cov(A, A) & cov(A, B) & cov(A, C) \\ cov(B, A) & cov(B, B) & cov(B, C) \\ cov(C, A) & cov(C, B) & cov(C, C) \end{pmatrix} \quad (12)$$

A diagonal principal desta matriz contém as variâncias e os demais valores a correlação entre as direções, formando assim uma matriz simétrica com autovetores ortonormais (ANTON; RORRES, 2004).

Para o cálculo do vetor médio dos dados para um conjunto  $M$  de vetores, utilizamos a equação 13 conforme descrita:

$$M_A = \frac{1}{M} \sum_{i=1}^M A_i \quad (13)$$

Tendo-se o valor médio dos dados na matriz  $M_A$  é possível realizar o cálculo da matriz de covariância, subtraindo-se cada  $A_i$  de  $M_A$ . Calculadas todas as subtrações calcula-se o produto de cada subtração por ela mesma, gerando assim a matriz de covariância.

A matriz de covariância para  $M$  amostras de vetores em um conjunto qualquer com vetor médio  $M_A$ , pode ser calculada como segue a Equação 14:

$$Cov_A = \frac{1}{M} \sum_{i=1}^M A_i A_i^T - m_A m_A^T \quad (14)$$

A matriz da covariância é real e simétrica, sendo possível encontrar um conjunto de  $n$  autovalores e consequentemente autovetores ortonormais para cada matriz analisada (ANTON; RORRES, 2004).

### 2.6.3 Autoespaços, autovetores e autovalores

Um vetor  $v$  é considerado um autovetor de uma matriz  $M$  quando  $M$  multiplicado por  $v$  resultar em um múltiplo  $v$ , ou seja, em  $\lambda v$  neste caso,  $\lambda$  é chamado autovalor de  $M$  associado ao autovetor  $v$  (ANTON; RORRES, 2004).

Quando nos referimos aos autovetores, subentende-se “autovetores de comprimento 1”, (não nulos), devido a propriedade desejada ser apenas a direção do vetor. Uma característica importante dos autovetores é que eles são perpendiculares (ortogonais) entre si, o que torna possível expressar os dados em termos dos autovetores, ao invés de termos dos eixos ( $A$ ,  $B$ ,  $C$ ). Para matrizes até  $3 \times 3$ , os autovetores são calculados utilizando a equação (15) de características de  $M$ .

$$\det = (M - \lambda \cdot I) = 0 \quad (15)$$

Onde  $I$  é a matriz identidade,  $M$  a matriz dada e os escalares não nulos,  $\lambda$ , que a solucionam serão os autovalores.

Para matrizes de tamanhos maiores, usualmente calcula-se usando algoritmo numérico iterativo, onde o último passo do algoritmo é ordenar os autovetores de acordo com os autovalores de maior valor, considerados os principais.

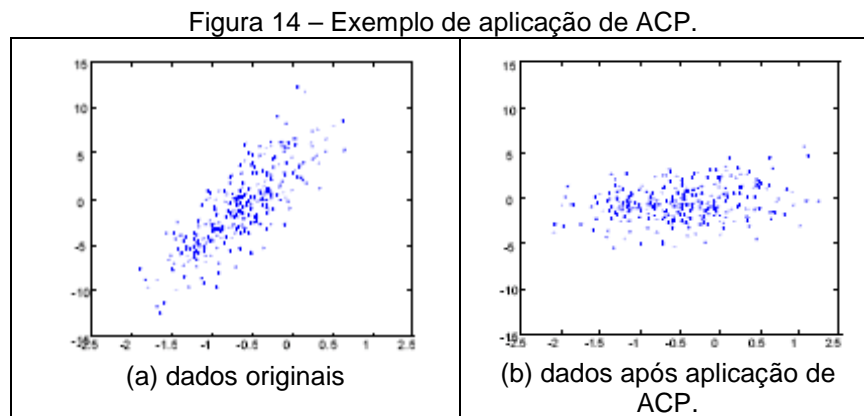
O autoespaço  $M$ , é o espaço de soluções de autovalores não nulos com solução em  $(\lambda \cdot I - M)v = 0$ , e este espaço está associado a  $\lambda$ . As bases para cada autovetores são chamadas de bases de autoespaços.

Dois resultados importantes da álgebra linear podem ser definidos por: se uma matriz  $n \times n$  tem  $n$  autovalores linearmente independentes, então ela é diagonalizável; se uma matriz é diagonalizável então ela tem  $n$  autovalores linearmente independentes que serão os seus elementos da diagonal principal. No processo de diagonalização de uma matriz não existe uma ordem preferencial para os autovetores, pois mesmo que os autovetores estejam em ordens distintas,

teríamos a diagonalização da matriz com outros valores, porém com um resultado válido (ANTON; RORRES, 2004).

#### 2.6.4 Análise dos componentes principais

A análise de componentes principais promove a redução linear dos dados de forma que os resultados desta operação tenham as mesmas características dos dados originais. A Figura 14a mostra o conjunto de dados original em forma gráfica e a Figura 14b mostra o mesmo conjunto de dados após aplicação de ACP.



Fonte: ANTON; RORRES, 2004

A matriz de transformação de dados utilizados pelo processo de ACP é uma matriz cujas linhas e colunas são autovetores da matriz de covariância calculada a partir dos dados originais. A matriz de covariância  $\Sigma$  é uma matriz simétrica positiva, esta pode ser definida pela Equação 16:

$$\Sigma = \frac{1}{N-1} \sum_i (a_i - \mu)^t (a_i - \mu) \quad (16)$$

onde  $N$  é o número de amostras dos dados  $x_i$ , e  $\mu$  é a média do conjunto.

Os autovetores desta matriz são uma nova base de dados que segue a variação dos dados originais. O ACP, portanto consiste na decomposição da base de dados por autovalores de uma matriz.

Vejamos um exemplo de aplicação de ACP em uma imagem colorida utilizando o sistema de cor RGB. Este processo pode ser realizado utilizando três etapas como descrito a seguir:

- primeiro geramos uma matriz de covariância:  $\Sigma = \text{cov}([RGB])$ ;
- calculada a matriz temos uma nova matriz de tamanho 3x3, desta forma pode-se agora calcular os autovalores e autovetores utilizando a fórmula  $[T, aut] = \text{eig}(\Sigma)$ , gerando assim duas novas matrizes ( $T$  e  $aut$ ),  $\text{eig}$  representa a operação de obtenção dos autovalores e autovetores da matriz  $\Sigma$ .

- desta forma aplicando a equação 
$$\begin{bmatrix} P_1(i) \\ P_2(i) \\ P_3(i) \end{bmatrix} = \begin{bmatrix} t_{1,1} & t_{1,2} & t_{1,3} \\ t_{2,1} & t_{2,2} & t_{2,3} \\ t_{3,1} & t_{3,2} & t_{3,3} \end{bmatrix} \cdot \begin{bmatrix} R(i) \\ G(i) \\ B(i) \end{bmatrix}$$
 temos um novo espaço de dados ( $P_1$ ,  $P_2$  e  $P_3$ ) que corresponde à nova matriz de dados com aplicação de ACP.

## 2.7 REDES NEURAIS ARTIFICIAIS

Segundo Gonzales & Woods (2002) e Pedrini & Schwartz (2007), redes neurais artificiais (RNA) podem ser consideradas como técnica de visão computacional devido sua vasta aplicabilidade em reconhecimento de padrões.

As RNAs foram desenvolvidas baseadas no funcionamento do cérebro humano. Tem-se como grande objetivo, a modelagem do mesmo para que as máquinas fiquem dotadas do poder de raciocinar, através de conhecimentos adquiridos do ambiente externo (aprendizado) (ROCHA, 2003).

Dentre as inúmeras definições de Redes Neurais Artificiais, Haykin (2002) descreve-as como um sistema onde existem inúmeros processadores distribuídos paralelamente gerando uma única unidade de processamento com habilidade de armazenar conhecimentos de forma natural, tornando-os disponíveis para uso futuro, assemelhando assim ao funcionamento do cérebro humano. Ainda cita a

capacidade de aprendizado por sistema de aprendizagem e utiliza a conexão entre os neurônios para armazenar este conhecimento adquirido.

O processo utilizado para o aprendizado de uma RNA é chamado de algoritmo de aprendizagem, que tem como principal função o ajuste dos pesos sinápticos da rede (RUSSEL, 2004).

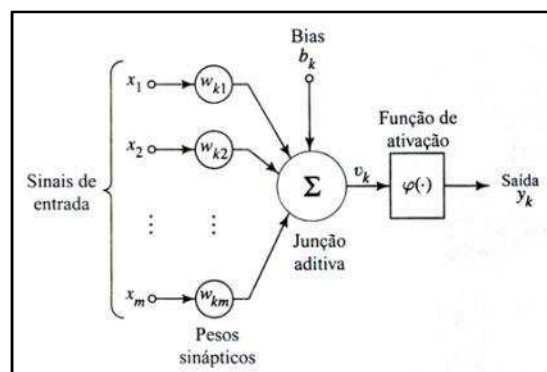
### 2.7.1 Modelo lógico de um neurônio

O neurônio é à base do processamento de uma rede neural. Nele temos três elementos básicos como mostrado:

1. **conjunto de sinapses ou elos de conexão:** neste conjunto são guardados e calculados os pesos sinápticos da rede. A notação deste é dada por  $w_{ij}$ ;
2. **somador:** responsável pela soma dos pesos multiplicados pelas entradas da rede;
3. **função de ativação:** usada para limitar a amplitude do sinal de saída de um neurônio.

Na Figura 15 é ilustrado o modelo de um neurônio artificial com todas as suas características intrínsecas. A seguir, descrevemos os componentes do neurônio.

Figura 15 – Modelo gráfico de um Neurônio Artificial



Fonte: HAYKIN, 2001

Como mostrado na Figura 15, o bias (sinal de disparo da RNA), serve para aumentar os graus de liberdade, permitindo uma melhor adaptação ao conhecimento fornecido à rede, é definido pelo  $b_k$ ,  $x_1$ ,  $x_2$ , ...,  $x_n$  são os sinais de entrada da rede,

$w_{k1}, w_{k2}, \dots, w_{kn}$  são os pesos sinápticos do neurônio  $k$ ,  $u_k$  representa a saída linear do combinador de acordo com os sinais de entrada ou estado de ativação da rede e  $\varphi(.)$  é a função de ativação da rede. Por fim o  $y_k$  é o sinal de saída do neurônio.

### 2.7.2 Modelo matemático de um neurônio

O neurônio pode ser representado matematicamente por meio das equações 17 e 18:

$$u_k = \sum_{j=1}^m w_{kj} x_j \quad (17)$$

$$y_k = \varphi(u_k + b_k), \quad (18)$$

onde  $x$  são os sinais de entrada,  $w$  são os pesos sinápticos de  $k$ ,  $u$  são as saídas,  $b$  é o bias e  $\varphi$  é a função de ativação.

Existe outro modelo matemático muito parecido com o anteriormente citado, só que neste novo modelo é adicionado um sinal de entrada fixo em +1 e um novo peso sináptico igual ao bias (HAYKIN, 2001).

Funções de ativação:

1. Função linear: neste tipo de ativação, a rede é acionada pelo valor 0 ou 1. Esta função também é conhecida como função de *Heaviside*. Nesse modelo a saída de um neurônio assume 1 se o local de indução for positiva e 0 para campos com valores negativos, como apresentado pela Equação 19 (HAYKIN, 2001).

$$\phi(v) = \begin{cases} 1, & v \geq 0 \\ 0, & v < 0 \end{cases} \quad (19)$$

2. Função limiar por parte: esta função de ativação pode ser classificada como um aproximador para um amplificador não linear de sinal dentro de uma região linear. Sua ativação possuiu três condições conforme mostrado na Equação 20 (HAYKIN, 2001).

$$\phi(v) = \begin{cases} 1, & v \geq +\frac{1}{2} \\ v, & +\frac{1}{2} > v > -\frac{1}{2} \\ 0, & v \leq -\frac{1}{2} \end{cases} \quad (20)$$

3. Função sigmóide: esta função é a forma tradicional de ativação das redes neurais. É definida por uma função crescente e gera um resultado balanceado para funções lineares e não-lineares. Um detalhe importante é que as Equações citadas anteriormente (19 e 20), ficam limitadas a uma ativação por 0 ou 1, enquanto que a função sigmóide possibilita uma ativação entre o intervalo -1 e 1, ou seja, no caso de uma ativação negativa, assume que é uma função ímpar do campo local induzido (HAYKIN, 2001).

$$\phi(v) = \begin{cases} 1, & v > 0 \\ 0, & v = 0 \\ -1, & v < 0 \end{cases} \quad (21)$$

### 2.7.3 Forma de propagação do sinal em RNA

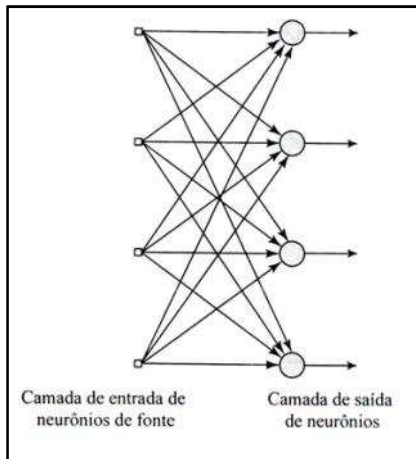
#### 2.7.3.1 Redes alimentadas adiante

As redes alimentadas adiante têm sua estrutura dividida em duas ou mais camadas. Existe uma camada de saída e uma camada de entrada, podendo existir uma ou mais camadas ocultas, que tem como função amplificar o poder de processamento da rede, capacitando-a para trabalhar com problemas mais complexos. Esta forma de rede é conectada diretamente entre si (TAFNER, 1995).

O fluxo de informações nesse modelo é em sentido único. Assim o sinal se propaga dos nós de entrada em direção aos nós de saída, passando por todas as camadas da rede até chegar à camada de saída. Estas ligações podem ser de dois tipos basicamente: totalmente conectada ou parcialmente conectada. Isto se difere pelo fato de todos os nós estarem conectados diretamente com todos os nós das camadas posteriores e anteriores ou não (HAYKIN, 2001).

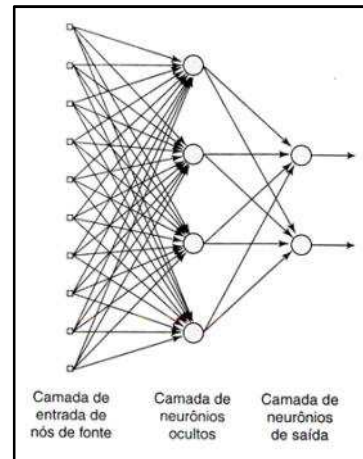
As Figuras 16 e 17 mostram estruturas possíveis para redes alimentadas adiante. Observamos no primeiro exemplo que não existem camadas ocultas na rede, e no segundo já há uma camada oculta.

Figura 16 – RNA alimentada adiante sem camada oculta.



Fonte: HAYKIN, 2001

Figura 17 – RNA alimentada adiante com uma camada oculta.



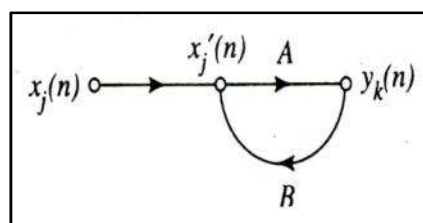
Fonte: HAYKIN, 2001

### 2.7.3.2 Realimentação

A realimentação se dá quando é necessário que a rede execute mais de um ciclo de ajustes de pesos, a qual ocorre da seguinte forma: quando uma saída é calculada, ela é realimentada na camada de entrada da rede, assim modificando a entrada de cada neurônio. Esta tarefa é repetida até que uma regra de parada seja alcançada (HAYKIN, 2001).

A representação gráfica da realimentação é mostrada na Figura 18.

Figura 18 – Representação de uma realimentação.



Fonte: HAYKIN, 2001

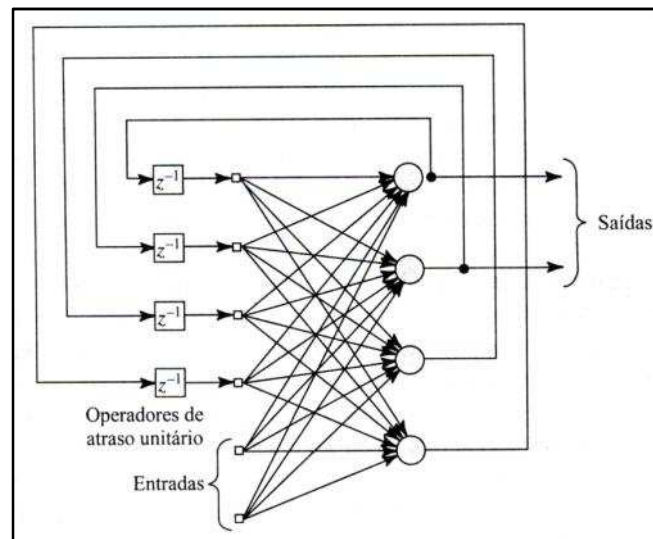
Esta estrutura é utilizada em redes recorrentes.

### 2.7.3.3 Redes recorrentes

As redes recorrentes podem possuir os dois tipos de estrutura, com camada única ou de múltiplas camadas. O que as diferenciam das estruturas anteriormente definidas é a capacidade da realimentação, ou seja, os sinais de saída são retornados para os nós de entrada e a rede processa quantas vezes forem necessárias para que se obtenha uma saída desejada (HAYKIN, 2001).

Essa rede possui uma capacidade de aprendizagem e desempenho muito grande, e muito superior à alimentada adiante pelo fato de existir a realimentação (RUSSEL, 2004), conforme esquema mostrado na Figura 19.

Figura 19 – Representação de uma rede recorrente.



Fonte: HAYKIN, 2001

### 2.7.4 Treinamento de Redes Neurais Artificiais

Conhecimento para RNAs refere-se à informação armazenada ou a modelos utilizados por uma pessoa ou máquina para interpretar, prever e responder ao mundo exterior (HAYKIN, 2001 APUD FISHER & FERCHEIN, 1987).

A representação do conhecimento tem duas características bem definidas:

1. que informação é realmente tornada explícita;
2. como a informação é codificada fisicamente para o uso subsequente.

Uma das mais importantes fases da implementação de uma RNA é o treinamento, pois nesta fase a rede adquire o conhecimento. Este treinamento pode ser feito de duas formas: supervisionado ou não supervisionado. Na primeira forma os dados de entrada são passados para rede juntamente com o resultado desejado, enquanto que na segunda forma os dados são passados de forma que a rede deve decidir sozinha qual deverá ser sua saída.

O conhecimento dentro da rede é representado pelos pesos sinápticos. Assim, aprendizado de uma RNA se dá, basicamente, pelo ajuste dos pesos sinápticos.

Existem dois modos de analisar uma entrada de uma rede neural: a incorporação prévia que consiste em restringir a arquitetura da rede e da escolha dos pesos sinápticos e compartilhando os mesmos, nesta forma as entradas não podem sofrer muitas variações em seu formato. O outro modo é a incorporação de invariâncias na rede que é a capacidade da rede reconhecer o objetivo mesmo não estando em sua forma real, como no caso de uma imagem rotacionada (RUSSEL, 2004).

#### 2.7.5 Aprendizagem

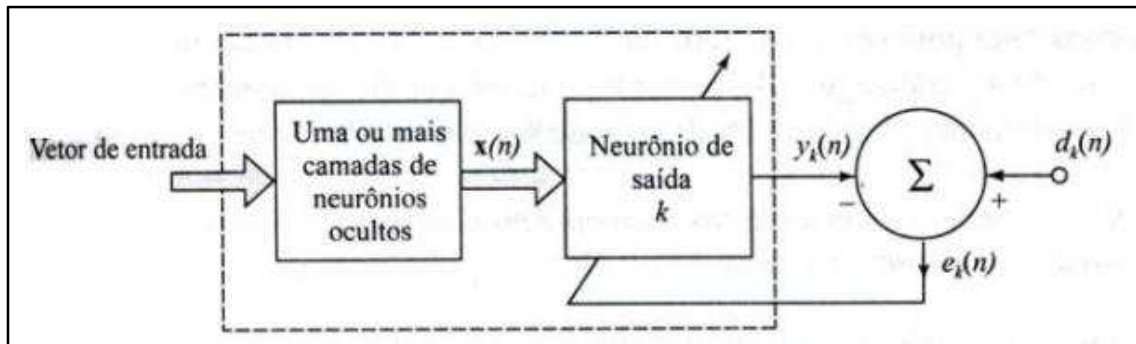
A etapa de aprendizagem é aquela em que a rede realmente aprende, onde os seus pesos são ajustados de acordo com o algoritmo. Aprendizagem em RNAs pode ser definida como um processo pelo qual os parâmetros livres de uma rede neural são adaptados através de um processo de estimulação pelo ambiente no qual a rede está inserida. O tipo de aprendizagem é determinado pela maneira pela qual a modificação dos parâmetros ocorre (RUSSEL, 2004 APUD MENDEL & MCCLARENT, 1970).

##### 2.7.5.1 Tipos de aprendizagem:

a) aprendizagem por correção de erro: consiste na rede receber um sinal de entrada e transmitir um sinal de saída, que é comparado com a saída desejada e

assim, o seu peso sináptico é ajustado proporcionalmente ao erro encontrado. Neste modo, aproximando ao máximo seu peso do objetivo. A Figura 20 representa um sistema de aprendizagem por correção de erro (HAYKIN, 2001).

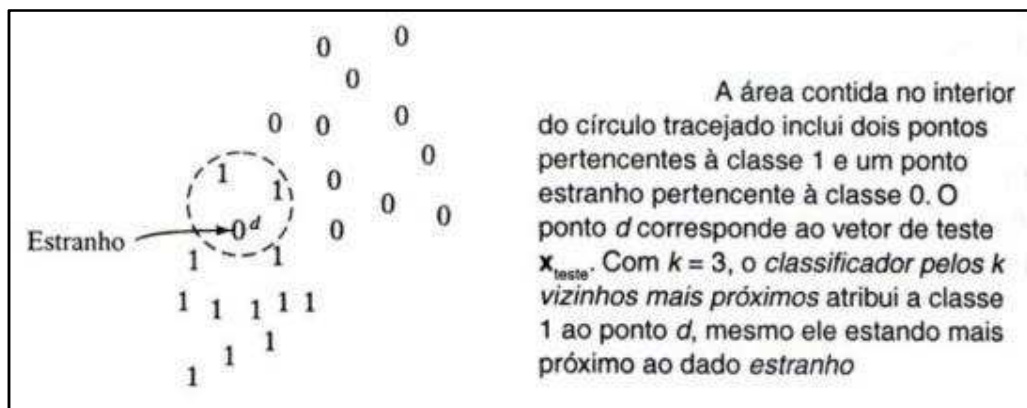
Figura 20 – Representação de um modelo baseado na correção de erro.



Fonte: HAYKIN, 2001

b) aprendizagem baseada em memória: baseia-se em experiências passadas. Necessita de uma grande quantidade de exemplos (entradas e saídas). O algoritmo para este modelo restringe-se a dois pontos, o critério para definir a vizinhança local e após isso, o treinamento da mesma, esta vizinhança é definida pela distância euclidiana, ou distância mínima. A Figura 21 representa esta estrutura (HAYKIN, 2001).

Figura 21 – Representação de um modelo baseado em Memória.



Fonte: HAYKIN, 2001

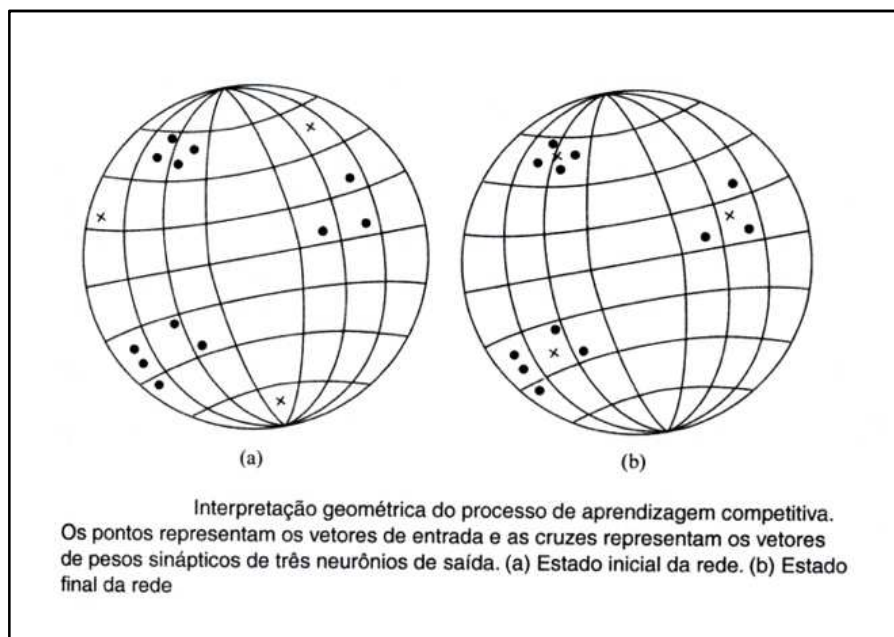
c) aprendizagem Hebbiana: este é o mais antigo e conhecido dos modelos de aprendizagem, e foi desenvolvido pelo neuropsicólogo Donald Hebb, em 1949.

Uma aprendizagem Hebbiana é uma sinapse que usa um mecanismo dependente do tempo, altamente local e fortemente interativo para aumentar a

eficiência sináptica como uma função de correlação entre as atividades pré-sinápticas e pós-sinápticas (HAYKIN, 2001 APUD BROWN, 1990).

d) aprendizagem competitiva: na aprendizagem competitiva, os neurônios de saída quando acionados competem entre si para ver qual será o vencedor, ou seja, o neurônio vencedor é o responsável por representar um padrão de entrada. A Figura 22 mostra um exemplo dessa estrutura (HAYKIN, 2001).

Figura 22 – Representação de um modelo baseado em competição.



Fonte: HAYKIN, 2001

#### 2.7.5.2 Aprendizagem Supervisionada

Nesse tipo de aprendizagem, o treinamento deve ser supervisionado por um agente externo (o professor) para que esse possa corrigir os erros. Nesta forma de aprendizagem, após a rede produzir sua resposta, ela é avaliada e, caso não esteja correta acontece a realimentação, para que a rede possa ajustar seus pesos sinápticos. Esta operação é repetida quantas vezes forem necessárias até que se atinja algum critério de parada estipulado (RUSSEL, 2004).

### 2.7.5.3 Aprendizagem Não Supervisionada

Na aprendizagem não-supervisionada a rede não tem uma supervisão de um agente externo em sua fase de treinamento, mas sim regras iniciais a serem seguidas e depois a possibilidade de ajustar-se automaticamente. Nesse tipo de aprendizagem as saídas desejadas não são necessárias, bastam apenas os vetores de entrada. Redes que usam esse tipo de aprendizagem são utilizadas como classificadores de padrões (RUSSEL, 2004).

### 2.7.6 Modelos de redes neurais

Será apresentada a seguir, uma breve descrição de alguns modelos de redes neurais.

#### 2.7.6.1 Percéptrons de camada única

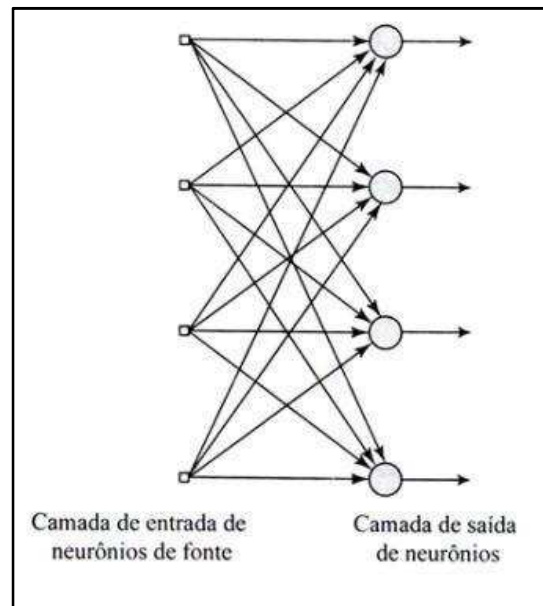
Percéptron é um neurônio não-linear e seu objetivo é a classificação de padrões, dentre duas classes pré-determinadas. Criado por Rosenblatt em 1958, o percéptron de camada única foi o primeiro modelo de aprendizagem supervisionada.

Dada sua simplicidade, o percéptron é um modelo de rede neural capaz apenas de classificar padrões linearmente separáveis (HAYKIN, 2001).

O neurônio neste modelo funciona por base de filtros e este filtro é aplicado pelo algoritmo do mínimo quadrado médio (LMS) que foi desenvolvido por Widrow e Hoff em 1960. Os pesos sinápticos do percéptron são ajustados a cada iteração do algoritmo que irá processar até que se atinja uma regra para parada, relacionada ao número de iterações ou à obtenção da saída desejada pelo LMS (RUSSEL, 2004).

Este modelo é formado por uma camada de entrada e uma camada de saída de neurônios. A Figura 23 representa esse modelo.

Figura 23 – Representação de um modelo Percéptron de Camada Única.



Fonte: HAYKIN, 2001

#### 2.7.6.2 Percéptron de Múltiplas camadas.

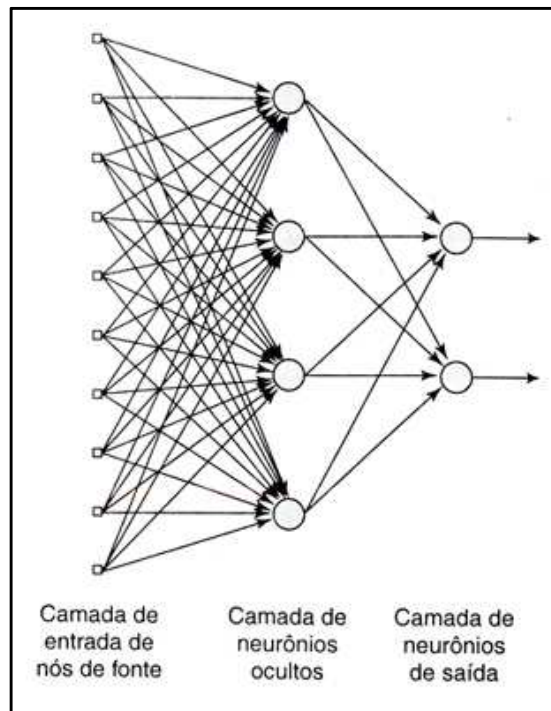
Este modelo de RNA possui camada de entrada, uma ou mais camadas ocultas e uma camada de saída. Os sinais de entrada se propagam para frente, passando por todas as camadas ocultas até chegar ao seu destino que é a camada de saída (TAFNER, 1995).

Para este modelo normalmente é utilizado o algoritmo de retropropagação de erro, conhecido como *Back-Propagation*. Este algoritmo é baseado em aprendizagem de correção de erro e é dividido da seguinte forma:

- propagação: nessa fase os pesos sinápticos da rede são fixos e a rede flui para frente até a camada de saída, conhecida como sinal funcional;
- retropropagação: nessa fase os pesos sinápticos da rede são ajustados de acordo com a saída desejada e a sua saída real da rede, que resulta num erro. Assim a rede flui em sentido contrário. Esse processo é conhecido como retropropagação de erro e foi criado por Rumelhart e McClelland em 1985.

A Figura 24 representa os percéptron de múltiplas camadas (MLP). Essa estrutura possui as camadas ocultas, que é a principal característica desse modelo.

Figura 24 – Representação de um modelo Percéptron de Múltiplas Camadas.



Fonte: HAYKIN, 2001

Esta característica de possuir múltiplas camadas, juntamente com a capacidade de treinamento gera o poder de processamento computacional conhecido nos dias de hoje, e também é essa junção que recebe mais atenção nos estudos também, pois o seu comportamento ainda não é sempre estável (ROCHA, 2003).

O algoritmo de retropropagação fornece uma aproximação dos pesos sinápticos da rede. Quanto menor for este valor maior é a taxa de aprendizagem. Para conseguir um alto grau de aprendizado na rede, deve existir um bom treinamento, com muitas entradas diferentes. Quanto maior for o número de exemplos apresentados à rede, melhor será o seu desempenho. O mau treinamento, ou seja, com poucos exemplos, pode deixar o treinamento mais rápido, mas pode causar instabilidade no seu funcionamento (RUSSEL, 2004).

#### 2.7.7 Algoritmo Back-Propagation

O algoritmo *back-propagation* é um modelo que derivou de um modelo antigo o percéptron/ adaline (HAYKIN, 2001). Seu funcionamento tinha como

principal característica o mínimo de três camadas: uma camada de entrada; uma ou mais camadas ocultas e uma de saída. Utiliza também a função do tipo *sigmóide* como função de limiar (ou ativação).

O *back-propagation* é um dos algoritmos mais utilizados para o treinamento de RNA's supervisionadas do tipo MLP. Ele tem como principal característica o ajuste dos pesos sinápticos da rede. Estes ajustes podem utilizar diversas funções, uma delas é a *sigmóide*. Os pesos são ajustados da seguinte maneira: após a entrada ser apresentada na rede, e calcular a resposta da rede, comparam-se a resposta da rede com a resposta desejada. Se a resposta da rede estiver errada, é calculado o erro e inicia-se o processo de ajustes dos pesos da rede fluindo em sentido contrário, do fim da rede (camada de saída), até o início da rede (camada de entrada), este processo é a retropropagação (HAYKIN, 2001).

A fase de treinamento de uma RNA utilizando *back-propagation* pode requerer muitas iterações computacionais e normalmente leva um longo tempo para que o aprendizado atinja o nível desejado. Este treinamento pode ser acelerado, mas isso pode levar a uma instabilidade no funcionamento da rede (RUSSEL, 2004 APUD RUMELHART, 1986).

A aplicação do algoritmo depende muito da preparação da entrada dos dados para a rede, e dependendo de como estiver esta entrada é necessário um processo de pré-processamento grande, que consistirá em transformar uma entrada qualquer em uma entrada disponível para a rede neural.

A estrutura do algoritmo pode conter uma ou mais camadas ocultas como já vimos anteriormente, mas em grande parte das aplicações não há necessidade de utilizar-se mais de uma camada oculta.

O treinamento de uma rede é considerado concluído quando sua taxa de erro for muito próxima de zero, ou ainda, se a rede atingir a estabilidade desejada.

#### 2.7.7.1 Lógica matemática para implementação do algoritmo

Vejamos a seguir um breve resumo de como é o funcionamento do algoritmo em relação a sua implementação matemática para o cálculo do erro no seu processo

de retropropagação, que é iniciado na sua última camada, caminhando para a camada inicial.

Iniciam-se os pesos e coeficientes de limiar com valores pequenos e randômicos. Apresenta-se o vetor de entrada (padrão) e a saída desejada. Calcula-se a saída, por meio da Equação 22:

$$v_j(n) = \sum_{i=0} w_{ji} \cdot y_i^{l-1}(n) \quad (22)$$

onde  $j$  é o neurônio,  $l$  a camada que se encontra,  $n$  a quantidade de entradas da rede,  $w$  peso,  $i$  neurônio da camada anterior,  $y$  é o sinal de saída do neurônio  $i$  na camada  $l-1$ .

Ao sinal de saída, aplica-se a função de transferência  $\phi$ . Para todas as camadas exceto a primeira camada oculta:  $y_j^{(L)} = \phi_j(v_j(n))$ . Caso esteja na primeira camada oculta aplica-se  $y_j^{(0)}(n) = x_j(n)$ . Caso o  $j$  esteja na última camada aplica-se  $y_j^{(l)} = o_j(n)$ .

1. Calcula-se o sinal de erro:

$$e_j(n) = d_j(n) - o_j(n)$$

onde  $d$  é a saída desejada.

Sempre os cálculos dos pesos da rede são baseados nos erros da camada de saída, que será a referência para todas as camadas da rede.

2. Calculam-se os gradientes locais para camada de saída:

$$\delta_j^{(l)}(n) = e_j^{(L)}(n) \cdot (\phi_j'(v_j^{(L)}(n)))$$

e calcula-se os gradientes locais para as camadas ocultas da rede:

$$\delta_j^{(l)}(n) = \phi_j'(v_j^{(L)}(n)) \cdot \sum_k \delta_k^{(l+1)}(n) \cdot w_{kj}^{(l+1)}(n)$$

o  $\phi_j'(\cdot)$  representa que a regra delta generalizada foi utilizada para o cálculo.

3. Volta-se no ponto três e repete-se o processo até que seja satisfeito o critério de parada, que normalmente é medido através do erro médio quadrático.

#### 2.7.7.2 Lógica Computacional para implementação do Algoritmo *Back-propagation*

A lógica computacional para implementação do algoritmo *back-propagation* é mostrada no Quadro 1.

Quadro 1 – Lógica computacional do algoritmo back-propagation.

**“função** *back-propagation* (exemplos, rede) **retorna** uma rede neural

**entradas:**      **exemplos:** um conjunto de exemplos, cada um com vetor de entrada  $x$  e um vetor de saída  $y$   
                      **rede:** uma rede de varias camadas com  $L$  camadas, pesos  $W_{j,i}$ , função de ativação  $g$

**repita**

**para cada**  $e$  **em** exemplos **faça**

**para cada** nó  $j$  na camada de entrada **faça**  $a_j \leftarrow x_j[e]$

**para**  $l = 2$  **até**  $M$  **faça**

$in_l \leftarrow \sum_j W_{l,j} a_j$

$a_l \leftarrow g(in_l)$

**para cada** nó  $i$  na camada de saída **faça**

$\Delta_i \leftarrow g'(in_i) \times (y_l[e] - a_i)$

**para**  $l = M-1$  **até**  $1$  **faça**

**para cada** nó  $j$  na camada  $l$  **faça**

$\Delta_j \leftarrow g'(in_j) \sum_i W_{l,j} \Delta_i$

**para cada** nó  $i$  na camada  $l+1$  **faça**

$W_{j,i} \leftarrow W_{j,i} + \alpha a_i^x \Delta_j$

**até** algum critério de parada ser satisfeito

**retornar** HIPÓTESE-DA-REDE-NEURAL (rede)” (RUSSEL, 2004).

O algoritmo *back-propagation* proposto neste trabalho foi implementado em linguagem C++. Assim desde o processo de aquisição da imagem até a saída final de todo o processo proposto, está em um único programa, dando continuidade nas rotinas de circularidade, conforme descrito a seguir.

## 2.8 CIRCULARIDADE

A circularidade tem como propósito medir a similaridade entre uma estrutura qualquer e um círculo. A circularidade em sua forma original é definida pela Equação 23:

$$CI = \frac{P^2}{A}, \quad (23)$$

onde  $CI$  é o resultado da circularidade,  $P$  é o perímetro (somatório dos pixels que compõem o contorno da imagem) da área em análise e  $A$  é área total da imagem em análise. Vale ainda ressaltar que a circularidade é perfeita quando seu valor é  $4\pi$ .

No capítulo 3 a seguir são apresentados os materiais utilizados no trabalho e os equipamentos desenvolvidos.

### 3 MATERIAIS E MÉTODOS

#### 3.1 EQUIPAMENTOS

Para a condução dos experimentos alguns equipamentos foram utilizados, sendo eles:

- um manômetro digital da marca Zurich modelo Z-10 (Figura 25). Este equipamento é utilizado para verificar a pressão em diversas áreas de aplicação, desde a indústria até a medicina, com seu uso nos aferidores de pressão;

Figura 25 – Manômetro digital marca Zurich.



Fonte: Autor

- três relógios comparadores analógicos (Figura 26), sendo estes um Mitutoyo modelo 1044F, o segundo Starrett modelo 3025-481 e o terceiro da marca Pantec modelo 12307NJ-10. Os relógios comparadores são medidores de deslocamento. Suas aplicações são focadas na indústria no processo de inspeção de peças produzidas, normalmente utilizados em bancadas de medição, pois devido sua precisão este equipamento tem que trabalhar fixo. Tem como base de funcionamento o deslocamento de um fuso que move o ponteiro de indicação;

Figura 26 – Relógios comparadores.



Fonte: BELAN; ARAUJO; LIBRANTZ, 2013

- uma peneira grano-volumétrica da marca Bertel modelo Inox 304 – 38 $\mu$ m, mostrada na Figura 27. Peneiras grano-volumétricas tem como objetivo a seleção de produtos, como por exemplo a seleção de grãos onde possam conter sujeira no meio, neste caso pode ser utilizado duas peneiras, uma com o tamanho onde o grão passe, ficando assim elementos maiores, e uma segunda menor que o grão, tirando assim as impurezas menores que os grãos, dentre outras muitas aplicações possíveis;

Figura 27 – Peneira grano-volumétrica.



Fonte: Autor

- um laser de Hélio-Neônio (He-Ne) operando em modo contínuo (cw) e sintonizado em 808nm que é usado para geração do *speckle* das superfícies dos moldes analisados;
- cinco moldes para injeção de plástico fabricados em aço descritos na Tabela 2;
- 1 câmera USB CMOS de 30 FPS (FPS – *frames per second*) para aquisição das imagens para calibração de manômetro digital e análise de superfície de moldes;

- 1 câmera USB CMOS de 60 FPS para o projeto de calibração do relógio comparador analógico, que possui uma taxa de FPS superior as demais devido à necessidade de alta velocidade de aquisição e processamento das imagens para a implementação do projeto, como será descrito na seção 3.3.2.;
- 1 microscópio digital USB 5~500x para aquisição das imagens para calibração da peneira grano-volumétrica;
- 3 motores de passo da marca AKYAMA modelo AK39H-12-1.8, utilizados para automação dos dispositivos, em todos os casos os motores utilizados possuem a força de 1,1 Kgf/cm, e um motor de passo de 5 fases da marca Autonics para um controle mais preciso utilizado no dispositivo para calibração de relógio comparador analógico;
- um *encoder* da marca Autonics para realizar a leitura do ângulo da rotação do motor, desta forma sendo possível a realização da medida do deslocamento do fuso utilizado para movimentação do relógio comparador, assim tendo uma medida precisa do deslocamento, e sendo assim o padrão metrológico do dispositivo;
- 3 *drivers* de controle do motor de passo sendo dois deles para controle de motores unipolares com controle de dois motores simultâneos, 1 *driver* da marca Autonics para um motor de 5 fases, utilizado no projeto de calibração de relógio comparador analógico;
- um computador com processador Intel Core2Duo 2,8 GHz, 1 Gb RAM, software Dev-C++ (BLOODSHED, 2012) para desenvolvimento dos aplicativos bem como as bibliotecas de processamento de imagem OPENCV (INTEL, 2012) e ProEikon (HAE, 2012).

Tabela 2 – Características dos moldes para injeção de plástico.

Exemplo	Fabricante	Modelo	Material	Dimensão (mm)	Polimento	Rugosidade Ra <sup>1</sup>	Defeito
1	Mitutoyo	Bloco Padrão Série 516	Aço	08 x 15	Mecânico	0,020 µm	Não
2	Usinado		Aço P20	13 x 20	Manual	0,028 µm	Sim
3	Usinado		Aço P20	13 x 20	Manual	0,028 µm	Sim
4	Usinado		Aço P20	10 x 10	Manual	0,034 µm	Não
5	Usinado		Aço P20	10 x 10	Manual	0,045µm	Não

<sup>1</sup>Ra - Média aritmética de rugosidade

Dentre estes cinco moldes quatro deles foram fabricados para a realização dos testes; nos moldes de número 2 e 3 foram incluídos propositalmente danos de excesso de polimento e riscos.

### 3.2 MÉTODOS

Para a automatização da calibração dos instrumentos de medição (manômetro digital, relógio comparador analógico e peneira grão-volumétrica) e inspeção dos moldes de injeção, foram utilizadas diversas técnicas de visão computacional, sendo a única característica em comum o uso de uma câmera USB como entrada de informação para os métodos propostos. Em todos os casos a câmera é ligada ao PC para aquisição e processamento das imagens em tempo real. No caso do manômetro é necessária a intervenção do técnico de calibração apenas para indicar o momento que deve ser realizado para a leitura do instrumento, devido ao controle de pressão no equipamento ser manual. Já nos outros três casos os processos foram todos automatizados por meio de protótipos conforme descrito no item 3.4 deste capítulo.

Todas as automações dos dispositivos foram implementadas em linguagem C++, bem como os algoritmos Bresenham, Otsu, componentes conexos, Filtros de Gabor e circularidade. Para o algoritmo de NCC foram utilizadas rotinas prontas da Biblioteca ProeiKon e toda a programação foi realizada utilizando o Software Dev-C++ (BLOODSHED, 2012). Apenas a tarefa de treinamento da ACP não foi implementada, para isso, foi utilizado o software WEKA (WEKA, 2012) para gerar a matriz de autovetores. Cada item deste trabalho (calibração manômetro, calibração relógio, calibração peneira e análise da superfície do molde) está em um programa distinto, gerando assim quatro códigos-fonte de programas.

Para avaliação dos tempos gastos nos processos de calibração de forma automática foram adquiridos os tempos de calibração para cada processo tratado neste trabalho junto a um laboratório de metrologia acreditado pelo INMETRO.

### 3.3 BANCOS DE IMAGENS

#### 3.3.1 Calibração de manômetro digital

Para a condução dos experimentos foram adquiridas 50 imagens com valores distintos no display do manômetro digital. As 50 imagens em questão geram então 250 dígitos que possuem a distribuição contida na Tabela 3. Como pode ser visto, todos os 10 dígitos aparecem em pelo menos 7,6% das imagens.

Tabela 3 – Distribuição dos dígitos no conjunto de imagens.

Dígito	0	1	2	3	4	5	6	7	8	9
%	14,4	13,2	10	10	7,6	7,6	9,6	9,2	9,6	8,8

Estes displays possuem em seu mostrador 5 dígitos e um ponto, o qual é descartado, visto que esta informação é determinada pela precisão do instrumento em análise. Uma amostra deste display é mostrada na Figura 28.

Figura 28 – Exemplo do display utilizado para reconhecimento dos dígitos.



Fonte: Autor

#### 3.3.2 Calibração de relógio comparador analógico

Para este experimento foram adquiridos seis vídeos de relógios comparadores em movimento, sendo dois vídeos de cada instrumento descrito anteriormente com o curso completo do instrumento. Cada vídeo tem duração média

de 3 minutos. Esse tempo está relacionado à velocidade de deslocamento do fuso, visto que esta velocidade está associada com a velocidade da câmera utilizada para captura (60 fps (frames por segundo)). O tamanho dos vídeos está relacionado à quantidade de passos necessários para o motor rotacionar o fuso de forma a percorrer a capacidade completa do relógio, visto que cada passo dado é adquirido um frame do vídeo, totalizando desta forma em média de 10.800 frames para cada vídeo. Um destes frames é mostrado na Figura 29. Ainda para este experimento foi desenvolvido um protótipo para facilitar estas aquisições, conforme item 3.5 desta seção.

Figura 29 – Exemplo de um quadro dos vídeos capturados para os experimentos.

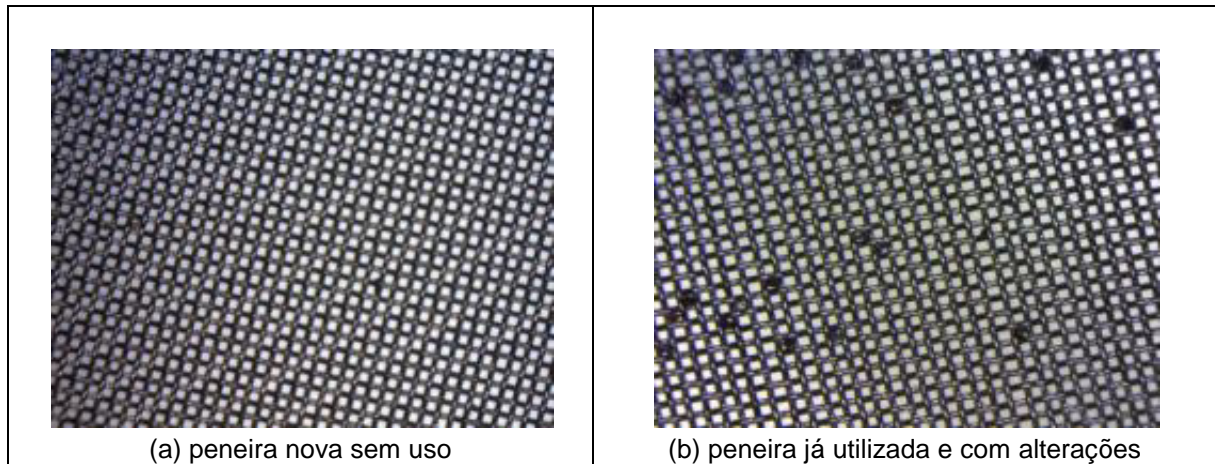


Fonte: BELAN; ARAÚJO, LIBRANTZ, 2013

### 3.3.3 Calibração de peneira grano-volumétrica

Para a condução desses experimentos foram adquiridas 20 imagens das tramas de duas peneiras grano-volumétricas, uma nova, sem uso, o que garante a qualidade e a conservação da medida de sua trama. Como mostrado na Figura 30a possui uma trama limpa, sem sujeiras e totalmente uniforme. Na segunda peneira utilizada, a trama já contém alterações consideráveis em seus tamanhos e também problemas de sujeiras, normalmente encontrados em equipamento de uso contínuo, conforme mostrado na Figura 30b.

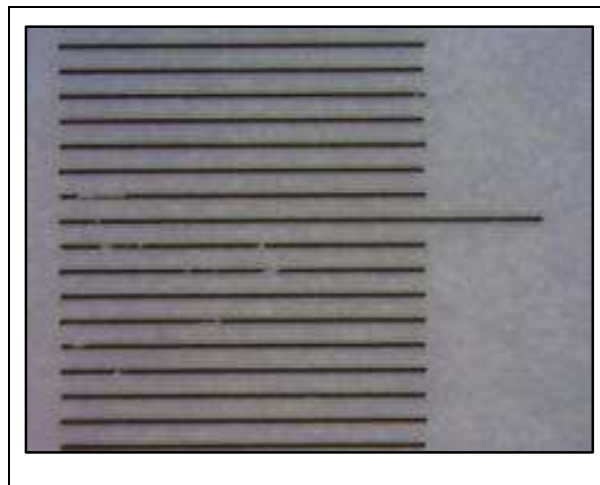
Figura 30 – Exemplos de imagens adquiridas de peneiras grano-volumétrica.



Fonte: Autor

Como referência para realiza-se as medidas da trama, foi adquirida uma imagem de uma régua graduada de cristal padrão. Esta imagem foi utilizada para calibrar o sistema, medindo a quantidade de pixels existentes entre os traços, conforme mostrado na Figura 31. Cada divisão desta escala equivale a 0,01mm.

Figura 31 – Régua graduada padrão.



Fonte: Autor

Para este experimento foi desenvolvido um protótipo conforme descrito no item 3.4.2 desta seção.

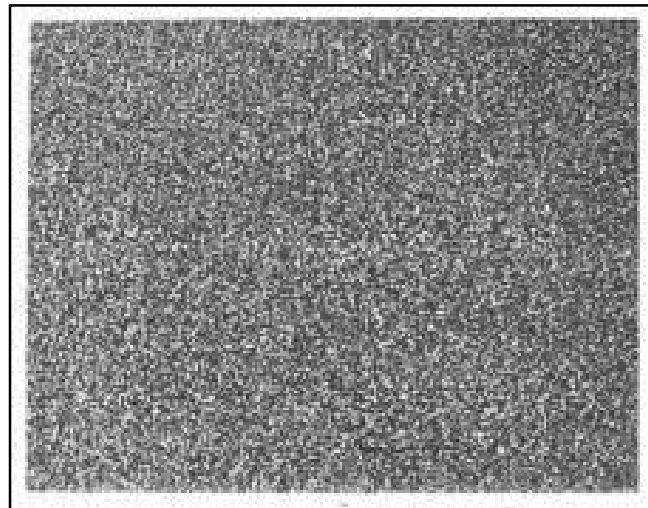
#### 3.3.4 Análise de superfície de moldes para injeção de plástico

O *speckle* é o um dos principais efeitos resultantes da interação de uma iluminação coerente (feixe laser) com uma superfície que possa ter reflexão. Efeito *speckle* refere-se ao fenômeno de interferências de ondas mutuamente coerentes

com uma variação ao acaso de fase, e o resultado desta interferência é a distribuição de máximos e mínimos na intensidade da luz. Os picos individuais de intensidade representando pequenas manchas claras ou escuras são chamados de *speckle*. O agrupamento de *speckle* na realidade são confundidos e chamados desta forma, enquanto que na realidade estes picos de luzes estão presentes em grupos gerando uma imagem como mostrado na Figura 32 (PAIVA; MURAMATSU, 2004).

Quando uma luz coerente incide sobre uma superfície que possui micro-relevo superficial, esta luz é espalhada e transmitida, chamadas de campo do *speckle*, gerando assim uma imagem (PAIVA; MURAMATSU, 2004). Na Figura 32 é apresentado o *speckle* de uma superfície metálica.

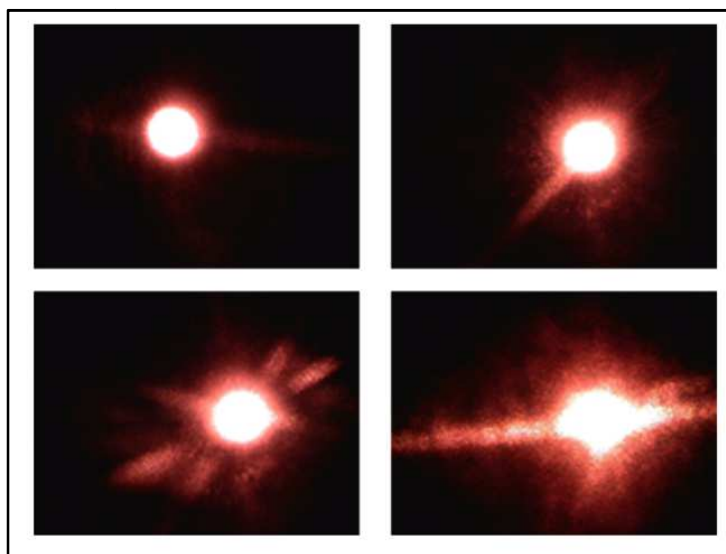
Figura 32 – Exemplo de geração de *speckle*.



Fonte: PIRES; TAGIAFERRI; THAUMATURGO, 2002

Para a condução desses experimentos um banco de imagens foi gerado com base nos cinco moldes descritos anteriormente. Essas imagens foram adquiridas a partir do *speckle* da superfície do molde projetada sobre um aparato acrílico. Utilizando a técnica de rasterização foi possível garantir a aquisição de imagens sobre toda a superfície do molde. Para o primeiro molde, 420 imagens foram adquiridas, enquanto que para os outros quatro foram adquiridas 3000 imagens (750 para cada um). A quantidade de imagens adquiridas para cada molde está relacionada ao tamanho do mesmo. A Figura 33 mostra uma sub-amostragem deste banco de imagens capturadas para análise futura.

Figura 33 – Sub-amostragem das imagens adquiridas dos moldes.



Fonte: Autor

Devido a necessidade de um deslocamento suave e preciso para a aquisição das imagens foi desenvolvido um protótipo conforme descrito na seção 3.4.3 deste capítulo.

### 3.4 PROTÓTIPOS DESENVOLVIDOS

Para condução e avaliação de resultados das metodologias propostas neste trabalho, foram desenvolvidos três protótipos, sendo um deles para calibração de relógio comparador, o segundo para calibração de peneira grão-volumétrica e o terceiro para condução dos experimentos de análise de moldes para injeção de plástico.

#### 3.4.1 Calibração de relógio comparador

Para realização dos experimentos envolvendo relógio comparador, foi desenvolvido um protótipo para que fosse possível fazer um deslocamento de forma suave e precisa no fuso de movimentação do relógio, visto que estes equipamentos trabalham com uma precisão alta e são extremamente sensíveis. Desta forma foram evitados pulos no ponteiro do relógio bem como a câmera ficou alinhada ao

equipamento, mantendo sempre o centro do relógio no mesmo ponto, como já apresentado na Figura 29.

Este protótipo foi produzido em aço, assim é possível garantir a rigidez do dispositivo bem como a precisão das medidas realizadas. Para seu desenvolvimento alguns equipamentos foram utilizados, dentre eles estão uma base usinada com os suportes para o relógio comparador, a câmera, o fuso, o *encoder* e motor de passo. A câmera USB escolhida possui uma taxa de 60 FPS, superior às câmeras de uso normal para vídeo conferência (normalmente 30 fps). Isto faz-se necessário devido a necessidade de agilidade no processo de calibração, visto que a velocidade do passo do motor está atrelada diretamente a velocidade da câmera, pois para cada passo que o motor um frame da câmera deve se aquisitado. O fuso utilizado no projeto também possui uma característica que garante precisão na medida, isto pelo fato de ser um fuso não rotativo. O motor de passo é da marca Autonics e o grau do passo é  $0,9^\circ$ . Esse motor tem um torque de 1,2kgf/cm e para seu controle foi utilizado um driver da marca Autonics. O encoder também é da marca Autonics e trabalha com a resolução de 1024 pulsos por volta. A Figura 34 mostra o protótipo de automação do relógio comparador.

Figura 34 – Protótipo calibrador de relógio comparador.



Fonte: Autor

### 3.4.2 Calibração de peneira grano-volumétrica

O dispositivo para calibração da peneira grano-volumétrico provém rotação para a peneira, de forma que este deslocamento dê uma volta completa na peneira durante o processo de calibração, fazendo cinco paradas no seu percurso para aquisição das imagens de sua trama, o que garante a amostragem requerida pelas normas de qualidade utilizadas pelo INMETRO.

Este dispositivo (Figura 35) necessita de uma iluminação especial. Para isso, utiliza-se uma luz por baixo da peneira e uma segunda luz vindo por cima da peneira (a própria iluminação do microscópio USB). Os equipamentos utilizados no desenvolvimento deste protótipo foram: uma base usinada com os suportes para câmera, mesa de iluminação inferior e motor de passo, um microscópio USB com ampliação de 500x com taxa de 30 fps com iluminação de LED, um eixo rotativo, uma luz polarizada de 10 watts, um motor de passo marca Akiyama modelo AK39H-12-1,8° e um driver para o motor (Jumpear).

Figura 35 – Protótipo calibrador de peneira grano-volumétrica.



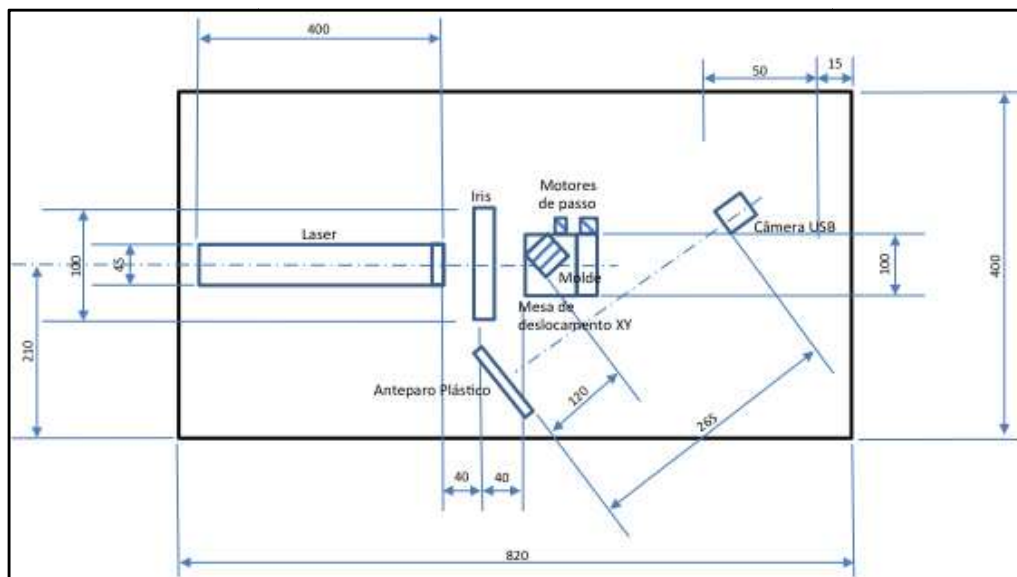
Fonte: Autor

### 3.4.3 Aparato Laser

Para condução dos experimentos da superfície da qualidade dos moldes foi construída uma caixa acrílica que acondicionou todos os dispositivos. Essa caixa tem como principal característica a cor do seu material (Azul Royal) que é capaz de bloquear o comprimento de onda do laser (808nm), tornando seguros os experimentos, visto que este comprimento de onda pode ser danoso a vista humana. Segue abaixo uma breve descrição do funcionamento do aparato.

Um feixe de luz laser incide sobre o molde, gerando uma reflexão de sua superfície, em um anteparo branco, que pode ser capturada pela câmera USB. A Figura 36 mostra a disposição de todos os itens do dispositivo.

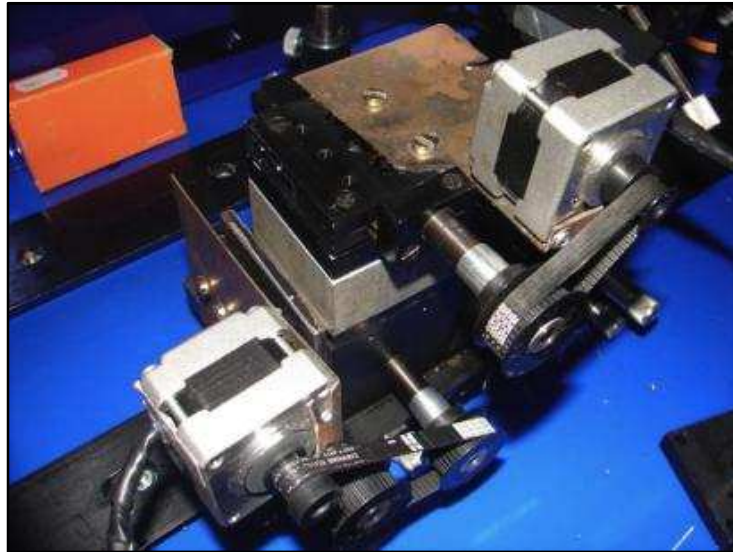
Figura 36 – Projeto da caixa de proteção com dispositivos.



Fonte: Autor

Para ter-se controle sobre a posição do molde analisado, a automatização da mesa de deslocamento XY mostrou-se necessária. Esta automação foi realizada com uso de dois motores de passo e um driver duplo interfaceado à porta paralela para controle dos motores. Assim a mesa tem controle automático de movimentação vertical e horizontal. O molde é deslocado automaticamente na forma *raster-way*, e devido a precisão no controle dos motores de passo, é possível fazer pequenas paradas (33 milésimos de segundos) para a aquisição da imagem e o registro do ponto capturado. Esta automação da mesa é mostrada na Figura 37.

Figura 37 – Mesa de deslocamento XY.

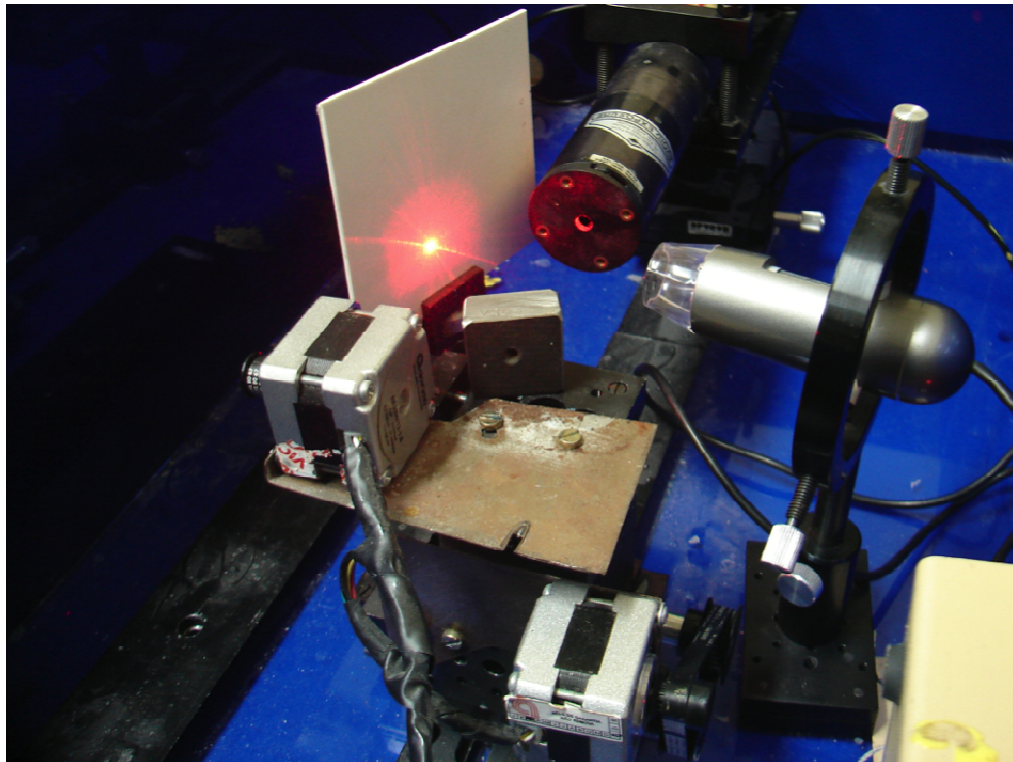


Fonte: Autor

O molde deve estar posicionado sobre a mesa de deslocamento em um ângulo de  $45^\circ$  em relação ao eixo do feixe laser para melhor reflexão (luz espalhada) sobre o mesmo.

A Figura 38 mostra o dispositivo em funcionamento com o laser ligado sendo projetada a reflexão da superfície do molde sobre o anteparo.

Figura 38 – Dispositivo em funcionamento.



Fonte: Autor

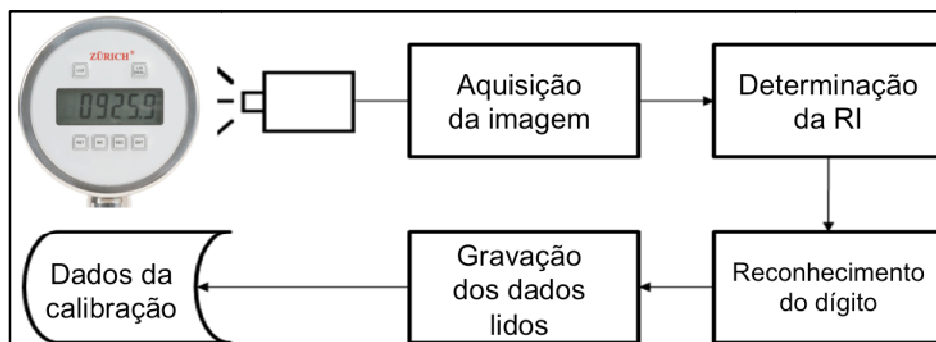
## 4 METODOLOGIAS PROPOSTAS

Neste capítulo são apresentadas as quatro metodologias para realização das tarefas descritas anteriormente. A sequência é dada por: calibração de manômetro digital, calibração de relógio comparador analógico, calibração de peneira grano-volumétrica e por fim a análise dos moldes de injeção para plástico.

### 4.1 CALIBRAÇÃO DE MANÔMETRO DIGITAL

A metodologia proposta para reconhecimento dos dígitos presentes no display de manômetros digitais é dividida em quatro etapas: aquisição da imagem, determinação da região de interesse (RI), reconhecimento dos dígitos e armazenamento dos dados em um banco de dados. Na segunda e terceira etapa descrita é utilizada a mesma técnica de processamento de imagens, casamento de padrões. O primeiro subitem deste item descreve como é realizada a determinação de RI e o segundo como é o reconhecimento do dígito. O esquema do processo é mostrado na Figura 39.

Figura 39 – Diagrama do processo de calibração para manômetro digital.



Fonte: BELAN; ARAÚJO; LIBRANTZ, 2013

#### 4.1.1 Determinação da RI

Na primeira etapa do algoritmo uma imagem colorida é adquirida, com as seguintes dimensões: 640×480 pixels. Esta imagem é adquirida por uma webcam

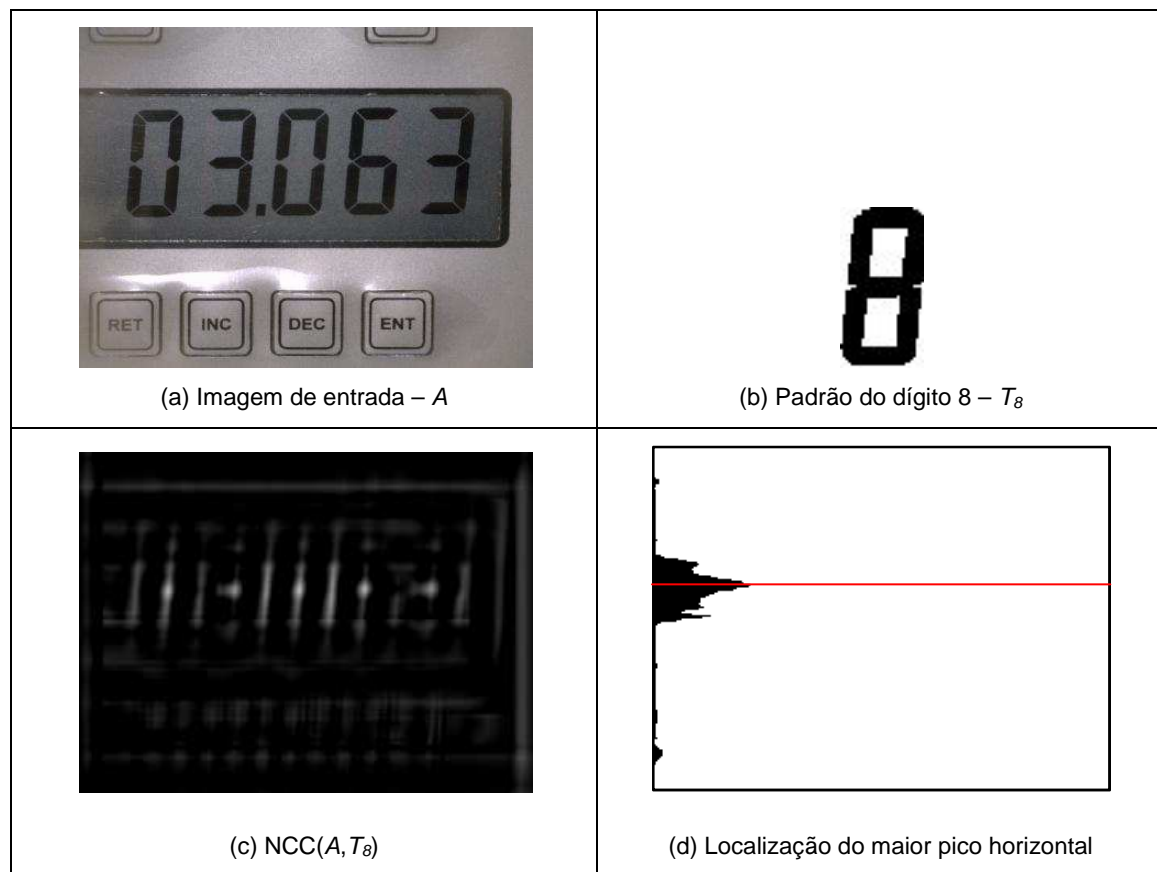
convencional de baixo custo. Após a aquisição a imagem é convertida para níveis de cinza e reduzida em 30% do seu tamanho original. Um exemplo de imagem é mostrado na Figura 40a.

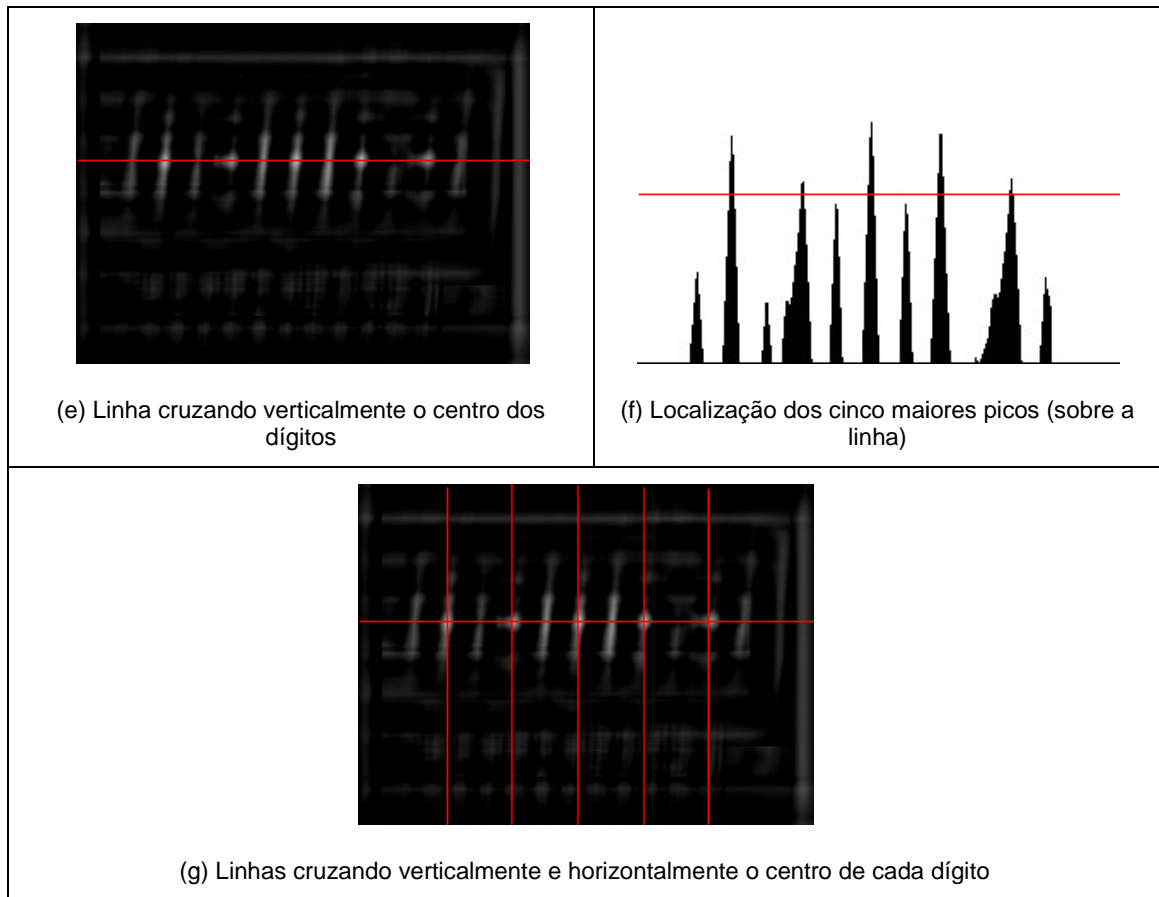
Devido a técnica de casamento de padrões consumir muito tempo computacional, a redução do tamanho da imagem mostrou-se necessária para reduzir a área de busca e acelerar o algoritmo.

Para a determinação da RI, primeiramente é executada a NCC entre a imagem de entrada (Figura 40a) e o padrão do dígito 8 ( $T_8$ ) (Figura 40b), denotado por  $NCC(A, T_8)$ . Este processamento gera uma nova imagem  $C$ , ilustrada na Figura 40c. Para determinar o centro dos dígitos um histograma de projeção horizontal é calculado (Figura 40d) e o maior pico é selecionado, conforme mostrado na Figura 40e.

Em complemento, para determinar um valor limiar  $t$ , é possível selecionar os cinco maiores picos (Figura 40f), que representam o centro de cada dígito, (Figura 40g).

Figura 40 – Processo de determinação da RI.





Fonte: BELAN; ARAÚJO; LIBRANTZ, 2013

Utilizando as coordenadas que marcam o centro de cada dígito, as posições iniciais e finais de cada sub-região  $S_0, \dots, S_4$  são determinadas conforme mostra a Figura 41.

Figura 41 – Detecção das sub-regiões.



Fonte: BELAN; ARAÚJO; LIBRANTZ, 2013

É válido lembrar que o ponto de separação decimal não precisa ser reconhecido pelo fato de que a quantidade de casas decimais deve ser especificada pelo técnico no momento da calibração, e está relacionada à precisão do processo.

#### 4.1.2 Reconhecimento dos dígitos

Nesta segunda etapa do algoritmo os dígitos do display do equipamento são reconhecidos e armazenados em um banco de dados. Para este propósito, o primeiro valor do cálculo da NCC é calculado entre todos os  $S_i$  (Figura 41) e cada padrão  $T_i$  (Figura 42) e o valor máximo de correlação é armazenado em uma matriz  $P$ , contendo cinco linhas (sub-regiões de  $A$ ) e 10 colunas (padrões), como segue (Equação 24):

$$P[l, c] = \text{MAX}[NCC(S, T_c)] \quad (24)$$











onde  $l=0, \dots, 4$  e  $c=0, \dots, 9$ .

O índice  $c$  de maior valor para cada linha na matriz  $P$  indica o número/ dígito mostrado no display do instrumento em análise na posição  $l$ . Então os dados de saída são armazenados em um vetor  $D$  contendo 5 valores inteiros, obtidos da seguinte forma (Equação 25):

$$D[l] = \underset{c=0}{\text{ARGMAX}}^9 [P[l, c]] \quad (25)$$

onde  $c$  é o argumento que maximiza  $P[l, c]$ .

Figura 42 – Modelo dos dígitos utilizados para o reconhecimento.

 (a) $T_0$	 (b) $T_1$	 (c) $T_2$	 (d) $T_3$	 (e) $T_4$
 (f) $T_5$	 (g) $T_6$	 (h) $T_7$	 (i) $T_8$	 (j) $T_9$

Fonte: BELAN; ARAÚJO; LIBRANTZ, 2013

A Tabela 4 ilustra a matriz  $P$  calculada para o reconhecimento dos dígitos do display da Figura 40a.

Tabela 4 – Matriz  $P$  calculada no reconhecimento do display da Figura 40a. Os valores de correlação variam de -1 a 1, porém foram normalizados para o intervalo de 0 a 255 para melhor visualização.

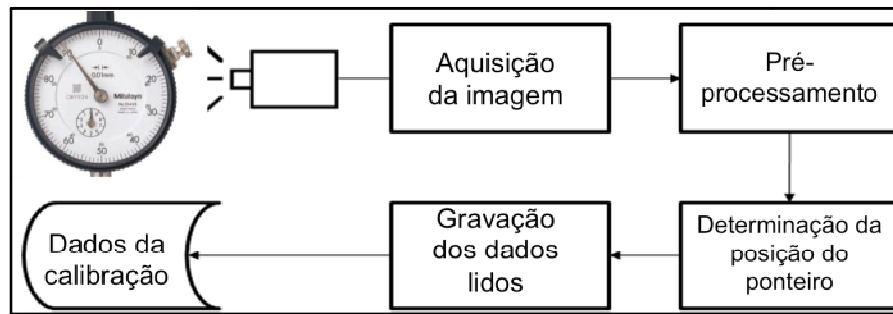
	0	1	2	3	4	5	6	7	8	9
0	<b>215</b>	149	106	122	112	93	148	184	176	127
1	133	144	137	<b>219</b>	103	124	101	163	139	163
2	<b>223</b>	150	104	130	100	112	162	178	181	136
3	146	152	80	126	93	188	<b>227</b>	168	171	125
4	114	150	127	<b>225</b>	110	129	97	162	138	164

Finalmente, os valores que se encontram no vetor  $D$ , representando os dígitos do display, são armazenados em um banco de dados, para posteriormente serem utilizados na emissão do certificado de calibração.

## 4.2 CALIBRAÇÃO DE RELÓGIO COMPARADOR ANALÓGICO

A leitura de um instrumento analógico de ponteiro pode ser determinada de acordo com a relação entre os limites de escala (ANDRIA *et al.*, 2009). O algoritmo proposto usa projeções de linhas radiais e o algoritmo de desenho de linha de Bresenham para detectar a posição do ponteiro. Existem ainda autores que propõem o uso transformadas de *Hough* para a detecção de linhas (SONG *et al.*, 2002). A metodologia proposta para a calibração de relógios comparadores é formada por cinco etapas. Destas cinco apenas será detalhada a etapa de determinação da posição do ponteiro (Figura 43). O pré-processamento desta imagem consiste apenas na conversão de uma imagem RGB para níveis de cinza.

Figura 43 – Diagrama do processo de calibração para relógio comparador analógico.



Fonte: BELAN; ARAÚJO; LIBRANTZ, 2013

#### 4.2.1 Algoritmo de detecção de ponteiro

Os equipamentos analógicos têm como mostrador normalmente um relógio de ponteiros para indicar a medida que está sendo lida. Assim, o algoritmo precisa ser capaz de detectar o ângulo de inclinação do ponteiro para determinar o valor lido com base nas características do instrumento analisado. Para isso, foi usada a idéia de amostragem radial proposta em (KIM; ARAÚJO, 2007; ARAÚJO; KIM, 2011), substituindo o valor médio de níveis de cinza por um somatório dos valores sobre a linha projetada.

Dada uma imagem  $I$ , a amostragem radial  $Ras_I^\lambda(x, y, \alpha)$  pode ser definida como a soma dos pixels de  $I$  localizados na linha radial com um vértice em  $(x, y)$ ,  $\lambda$  representa o comprimento e  $\alpha$  representa a inclinação, conforme descrito pela Equação 26:

$$Ras_I^\lambda(x, y, \alpha) = \int_0^\lambda I(x + t \cos \alpha, y + t \sin \alpha) dt \quad (26)$$

Na prática, a integral deve ser substituída por um somatório e o algoritmo de desenho de linha de Bresenham pode ser usado de forma eficiente para encontrar todos os pixels que pertencem a linha.

Dado o comprimento de  $\lambda$  (de acordo com o instrumento que está sendo analisado) e um conjunto de inclinações angulares  $m(\alpha_0, \alpha_1, \dots, \alpha_{m-1})$ ,  $I$  é radialmente amostrada, gerando um vetor de RI com valores  $m$ , como mostrado na Equação 27:

$$R_i[j] = \text{Ras}_i^\lambda(x_c, y_c, \alpha_j), 0 \leq j < m \quad (27)$$

onde  $(x_c, y_c)$  é o pixel central de  $i$ .

Assim, o ângulo de inclinação do ponteiro ( $a_p$ ) é definido pela Equação 28:

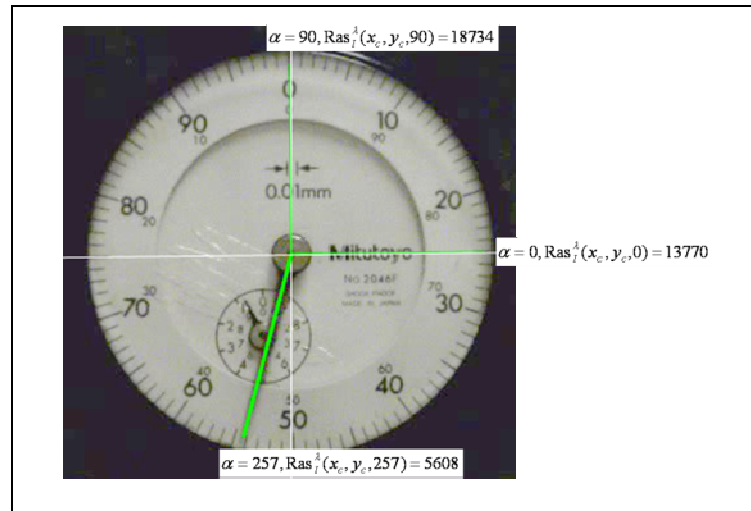
$$a_p = \underset{j=1}{\text{ARGMIN}}^m [R_i[j]] \quad (28)$$

onde  $j$  é o argumento que minimiza  $a_p$ .

Como mencionado anteriormente, usando o valor da inclinação angular do ponteiro é possível determinar o valor mostrado no display do equipamento.

A Figura 44 mostra um exemplo da metodologia proposta onde três diferentes ângulos são utilizados.

Figura 44 – Exemplo da metodologia proposta utilizando três valores de  $\alpha$  ( $0^\circ$ ,  $90^\circ$  e  $257^\circ$ ),  $\lambda=150$ ,  $x_c=175$  e  $y_c=175$ . O tamanho da imagem é  $350 \times 350$  pixels.



Fonte: BELAN; ARAÚJO; LIBRANTZ, 2013

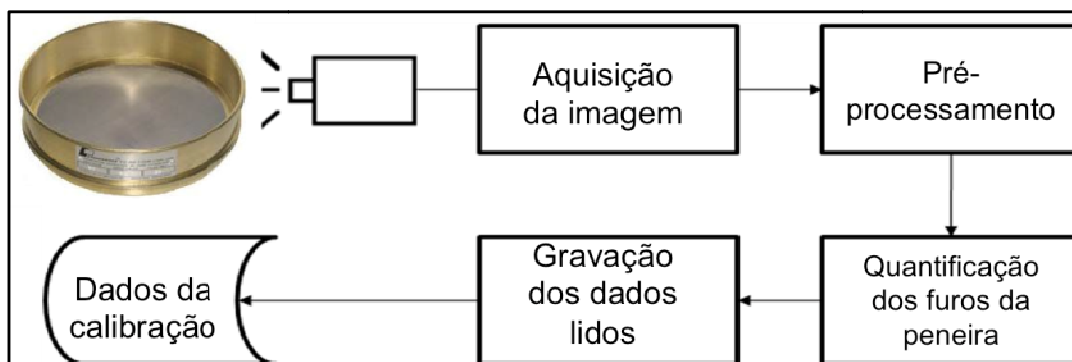
Cabe salientar que, se o ponteiro for claro e o fundo do instrumento for escuro, o ângulo de inclinação será determinado pelo argumento  $j$  que neste caso maximiza  $a_p$ . Essa informação, bem como o diâmetro do instrumento para determinar  $(x_c, y_c)$  e  $\lambda$  precisam estar armazenados em um banco de dados.

### 4.3 CALIBRAÇÃO DE PENEIRA GRANO-VOLUMÉTRICA

A calibração de peneira grano-volumétrica consiste em medir os espaços da trama da peneira, o qual deve ser realizado por amostragem. Esta leitura deve levar em conta cinco regiões distintas da peneira e totalizar pelo menos 600 furos da trama, segundo normas do INMETRO.

O método proposto consiste em adquirir a imagem da trama da peneira por meio do dispositivo descrito. Para tal, utiliza-se a ampliação da câmera em 500x dado o tamanho da trama da peneira que foi utilizada para os testes (38  $\mu\text{m}$  de abertura). Após a aquisição da imagem é executado um pré-processamento que consiste na utilização do algoritmo de limiarização de Otsu, que realiza uma limiarização automática, baseada nos histogramas da imagem. Com a imagem binária de saída é realizado um processo de inversão, onde os pixels que continham o valor 0 (preto) tornam-se 1 (branco) e os com valor 1 passam a ser 0. Isto se faz necessário visto que na saída do Otsu fica apenas a trama da peneira, e o que interessa para análise são seus furos. Ainda como pré-processamento é aplicado um filtro de componentes conexos de 4, a fim de descartar sujeiras na trama bem como alguma falha que possa ocorrer no processo de limiarização de OTSU, visto que a imagem em questão é muito sensível a ruídos, os quais atrapalhariam as medidas da trama. O diagrama das etapas do processo de calibração de peneira grano-volumétrica é mostrado na Figura 45.

Figura 45 – Diagrama do processo de calibração para relógio comparador analógico.



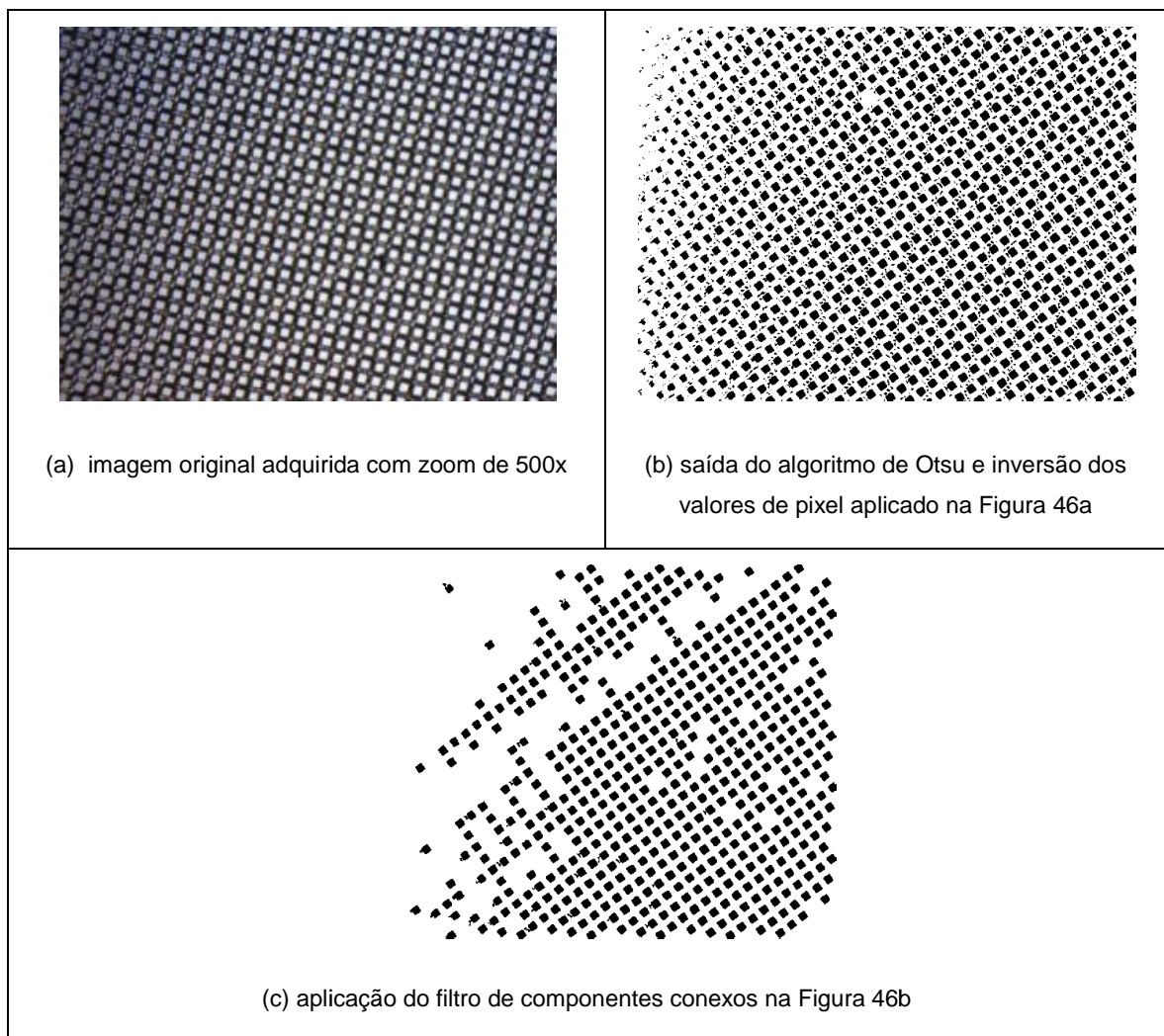
Fonte: Autor

O filtro de componentes conexos consiste no processo de excluir da imagem agrupamentos de componentes conexos que contenham ou mais ou menos que um valor pré-definido de componentes conexos. No trabalho proposto foi utilizado o filtro

de componentes conexos pra remoção de valores considerados inferiores e superiores ao esperado para a peneira, assim descartando todos os possíveis ruídos da imagem, aumentando a qualidade da resposta gerada.

O processo de pré-processamento é mostrado na Figura 46. A Figura 46a mostra o exemplo de uma imagem capturada, enquanto que a 46b representa a aplicação o limiar de Otsu e por fim a 46c mostra o final da etapa de pré-processamento, com a imagem filtrada.

Figura 46 – Etapas do pré-processamento da imagem adquirida da trama da peneira grano-volumétrica.



Fonte: Autor

Vale ressaltar que apesar da impressão de que a imagem mostrada na Figura 46c esteja muito falha, a quantidade de componentes conexos restantes na imagem é uma amostragem aproximadamente 10 vezes maior do que a analisada pelo processo manual.

Finalizado o pré-processamento da imagem, a próxima etapa é a análise dos tamanhos dos furos que são representados na Figura 46c. Este processo consiste em realizar a contagem de componentes conexos de 8 na imagem. Esta contagem na realidade é o processo de medir o tamanho da área de cada componente (furo da peneira), e após esta medida compara-se este valor resultante em pixels, com o valor obtido na calibração do sistema com base na escala padrão (Figura 31). Após realizar o processo de medida do furo é realizada a verificação da espessura do fio da trama da peneira, para isso realizam-se rotações na imagem no intervalo de  $-45^{\circ}$  a  $45^{\circ}$  com passo de  $1^{\circ}$ , a cada rotação realizada, calcula-se um histograma de projeção vertical para a imagem e guarda-se o ângulo de maior soma ao quadrado do histograma. Dessa forma é possível detectar o melhor ângulo para que seja alinhada a trama da peneira. Nessa etapa calcula-se novamente o histograma de projeção para esta imagem e a espessura da trama é a quantidade de pixels entre os picos do histograma de projeção vertical multiplicado pelo valor da escala padrão. Estas etapas de pré-processamento e reconhecimento são extremamente simples e rápidas, o que torna a técnica robusta e muito eficiente.

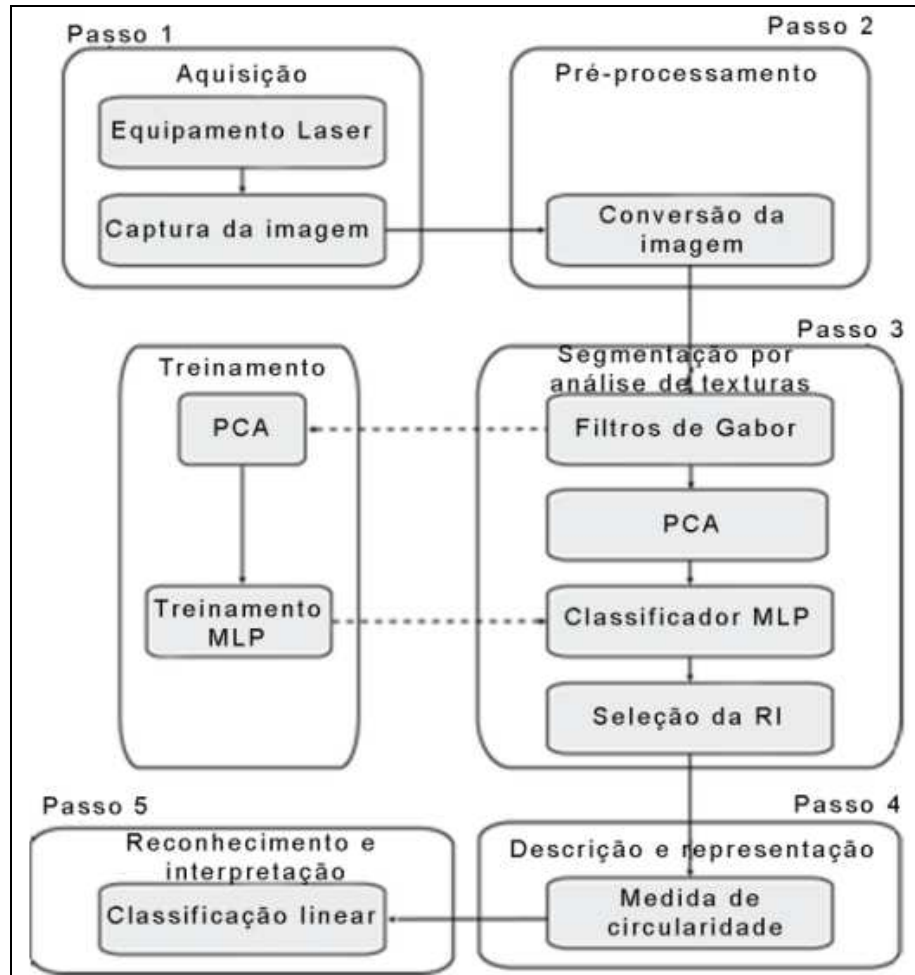
Todos os valores adquiridos durante o processo de calibração das cinco imagens de regiões distintas da peneira são armazenados em um arquivo para gerar posteriormente o certificado de calibração.

#### **4.4 ANÁLISE DA SUPERFÍCIE DE MOLDES PARA INJEÇÃO DE PLÁSTICO**

A abordagem de visão por computador para analisar a superfície dos moldes pode ser descrita da seguinte forma: na primeira etapa é realizada a aquisição de imagens a partir do molde em análise. O próximo passo é o pré-processamento da imagem, que consiste em converter a imagem capturada, originalmente na cor RGB, em níveis de cinza. O terceiro passo é a segmentação de imagens através da análise de textura. Esta etapa envolve filtragem usando filtros de Gabor, a redução do vetor de características usando ACP, classificação da textura utilizando uma RNA do tipo MLP e a seleção da RI. O próximo passo, chamado descrição e representação, consiste na análise de circularidade da RI a partir da terceira etapa.

Na última etapa é realizada a avaliação da qualidade da superfície do molde. Esta sequência de passos é mostrada na Figura 47.

Figura 47 – Diagrama da proposta para avaliação de moldes para injeção de plástico.



Fonte: Autor

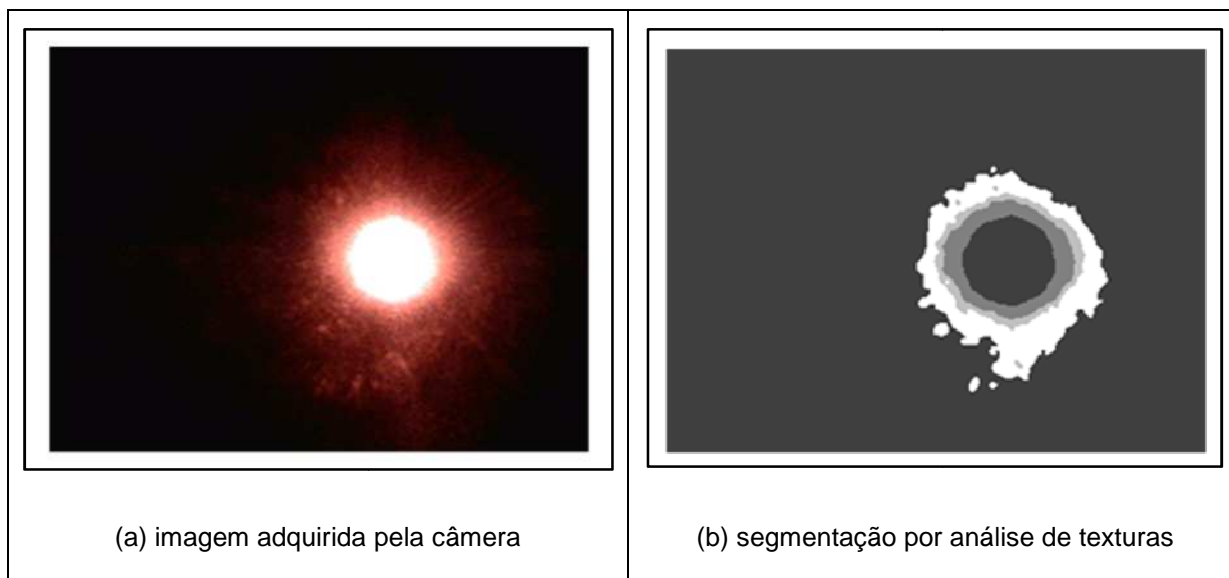
Mais detalhes desse processo são descritos nos próximos subitens.

#### 4.4.1 Segmentação por Análise de Texturas

Na etapa de segmentação por análise de texturas, são utilizadas algumas técnicas descritas na seção 4, sendo o primeiro passo a aplicação do filtro de Gabor, criando um Kernel com 24 descritores, desta forma gerando 24 matrizes de características. Estes 24 descritores devem ser a alimentação do ACP, para que os mesmos sejam reduzidos linearmente para 6 elementos que representem o conjunto completo. Com uma saída reduzida a 6 elementos, é alimentado o classificador, uma MLP, que classifica a região de acordo com a sua característica de textura.

A Figura 48b ilustra o resultado obtido pela segmentação por análise de texturas, lembrando que a imagem original (Figura 48a) é convertida de RGB para níveis de cinza, antes de ser aplicada a segmentação.

Figura 48 – Exemplo de segmentação por análise de textura.



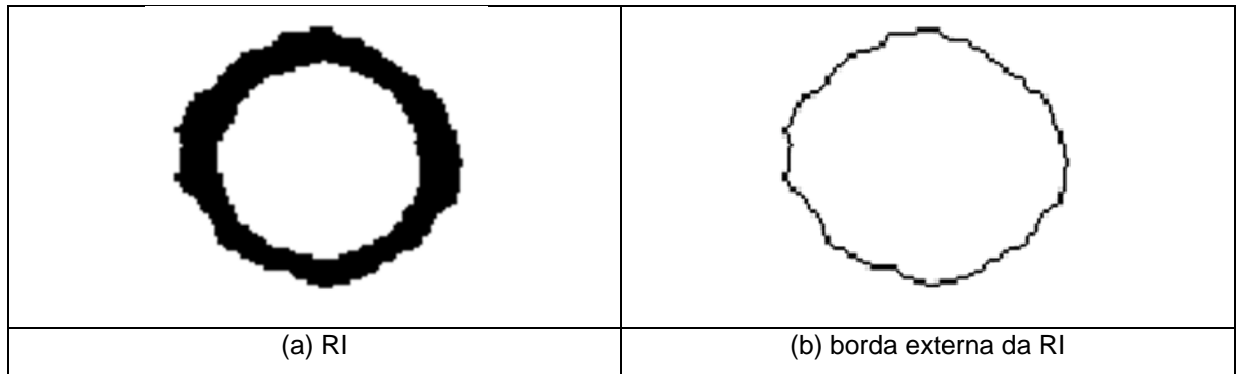
Fonte: Autor

Na imagem segmentada, nota-se quatro classificações: a primeira (cinza mais escuro) é o núcleo e a área externa da imagem; a segunda tonalidade de cinza escuro: é a camada mais próxima do núcleo da imagem; a terceira camada de textura e por fim a última camada de textura da imagem, representada pela região em branco na imagem. Após a realização da segmentação, o próximo passo é determinar a RI da imagem, a qual será analisada posteriormente.

#### 4.4.2 Determinação da RI

Após ser realizada a segmentação, é determinada a RI da imagem. Para isso uma imagem binária é gerada, contendo apenas a segunda classificação de textura da imagem (Figura 49a), descartando assim as 3 camadas restantes da imagem. Essa camada foi selecionada pelo fato de apresentar a maior sensibilidade associada às mudanças na superfície do molde. O próximo passo a ser realizado é a separação da borda externa da RI (Figura 49b), para esta ser utilizada na medida de circularidade. Nesta operação foi utilizada a biblioteca ProEikon (KIM, 2011).

Figura 49 – Exemplo da determinação da RI.



Fonte: Autor

Uma vez determinada a RI, aplica-se uma medida de circularidade conforme descrita no próximo item.

#### 4.4.3 Medida de circularidade proposta

A principal questão para avaliar a qualidade da superfície do molde é determinar quão circular são os componentes conexos obtidos a partir da imagem gerada pela etapa de determinação da RI. Esta medida é obtida da seguinte forma: seja  $X$  um componente conexo e  $C$  o contorno do  $X$ . A distância entre o  $i$ -ésimo pixel  $(x_i, y_i)$ , de  $C$  para o centróide  $(x_c, y_c)$  de  $X$  e a média destas distâncias podem ser definidas pelas Equações 29 e 30, respectivamente:

$$distC_i = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} \quad (29)$$

$$distM = \frac{1}{N} \sum_{i=1}^N distC_i, \quad (30)$$

Onde  $N$  é o número de pixels sobre  $C$ .

Desta forma, o desvio padrão das distâncias dos pixels de  $C$  para  $(x_c, y_c)$  é calculado pela Equação 31:

$$stdC = \sqrt{\frac{1}{N} \sum_{i=1}^N (distC_i - distM)^2} \quad (31)$$

A medida de circularidade proposta é definida pela Equação 32:

$$circularidade = \left( \frac{\max - \min}{\max} \right) stdC, \quad (32)$$

onde *min* e *max* são as menores e maiores *distC<sub>i</sub>* respectivamente. Em outras palavras, a circularidade é caracterizada pelo desvio padrão normalizado das distâncias a partir dos pixels de *C* para (*x<sub>c</sub>*, *y<sub>c</sub>*). Esta normalização penaliza os componentes conexos com grandes diferenças entre distâncias menores e maiores, que podem indicar um problema na superfície do molde.

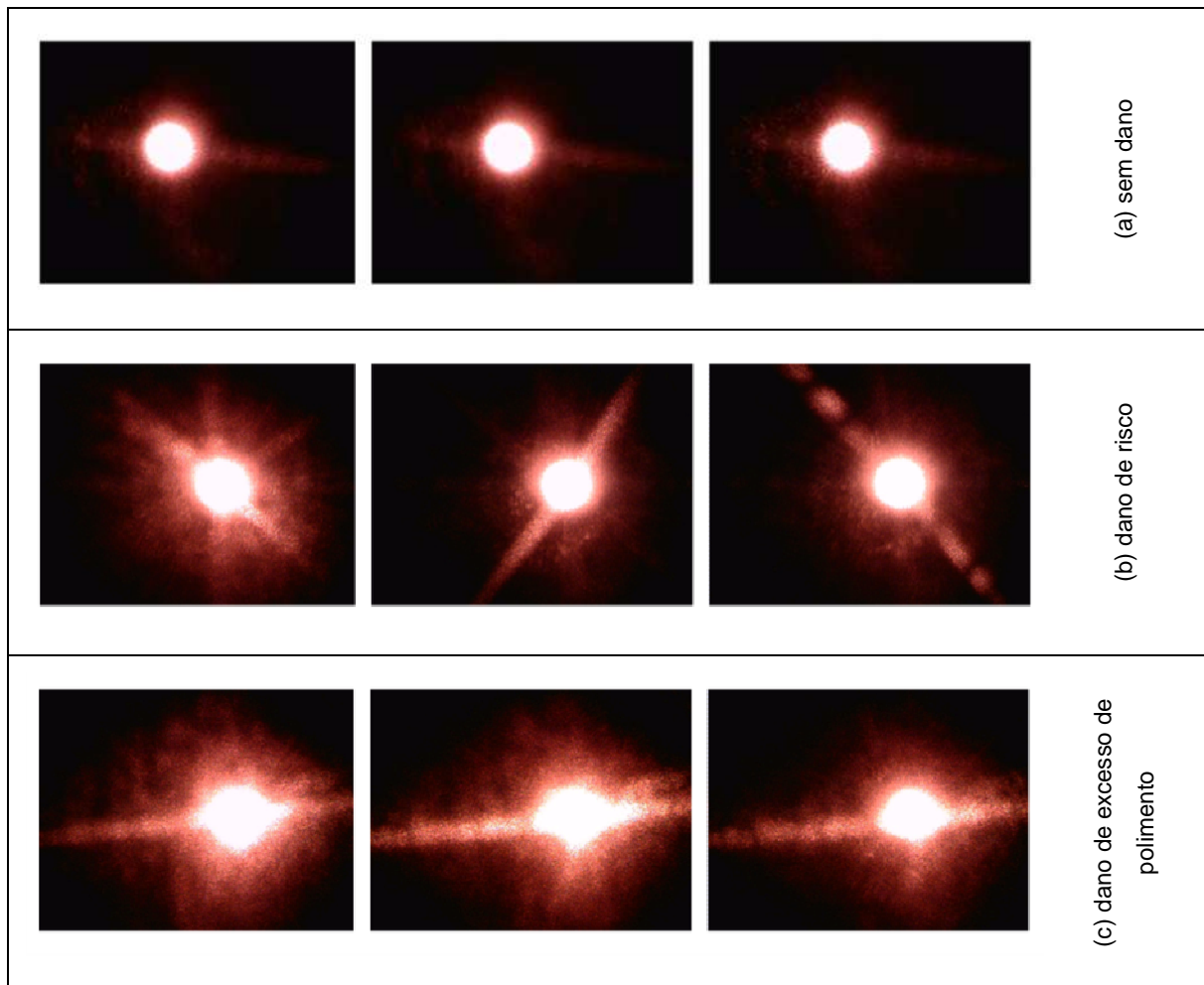
#### 4.4.4 Reconhecimento e interpretação

Finalmente, a circularidade da borda externa da RI é calculada e o valor obtido é comparado com o limiar estabelecido (o limiar é calculado pela média ponderada dos valores obtidos e multiplicado por 2 (fórmula obtida empiricamente)). Este procedimento é realizado a cada amostra de imagem adquirida do molde e de acordo com sua medida de circularidade é possível avaliar se existe algum dano na superfície e, se existir, qual é o tipo do dano: um risco ou excesso de polimento (“casca de laranja”).

Se a circularidade da RI for maior que o limiar, isso indica que a região da superfície tem algum dano. Assim, este deve ser caracterizado como excesso de polimento “casca de laranja” ou um risco. A fim de determinar estas duas classes de danos, a abordagem proposta leva em conta os valores de circularidade dos vizinhos do ponto em análise. O dano casca de laranja tem altos picos de circularidade, normalmente excedendo 15 pontos. Caso este valor seja menor que 15 pontos e maior que o limiar estabelecido, o dano é considerado como risco. Outras características como a área afetada pelo dano, pode também ser usada para classificar a região de superfície. Por exemplo, a área afetada por “casca de laranja” é geralmente maior que a área afetada por um risco (SHIMIZU; FUJI, 2003; HIPPENSTIEL, 2001).

A Figura 50 mostra um conjunto de nove imagens adquiridas com as três condições possíveis para análise nesse experimento.

Figura 50 – Exemplos de imagens adquiridas em regiões diferentes da superfície do molde.



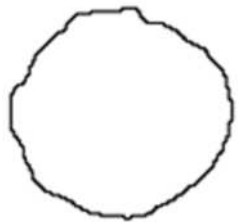


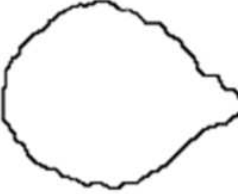


Fonte: Autor

Como pode ser visto na Figura 50, características diferentes são observadas em três amostras "a", "b" e "c". Onde "a" representa uma região boa e as duas outras regiões apresentam danos. Nas amostras "a" e "b" o centro do foco laser apresenta como característica uma uniformidade, diferentemente de "c" que apresenta deformação neste centro.

Dois exemplos da abordagem proposta de visão computacional com a medida de circularidade são mostrados na Figura 51. Uma delas é uma imagem adquirida sem danos e a segunda tem um dano de risco.

Figura 51 – Exemplo da metodologia proposta com a abordagem de visão computacional com a medida de circularidade proposta.

			circularidade 1,37
			circularidade 11,63
(a) imagem adquirida	(b) imagem segmentada por análise de texturas	(c) borda externa da RI	

Fonte: Autor

Como pode ser visto, a medida de circularidade na primeira imagem da Figura 51 (valor 1,37) tem um valor menor quando comparado com a segunda (valor 11,63), indicando assim uma melhor qualidade na superfície do molde sobre a primeira posição. Para o desenvolvimento da metodologia proposta para análise de superfície de moldes para injeção de plástico, foi desenvolvido um programa em linguagem C++ utilizando as duas bibliotecas de processamento de imagens já descritas anteriormente, sendo a OpenCV no processo dos filtros de Gabor e a ProEikon na análise de circularidade. Todas as outras rotinas como a MLP e aplicação de ACP, foram aplicadas sem a utilização de outras bibliotecas, bem como o controle da automação dos motores e aquisição das imagens.

## 5 RESULTADOS E DISCUSSÃO

Nesta seção são discutidos detalhes dos experimentos para avaliar o desempenho dos algoritmos propostos bem como os resultados obtidos para cada proposta.

### 5.1 CALIBRAÇÃO DE MANÔMETRO DIGITAL

Para avaliar a robustez do algoritmo proposto, foram realizados experimentos utilizando um conjunto de 50 imagens conforme descrito no item 3.1. Destas imagens algumas foram afetadas por mudanças de iluminação (Figura 52a e 52b); ruído (Figura 52c) e compressão JPEG (Figura 52d). Neste experimento todos os 250 dígitos foram perfeitamente reconhecidos, sem nenhum falso negativo ou falso positivo, mesmo na presença de alguns dígitos ligeiramente visíveis (Figura 52b, por exemplo).

Figura 52 – Exemplo de imagens afetadas.



Fonte: BELAN; ARAÚJO; LIBRANTZ, 2013

Quanto ao tempo de processamento, o algoritmo levou em média 0,3 segundos para reconhecimento dos dígitos apresentados no display do equipamento. Estes resultados foram obtidos utilizando um computador com

processador Pentium Core2Duo 1,4 GHz. Entretanto, este tempo poderia ser ainda menor, se na segunda etapa do algoritmo ao invés de executar casamento de padrões em todos os pixels de cada subárea, fosse realizado apenas em alguns pixels em torno dos picos que representam o centro dos dígitos.

Esta redução no tempo de processamento pode permitir o uso do algoritmo em outros processos de calibração de instrumentos que exijam uma resposta mais ágil, visto que muitos instrumentos de medição atualmente contam com display digital e não possuem uma interface de comunicação com um PC.

Vale ressaltar ainda que este processo aplicado na calibração de manômetros gerou uma redução de aproximadamente 30% no tempo total gasto no processo de calibração e emissão do certificado de calibração. O tempo de processo antes da automação estava em torno de 15 minutos, ficando próximo de 11 minutos após a automação.

## 5.2 CALIBRAÇÃO DE RELÓGIO COMPARADOR ANALÓGICO

Para validar o algoritmo proposto foram utilizados três relógios comparadores de marcas diferentes. Em todos os casos o ponto central do relógio foi determinado com base no suporte que foi desenvolvido para fixar o instrumento e algumas informações armazenadas previamente em um banco de dados, com as características dos instrumentos, como tamanho, cor do ponteiro, capacidade e divisão. Este ponto central é indicado pelas linhas brancas que cruzam as Figuras 53a-c.

A resposta produzida pelo algoritmo são números inteiros, visto que estes equipamentos analógicos não podem ser utilizados para cálculos com valores intermediários, apenas têm validade os valores que cruzam os traços, de modo que outro formato não pode ser aceito pelos critérios de calibração existentes autorizados pelo INMETRO. Em todos os experimentos foi estabelecido o parâmetro  $m = 360$  para inicialização dos ângulos. Porém foram realizados testes com outros valores como 180 e 720 os quais não foram satisfatórios, por falta de precisão na detecção do ponteiro (para  $m=180$ ). No caso de  $m=720$ , foi necessário mais tempo de processamento sem ganho no processo.

O processo de calibração pelo algoritmo proposto não requer um controle das condições ambientais de iluminação, uma vez que com esta técnica é possível detectar o ponteiro em diferentes condições de iluminação. No entanto, durante a aquisição de imagens deve ser evitada a iluminação que pode favorecer o aparecimento de sombras, podendo causar falsos positivos. Todavia, outros cuidados como temperatura e umidade do ambiente devem ser respeitados segundo regras pré-estabelecidas pelo INMETRO. Na Figura 53 são ilustrados alguns quadros capturados, cada um mostrando a posição do ponteiro detectado.

Figura 53 – Resultado do algoritmo de detecção de ponteiro.



Fonte: BELAN; ARAÚJO; LIBRANTZ, 2013

Os testes foram realizados com o ponteiro do relógio comparador em movimento e apenas os valores múltiplos de 10 foram considerados para o armazenamento em um banco de dados, para serem usados posteriormente na emissão do certificado do instrumento.

Vale ressaltar que a agregação de um medidor digital ao protótipo construído, tornaria possível a leitura do valor real do instrumento no ponto analisado, assim destacando o erro do instrumento em análise. Porém este equipamento não foi adquirido e adicionado ao protótipo devido ao custo envolvido em sua compra. Uma grande vantagem no uso de visão computacional neste tipo de equipamento, é o fato de conseguir evitar erros de paralaxe, comum de ocorrer e um problema natural da visão humana, dada a dificuldade de sempre olhar para o instrumento do mesmo ponto de perspectiva. Este processo de calibração automatizado gerou uma redução de 30 minutos para aproximadamente 7 minutos, tendo assim um ganho de mais 70% no processo completo.

### 5.3 CALIBRAÇÃO DE PENEIRA GRANO-VOLUMÉTRICA

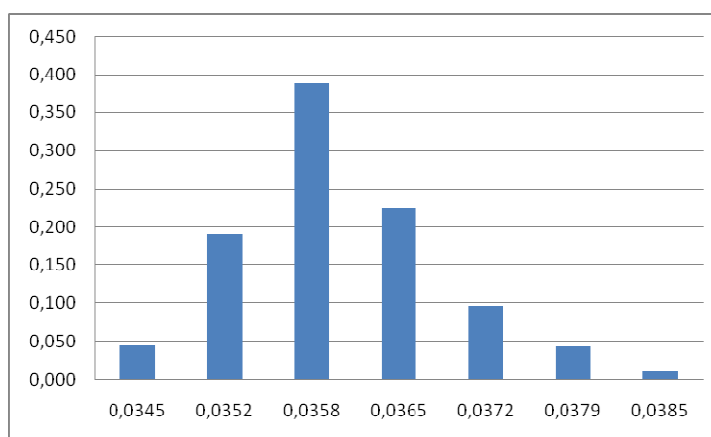
O método proposto analisa cerca de 600 furos por região, perfazendo assim um total de 3000 furos em média por peneira, garantindo desta forma uma maior amostragem do que o método tradicional, realizado atualmente por projetor de perfil e com as medidas realizadas de forma manual ou eletrônica para projetores computadorizados. Apesar do equipamento ser eletrônico, este processo é realizado com o auxílio do mouse, tornando o trabalho repetitivo e cansativo.

Uma calibração no método tradicional necessita de pelo menos 3 horas de trabalho do técnico realizando as medidas individuais dos furos, enquanto que quando aplicada a metodologia proposta este processo foi reduzido para cerca de 2min e 30s de processamento. Contando com o tempo de preparo e ajuste de foco da câmera, o processo completo fica em torno de 5 minutos, representando assim em média 97% de redução de tempo.

Os resultados após analisados pelo sistema proposto foram comparados aos obtidos pelo método tradicional e constatou-se que ambos estão gerando a mesma resposta, garantindo assim a fidelidade dos dados e exatidão das medidas.

A Figura 54 mostra o histograma de frequência dos valores de abertura lidos durante a calibração da peneira, lembrando que a medida do furo da peneira é de  $38\mu\text{m}$ . Mesmo a peneira nova possui um erro como pode ser visto na amostragem, este erro é permitido desde que esteja dentro dos limites exigidos pelas normas de fabricação.

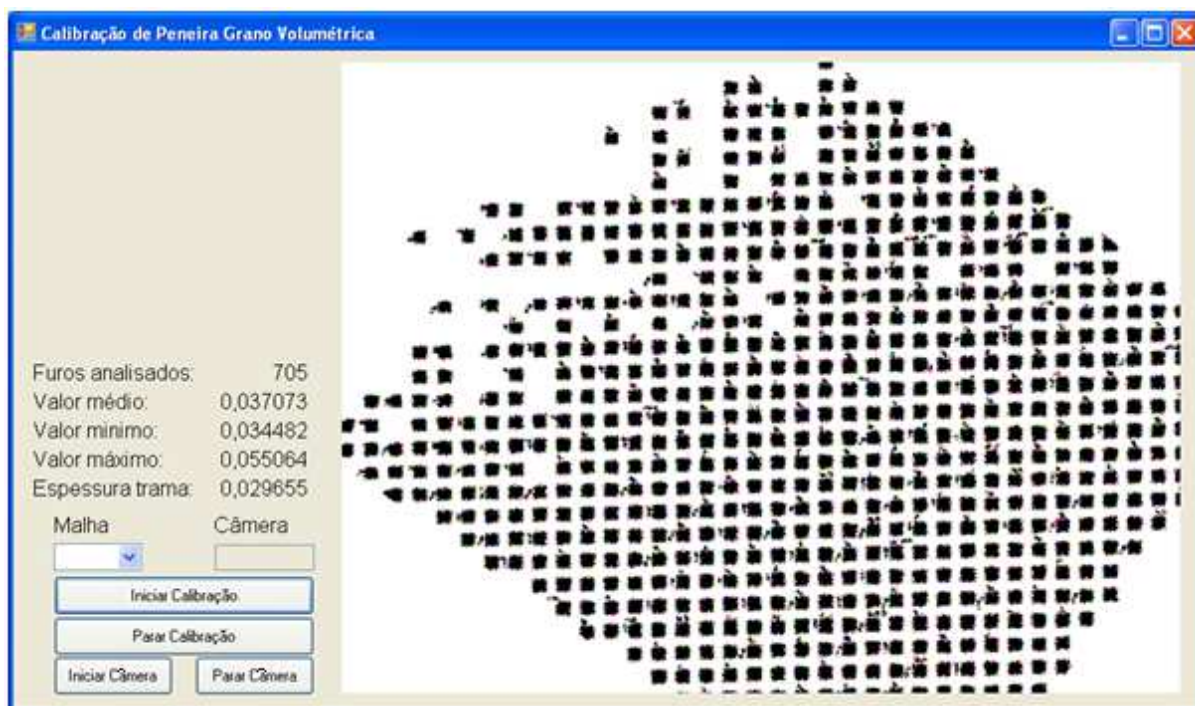
Figura 54 – Histograma de frequência dos valores lidos



Fonte: Autor

A análise completa da peneira tem a saída como mostrada na Figura 55.

Figura 55 – Tela com resultado de saída do sistema de calibração de peneira.



Fonte: Autor

Como mostrado o resultado médio do sistema tem um desvio de 1  $\mu\text{m}$ , o que está dentro dos parâmetros descritos anteriormente, validando desta forma os resultados e a aplicabilidade do sistema em um ambiente de trabalho.

Ressalta-se ainda que estes valores foram obtidos e comparados em um laboratório de calibração com acreditação do INMETRO.

#### 5.4 ANÁLISE DA SUPERFÍCIE DE MOLDES PARA INJEÇÃO DE PLÁSTICO

Para essa finalidade, foram realizados teste com cinco amostras de moldes, como descrito na seção 3. A Tabela 5 apresenta uma amostragem dos resultados obtidos com base na medida de circularidade para cinco dos moldes com os seus limiares calculados.

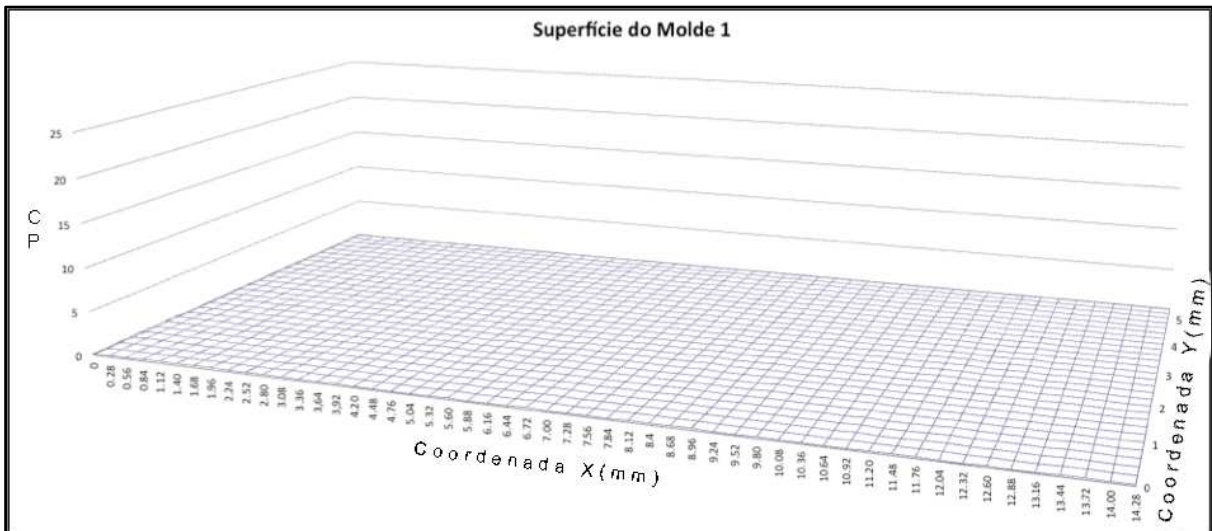
Tabela 5 – Amostragem da medida da circularidade proposta (CP) na superfície dos cinco moldes.

Molde 1 (Limiar 0,76925)		Molde 2 (Limiar 1,49298)		Molde 3 (Limiar 1,52754)		Molde 4 (Limiar 30,22640)		Molde 5 (Limiar 83,49200)	
CP	Diferença CP e Limiar	CP	Diferença CP e Limiar	CP	Diferença CP e Limiar	CP	Diferença CP e Limiar	CP	Diferença CP e Limiar
0,47751	-0,29174	12,73738	11,24440	5,53130	4,03832	20,21066	-10,04574	44,15082	-39,34119
0,36971	-0,39954	26,84712	25,35414	4,08166	2,58867	18,73492	-11,52149	43,54587	-39,94614
0,42779	-0,34146	10,16563	8,67265	3,50742	2,01443	18,85344	-11,40296	43,39118	-40,10083
0,47897	-0,29028	0,42493	-1,06806	2,70519	1,21221	20,05621	-10,20020	43,02007	-40,47194
0,42274	-0,34651	2,09265	0,59967	0,52943	-0,96355	19,22858	-11,02782	42,48246	-41,00955
0,31163	-0,45762	10,15637	8,66339	1,18029	-0,31269	13,26112	-16,99528	42,34187	-41,15014
0,32796	-0,44129	0,69750	-0,79548	13,29893	11,80595	13,98935	-16,26705	41,94157	-41,55044
0,40617	-0,36308	0,60700	-0,88598	17,26165	15,76867	16,65179	-13,60461	42,56937	-40,92264
0,40340	-0,36585	0,50700	-0,98598	0,46930	-1,02368	14,57716	-15,67925	41,48375	-42,00826
0,38933	-0,37992	0,47904	-1,01394	0,47714	-1,01584	10,92652	-19,32988	36,26704	-47,22497
0,27214	-0,49711	0,62861	-0,86437	3,41476	1,92178	15,43334	-14,82306	39,56946	-43,92255
0,39728	-0,37197	1,20752	-0,28546	0,38647	-1,10651	18,61149	-11,64491	41,02795	-42,46406
0,30596	-0,46329	1,19058	-0,30241	3,06337	1,57038	19,22577	-11,03064	39,95356	-43,53845
0,27753	-0,49172	1,04628	-0,44670	2,05000	0,55702	15,62999	-14,62641	39,47735	-44,01466
0,45059	-0,31867	0,55810	-0,93488	0,41931	-1,07367	15,88027	-14,37613	39,30619	-44,18582
0,30608	-0,46317	0,40621	-1,08677	0,49005	-1,00293	17,28968	-12,96672	39,77925	-43,71276
0,29454	-0,47471	3,75326	2,26028	1,61529	0,12231	11,91887	-18,33754	39,79282	-43,69919
0,52575	-0,24350	1,59658	0,10360	3,54973	2,05675	16,17722	-14,07918	39,72603	-43,76598
0,57200	-0,19725	0,51738	-0,97560	8,14531	6,65233	18,41234	-11,84407	39,34056	-44,15144
0,27689	-0,49236	0,53360	-0,95938	5,60801	4,11503	18,06032	-12,19609	36,38836	-47,10365

Como pode ser visto na Tabela 5, todos os valores da diferença entre Circularidade e o Limiar para o molde 1, 4 e 5, são menores que zero, o que significa uma homogeneidade na superfície, sem a presença de danos. Um ponto importante na análise do molde 1 são os valores da Circularidade, onde todos eles são baixos, indicando um tipo de polimento mais preciso, de acordo com especificações de blocos de medição padrão que possui uma superfície perfeita.

A Figura 56 é a representação da superfície do molde 1 após a análise de circularidade para todo o molde.

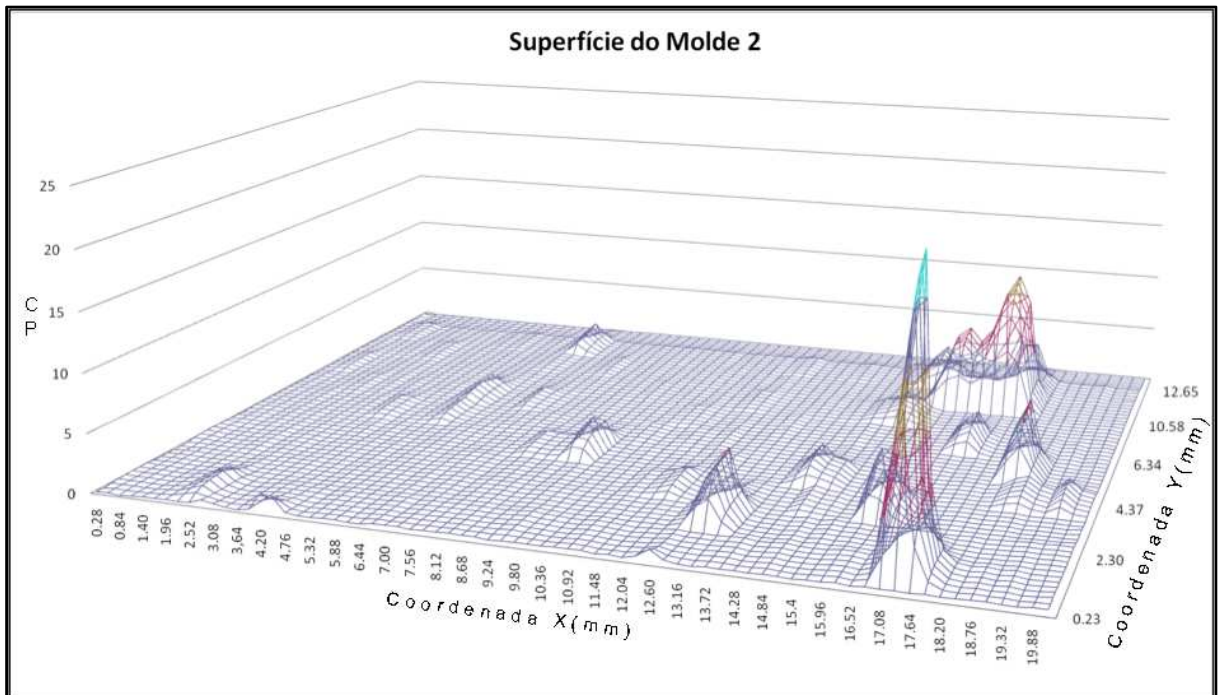
Figura 56 – Representação da superfície do molde 1 gerado pela metodologia proposta. O eixo X e Y estão representados em mm e o eixo Z é a medida de circularidade (CP).



Fonte: Autor

A uniformidade obtida no primeiro molde de amostra não foi observada nas amostras dois e três. Já a representação da superfície dos moldes 4 e 5 são muito similares a esta. A única diferença é que a uniformidade não está próxima a zero e sim em um nível acima. As amostras dois e três possuem problemas conforme descrito na Tabela 2. Em ambos os casos, “casca de laranja” e riscos podem ser encontrados nas superfícies, de acordo com a análise de circularidade proposta. Ainda na Tabela 5, são apresentados os valores de amostragem da circularidade aplicada para os moldes dois e três, respectivamente. Nesses casos, os valores de circularidade para estes dois moldes possuem grande variação. No entanto, esta não uniformidade não compromete a qualidade da superfície toda, mas sim de regiões. Na amostra 2, o limiar calculado foi de 1,49298 e de 1,52754 para a amostra 3 (média ponderada multiplicada por 2). Estes valores de limiar são maiores quando comparados à primeira amostra, principalmente devido ao tipo de acabamento de sua superfície. A Figura 57 representa a superfície do molde dois obtida pela análise de circularidade.

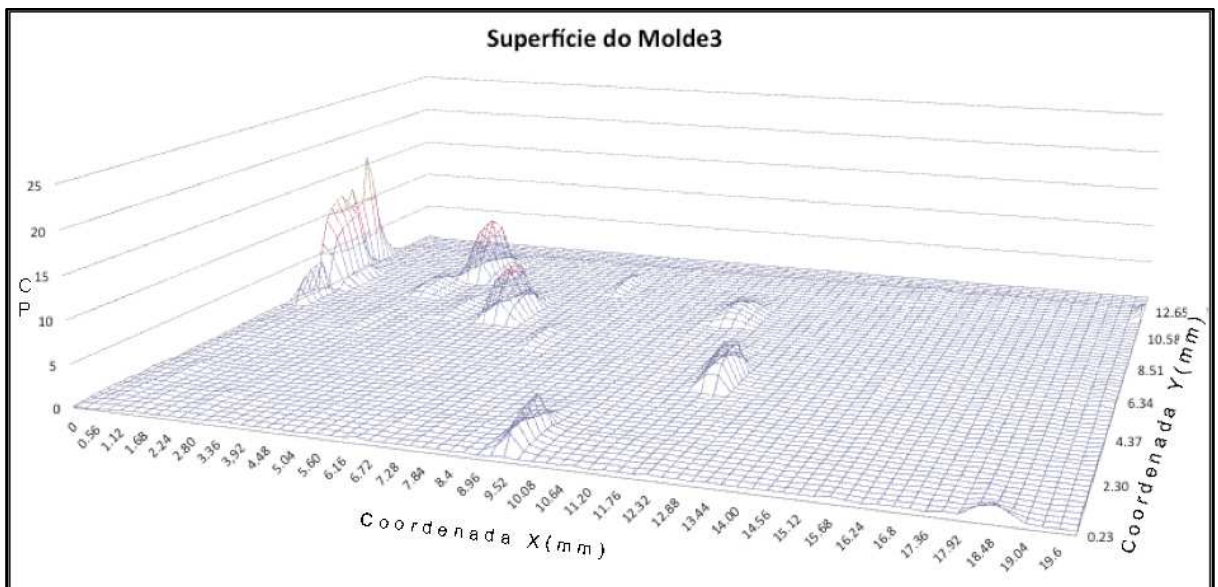
Figura 57 – Representação da superfície do molde 2 gerada pela metodologia proposta. O eixo X e Y estão representados em mm e o eixo Z é a medida de circularidade (CP).



Fonte: Autor

A Figura 58 representa o molde de amostra número três. Como mostrado, essa amostra não possui uma superfície tão irregular quanto a do molde dois, representado pela Figura 57.

Figura 58 – Representação da superfície do molde (exemplo 3) gerada pela metodologia proposta. O eixo X e Y estão representados em mm e o eixo Z é a medida de circularidade (CP).

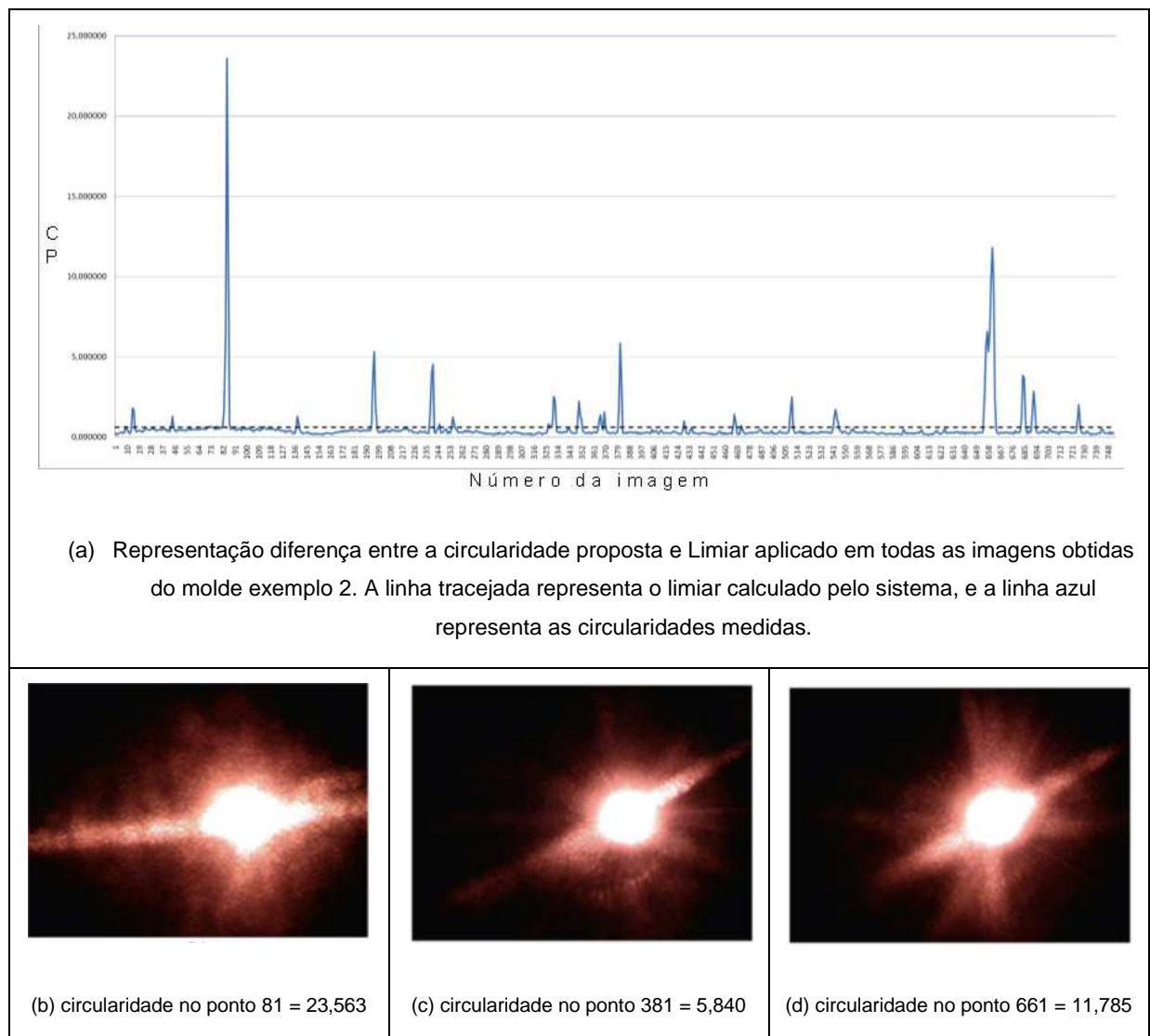


Fonte: Autor

A superfície dos moldes 2 e 3, polidas manualmente e adicionado a eles danos proposital, como pode ser visto, não apresentam uma homogeneidade no polimento como as outras amostras. A classificação da região do molde que está em análise depende do limiar e da identificação do tipo de dano, representado pelas irregularidades, e deve ser levado em conta o comportamento dos vizinhos e o valor calculado pela medida de circularidade. A análise do segundo molde mostra que cerca de 10% da superfície do molde possui danos, enquanto que na terceira amostra este valor corresponde a 7,6% da superfície.

Para explicar o funcionamento do limiar aplicado ao conjunto de imagens, Figura 59 apresenta esta classificação no segundo molde.

Figura 59 – Exemplo de classificação da circularidade proposta no molde 2.



Fonte: Autor

Na Figura 59a, pode-se ver que os valores de circularidade excedem o valor do limiar calculado, caracterizando as regiões danosas. Os três picos mais altos foram obtidos após a aplicação da medida de circularidade nas Figuras 59b, 59c e 59d.

O pico mais elevado mostrado na Figura 59a está relacionado com a Figura 59b e tem um valor alto de circularidade, o que poderia caracterizar uma região com dano “casca de laranja”. Além disso, outros dois picos mais elevados gerados a partir das Figuras 59c e 59d, sugerem danos de risco.

O tempo de processamento de análise para cada imagem é de aproximadamente 1,1 segundo, o que viabilizou o uso desta técnica em tempo real. Os testes foram realizados com um computador com processador Intel Core2Duo 2,0 GHz e 1 Gb de RAM, o que sugere que o uso de um processador mais rápido, utilizado nas máquinas atuais, leve a redução considerável no tempo. Outra possibilidade é a implementação de processamento paralelo.

## 6 CONCLUSÕES

Neste trabalho foram apresentadas técnicas de visão computacional aplicadas na área metrológica, visando automatização de processos de calibração e inspeção. Três automações de calibração de instrumentos foram propostas, uma para automatização de calibração de manômetros digitais, a segunda para calibração de relógio comparador analógico e a terceira para calibração de peneira grano-volumétrica, todas para dispositivos que não possuem interface de comunicação.

Também foram desenvolvidos três dispositivos para auxiliar os experimentos, sendo eles: dispositivo para calibração de relógio comparador analógico; dispositivo para calibração de peneira grano-volumétrica; aparato para avaliação da superfície de moldes para injeção de plástico. Para as três automações foram utilizados motores de passo interfaceado à porta paralela do PC.

Diferentemente da maioria das abordagens encontradas na literatura, os algoritmos propostos para a primeira e segunda abordagem, não descartam as informações de níveis de cinza das imagens, bem como não utilizam operações como a detecção de bordas e segmentação/ binarização. Assim, a robustez destes algoritmos é aumentada.

No experimento sobre manômetro digital, todo o conjunto de dígitos de todas as 50 imagens foram reconhecidas corretamente, mesmo considerando as imagens afetadas por condições de iluminação, ruído e compressão JPEG. O algoritmo levou cerca de 0,3 segundo, em média, para o processamento de cada imagem. O desempenho mostra que esta técnica pode ser aplicada com eficácia na calibração de instrumentos de medição que possuam displays digitais, aumentando assim sua aplicabilidade em laboratórios de calibração.

Com relação ao algoritmo para relógios comparadores analógicos, os testes foram realizados utilizando 5.400 imagens (frames extraídos de seis vídeos realizados dos relógios em movimento). Neste caso, todos os frames analisados foram corretamente reconhecidos. Este algoritmo apresentou alta precisão e desempenho, uma vez que foi capaz de analisar cerca de 90 imagens por segundo. A aplicação do método proposto, permitiu uma redução considerável no tempo

médio do processo de calibração e o seu desempenho torna possível sua aplicação em laboratórios de calibração.

A metodologia proposta para calibração de peneiras grano-volumétricas mostrou-se eficaz no processo de reconhecimento e medição dos furos, apresentando a maior redução de tempo no processo de calibração e emissão do certificado, com um ganho de aproximadamente 97,0% do tempo gasto pelas técnicas atuais de calibração, tornando-se assim, uma alternativa muito interessante a ser implementada nos laboratórios de calibração.

Os resultados obtidos para o algoritmo de avaliação das superfícies de moldes para injeção de plástico, mostraram que a metodologia proposta pode ser aplicada a superfícies lisas de moldes para o mapeamento de sua superfície, detectando as regiões danosas e atribuindo os tipos de danos (excesso de polimento ou risco) para cada região. A utilização do feixe laser visível com a adição de uma lente óptica para melhorar o foco, provou ser uma boa combinação para projetar imagens de forma adequada sobre o anteparo, etapa essencial do processo. Com base nestes resultados, pode-se inferir que esta abordagem tem aplicabilidade na indústria visando reduzir o custo e o tempo gasto nas tarefas de inspeção da superfície de moldes.

Em trabalhos futuros pretende-se agregar os algoritmos propostos em sistemas para trabalhar em tempo real e para lidar com uma grande variedade de instrumentos analógicos e digitais, bem como a construção dos hardwares específicos para cada tipo calibração automática de instrumentos. Pretende-se ainda ampliar a análise da superfície dos moldes para moldes texturizados.

## REFERÊNCIAS

ALEGRIA, F.C.; SERRA, A.C. Computer vision applied to the automatic calibration of measuring instruments. **Measurement**. v. 28, p. 185–195, 2000.

ALEGRIA, F.C.; SERRA, A.C. Automatic Calibration of Analog and Digital Measuring Instruments Using Computer Vision. **IEEE Transactions on Instrumentation and Measurement**. v. 49, p. 94–99, 2000.

ANDRIA, G.; CAVONE, G.; FABBIANO, L.; GIAQUINTO, N.; SAVINO, M. Automatic Calibration System for Digital Instruments Without Built-In Communication Interface. In: **Anais do XIX IMEKO World Congress**, p. 857–860, 2009, Lisbon.

ANGELO, N.P.; HAERTEL, V.; Investigação com respeito a aplicação de filtros de gabor na classificação supervisionada de imagens digitais. In: **Anais X SBSR**, p. 1193–1200, 2001, Foz do Iguaçu.

ANTON, H.; RORRES C., **Álgebra Linear com Aplicações**. Porto Alegre: Bookman, 2004.

ARAUJO, S.A. **Casamento de Padrões em Imagens Digitais Livre de Segmentação e Invariante Sob Transformações de Similaridade**, 2009, Tese (Doutorado) – Escola Politécnica, Universidade de São Paulo, SP, 2009.

ARAÚJO, S. A.; BELAN, P. A.; LIBRANTZ, A. F. H. A template matching based algorithm applied to automatic calibration of digital measuring instruments without built-in communication interface. In: **XLIII SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL**, 2011, Ubatuba, São Paulo. p. 1-9

ARAÚJO, S.A.; KIM, H.Y. Ciratefi: An RST-invariant template matching with extension to color images. **Integrated Computer-Aided Engineering**. v. 18, p. 75–90, 2011.

BEALL, G. By Design: Part design 109 – Textured finishes. **IMMNE Magazine**, 2000.

BELAN, P.A.; ARAÚJO, S.A, LIBRANTZ, A.F.H. Técnicas de visão computacional aplicadas no processo de calibração de instrumentos de medição com display numérico digital sem interface de comunicação de dados. **EXACTA**. v. 10. p. 82–91, 2012.

BELAN, P.A.; ARAÚJO, S.A, LIBRANTZ, A.F.H. Segmentation-free approaches of computer vision for automatic calibration of digital and analog instruments. **Measurement**. <http://dx.doi.org/10.1016/j.measurement.2012.06.005>. 2013

Bloodshed, **Dev-C++**. BloodshedSoftware, 2012. Disponível em <http://www.bloodshed.net>. Acesso em março de 2012.

BOUGON, M. G. Congregate cognitive maps: a unified dynamic theory of organization and strategy. **Journal of Management Studies**. v. 29, p. 369–389, 1992.

BOUJELBENE, M.; MOISAN, A.; TOUNSI, N.; BRENIER, B. Productivity enhancement in dies and molds manufacturing by the use of C1 continuous tool path. **International Journal of Machine Tools & Manufacture**. v. 44, p. 101–107, 2003.

BRESENHAM, J.E. Algorithm for computer control of a digital plotter. **IBM Systems Journal**. v. 4, p. 25–30, 1965.

BRESENHAM, J.E. A linear algorithm for incremental digital display of circular arcs. **Communications of the ACM**. v. 20, p. 100–106, 1977.

BRESENHAM, J.E. Incremental Line Compaction. **THE COMPUTER JOURNAL**. v. 25, p. 116–120, 1982.

CHAN, W.L.; FU, M.W.; LU, J. An integrated FEM and ANN methodology for metal-formed product design. **Engineering Applications of Artificial Intelligence**. v. 21, p. 1170–1181, 2008.

CONCI, A.; AZEVEDO, E.; LETA, F. R. **Computação Gráfica: Teoria e Prática**. v. 2, Rio de Janeiro: Campus, 2008. 407p.

EDELSTAHL, B. **Catálogo técnico aço M333**, disponível em: [http://www.bucanada.ca/m333\\_isoplast.htm](http://www.bucanada.ca/m333_isoplast.htm). Acesso em Outubro de 2009.

FASEL, I.R.; VARTLETT, M.S.; MOVELLAN, J.R. A comparison of Gabor filter methods for automatic detection of facial landmarks. In: **Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition**. v. 21, p. 242–246, 2002.

FENG, H-P; ZHAO, J. Application Research of Computer Vision in the Auto-Calibration of Dial Gauges. **Computer Science and Software Engineering**. p. 845–848, 2008.

FERNANDES, A.M.R, **Inteligência Artificial**. v. 1, Visual Books, 2003. 160 p.

FOGEL, I.; SAGI, D. Gabor filters as texture discriminator. **Biological Cybernetics**. v. 61, p. 103-113, 1989.

FRACASSO, P. T.; SILVA, T. J.; SELVATICI A. H. P. Bancada inteligente para calibração de hidrômetros. In: **14o Congresso Internacional de Automação, Sistemas e Instrumentação**. São Paulo, p. 1-10, 2010.

GONZALEZ, R. C.; WOODS, R. E. **Digital Image Processing**. Massachusetts: Addison-Wesley, 2002. 793 p.

HAYKIN, S. **Redes Neurais – Princípios e práticas**. v. 2, Porto Alegre: Bookman, 2001. 900 p.

HE, W; ZHAOSHENG, T.; SHENGJIE, Y; SHURONG, L. Intelligent reading method for analog meter based on computer vision. **Chinese Journal of Scientific Instrument**. v. 28, p. 1234–1239, 2007.

HEMMING, B.; LEHTO, H. Calibration of dial indicators using machine vision. **Measurement Science and Technology**. v. 13, p. 45–49, 2002.

HEMMING, B.; FAGERLUND, A.; LASSILA, A. High-accuracy automatic machine vision based calibration of micrometers. **Measurement Science and Technology**. v. 18, p. 1655–1660, 2007.

HII, A.J.; HANN, C.E.; CHASE, J.G.; VAN HOUTEN, E.E. Fast normalized cross correlation for motion tracking using basis functions. **Computer Methods and Programs in Biomedicine**. v. 82, p. 144–156, 2006.

HIPPENSTIEL, F. **Haddbook of Plastic Mould Steels**. Edelstahlwerke Buderus AG, 2001. 294 p.

HUTCHINSON, S.; HAGER, G. D.; CORKE, P. I. A tutorial on visual servo control. **IEEE Transactions on Robotics and Automation**, v. 13, n. 5, p. 651-670, 1996.

INMETRO. **Regulamentação metrológica: resolução CONMETRO nº 11/88**. 3. ed. Rio de Janeiro: SENAI, 2007.

INTEL, **OpenCV - Open Source Computer Vision Library**. Disponível em: <http://sourceforge.net/projects/opencvlibrary>. Acesso em janeiro de 2012.

JAIN, A.K.; FARROKHNI, F. Unsupervised texture segmentation using Gabor filters. **Pattern Recognition**. v. 24, p. 1167-1186, 1991.

JOLIFFE, I.T. **Principal Component Analysis**. Springer, 2002. 487 p.

KIM, H. Y. **ProEikon - Rotinas e programas em C++ para processamento de imagens e visão computacional**, São Paulo, 2012. Disponível em: <http://www.lps.usp.br/~hae/software>. Acesso em março 2012.

KIM, H.Y., ARAÚJO, S.A. Grayscale Template-Matching Invariant to Rotation, Scale, Translation, Brightness and Contrast. **IEEE Pacific-Rim Symposium on Image and Video Technology, Lecture Notes in Computer Science**. v. 4872 p. 100–113, 2007.

LEE, Chih-Jen; WANG, Sheng-De; WU, Kuo-Ping; Fingerprint recognition using principal Gabor basis function. In: **International Symposium on Intelligent Multimedia, Video and Speech Processing**. p. 393–396. Taiwan, 2001.

LONCARIC, S. A. Survey of Shape Analysis Techniques. **Pattern Recognition**, v. 31, n. 8, p. 983-1001, 1998.

LONG, F.; ZHANG, H.; FENG, D. Fundamentals of Content-based Image Retrieval. In: FENG, D.; SIU, W. C.; ZHANG, H. **Multimedia Information Retrieval and Management: Technological Fundamentals and Applications (Signals and Communication Technology)**, Berlin, 2005. p. 1-26.

MESQUITA, R. A.; SCHNEIDER, R.S.E. Tool steel quality and surface finishing of plastic molds. **Exacta** v. 8, p. 307-318, 2010.

MESQUITA, R.A. Moldes ganham nova geração de aço. In: **X Seminário de Atualidades Tecnológicas, Elastômeros, Plásticos, Adesivos e Moldes**, p. 50, 2005.

PAIVA JR., R.D.; MARAMATSU, M. Desenvolvimento de um Rugosímetro a Laser. **Revista de Física Aplicada e Instrumentação** . v. 17, p. 60-70, 2004.

PASTERNAK, R.M.; QIAN, Z.; ZHENG, J.-Y.; METAXAS, D.N.; BOUSTANY, N.N. Highly sensitive size discrimination of sub-micron objects using optical Fourier processing based on two-dimensional Gabor filters. **Optics express**. v. 17, p. 12001-12012, 2009.

PEDRINI, H; SCHWARTZ, W.R. **Análise de Imagens Digitais**. v. 1, São Paulo: Thomson, 2007. 528 p.

PIRES, G.P.; TAGIAFERRI, A.A.; THAUMATURGO, C. Interferometria Óptica Para Ensaios Não Destrutivos. **Revista Matéria**. v. 7, p. 1–17, 2002.

RUSSEL, S., NORVIG, S. **Artificial Intelligence: A Modern Approach**. New Jersey: Prentice Hall, 1995. 932 p.

SANKAR, M. R.; JAIN, V.K.; RAMKUMAR, J. Rotational abrasive flow finishing (R-AFF) process and its effects on finished surface topography. **International Journal of Machine Tools & Manufacture**. v. 50, p. 637–650, 2010.

SHIMIZU, T.; FUJII, T. Mirror surface finishing properties of plastics mold steels. Daido Steel Co. Ltd. **Journal: Electric Furnace Steel**. v.74, v.2, p. 125-130, 2003.

SILVA, L. R. O.; ALVEZ, M. L. A calibração periódica de instrumentos de medição e padrões e suas relações com custo benefício. In: ENQUALAB, 2004, São Paulo.

SONG, J.; CAI, M.; LYU, M. R.; CAI, S. A New Approach for Line Recognition in Large-size Images Using Hough Transform. **Pattern Recognition**, v. 1, p. 33–36, 2002.

SUM, J.J., TAYLOR, E.J., SRINIVASAN, R. MREF-ECM process for hard passive materials surface finishing. **Journal of Materials Processing Technology**. v. 108, p. 356–368, 2001.

TSAI, D. M.; TSAI, Y. H. Rotation-invariant pattern matching with color ring-projection. **Pattern Recognition**, v. 35, n. 1, p. 131-141, 2000.

VÁZQUEZ-FERNÁNDEZ, E.; DACAL-NIETO, A.; GONZÁLES-JORGE, H.; MARTÍN, F.; FORMELLA, A.; ALVAREZ-VALADO, V. A machine vision system for the calibration of digital thermometers. **Measurement Science and Technology**. v. 20, p. 1–7, 2009.

YAN, B-H; TZENG, H-J; HUANG, F. Y.; LIN, Y-C; CHOW, H-M. Finishing effects of spiral polishing method on micro lapping surface. **International Journal of Machine Tools & Manufacture**. v. 47, p. 920–926, 2007.

WEKA, **Weka 3: Data Mining Software in Java**. Waikato, 2012. Disponível em <http://www.cs.waikato.ac.nz/ml/weka/>. Acesso em março de 2012.

ZANATTA, A. M.; GOMES, J. O.; MESQUITA, R. A. Influência do enxofre na usinabilidade e polibilidade de aços ferramenta para moldes. In: **XII COLÓQUIO DE USINAGEM**, 2008, Uberlândia.

**APÊNDICE A – PATENTE DEPOSITADA PELO AUTOR NO PERÍODO DO  
MESTRADO**

Processo nº:	PI 018110040724 - nº. Provisório
Privilegio:	Patente de Invenção
Título:	Processo e equipamento para análise de superfície de moldes para injeção por meio de reflexão laser.
Data:	20/10/2011
Situação:	Em sigilo e aguarda Publicação

**ANEXO A – ARTIGO PUBLICADO NO CONGRESSO: XLIII SIMPÓSIO  
BRASILEIRO DE PESQUISA OPERACIONAL, 2011, Ubatuba, São Paulo**

## **A template matching Based Algorithm applied to Automatic Calibration of Digital Measuring Instruments without built-in communication interface**

**Sidnei Alves de Araújo**

Programa de Pós-Graduação em Engenharia de Produção – Universidade Nove de Julho  
Av. Francisco Matarazzo, 612, Água Branca, São Paulo – SP, Brasil  
saraujo@uninove.br

**Peterson Adriano Belan**

Programa de Pós-Graduação em Engenharia de Produção – Universidade Nove de Julho  
Av. Francisco Matarazzo, 612, Água Branca, São Paulo – SP, Brasil  
peterson@gruponef.com.br

**André Felipe Henrique Librantz**

Programa de Pós-Graduação em Engenharia de Produção – Universidade Nove de Julho  
Av. Francisco Matarazzo, 612, Água Branca, São Paulo – SP, Brasil  
librantz@uninove.br

### **ABSTRACT**

A common problem found in the calibration laboratories is the reliability of the results obtained in the calibration of instruments especially when they do not have a built-in communication interface. In this case, the time consuming is increased significantly and the calibration may be subject to human error since this task involves several manual data readings and transcriptions. In this context, many authors have proposed computer systems with the purpose of automating calibration processes. This paper describes an algorithm using template matching with normalized cross correlation for reading the display of digital measuring instruments without built-in communication interface and storing the read ASCII value in a database used in the certification process. The experimental results showed that the proposed algorithm presents a high performance and can be used in real time computer vision systems applied to calibration processes.

**KEYWORDS.** Calibration. Computer Vision. Template Matching.

## 1. Introduction

A common problem found in the calibration laboratories nowadays is the productivity of their technicians and the reliability of the results obtained in the calibration process and the transcript of the calibration's certificate. This is mainly caused by the fact that, in most cases, the technician who performed the calibration service does not issue the certificate, but only signs it after the filling done by another technician. In some cases the result in the certificate can be different that one obtained during the calibration process, frequently caused by human errors in transcription data.

The calibration processes accuracy may be significantly affected when instruments without built-in communication interface is involved because the several manual readings and transcriptions of the data are more subject to human errors. Moreover, this task is time consuming and stressful.

Thus, automated systems play a very important role nowadays in the calibration of measuring instruments because they provide greater accuracy, repeatability and cost savings, beyond the reduction of monotonous and complex tasks (Alegria & Serra, 2000).

In this context, many authors have proposed automatic calibration systems using computer vision techniques (Alegria & Serra, 2000; Andria et al., 2009; Fernández et al., 2009; Fracasso, Silva & Selvatici, 2010; He et al., 2007).

Computer vision can be defined as a sub-area of image processing that studies the development of methods and techniques that enable a computer system to recognize objects in images imitating some capabilities of the human visual system, as the ability to describe a scene contained in a digital image (Gonzalez, 2002).

An efficient computer vision system must be able to extract a set of attributes that accurately describe a scene and is small enough to reduce the processing time to be used in practical applications such as robot vision systems, autonomous vehicles, surveillance systems, automatic license plate recognition, industrial inspection and biometrics patterns recognition, among others (Araújo, 2009).

One of the most known technique in computer vision systems is the template matching and is used to find small parts of an image, called template, in another image that has been analyzed (Araújo, 2009; Gonzalez, 2002).

In this paper is presented an approach, based on template matching with normalized cross correlation, to automate the process of calibration of digital measuring instruments without a built-in communication interface. The proposed algorithm uses computer vision techniques and allows the calibration values to be automatically input in a database, reducing the possibility of human errors in reading and transcription data tasks.

The remainder of the paper is organized as follows. Section 2 introduces the template matching technique using normalized Cross Correlation. Section 3 describes in detail the proposed algorithm. Section 4 presents some experimental results on real images and finally, the Section 5 concludes the paper and shows our plans for the future work.

## 2. Template Matching with Normalized Cross Correlation

Template matching is a technique widely used for finding patterns in digital images. Let  $A$  be the image to be analyzed and  $T$  the query template. The goal of a template matching algorithm is to find all occurrences of  $T$  in  $A$ , as illustrated in the Figure 1.

Template matching can be done in two ways: feature-based or template-based. The first way uses the features of the  $T$  and  $A$  to find the best matching locations of  $T$  in  $A$ . The second, also called global approach, uses the entire template to determine the matching positions. In both cases, some similarity measure needs to be used. Template-based approaches can be conducted using, for example, sum of square differences (SSD), sum of absolute differences (SAD) or normalized cross correlation (NCC) measures (Hii et al., 2006).

In most cases the instances of  $T$  in  $A$  may appear rotated, scaled, shifted and with diverse brightness and contrast (Araújo & Kim, 2011). Due this fact template matching technique

is well known to be an expensive operation and the execution time is proportional to the size of the images  $A$  and  $T$ .

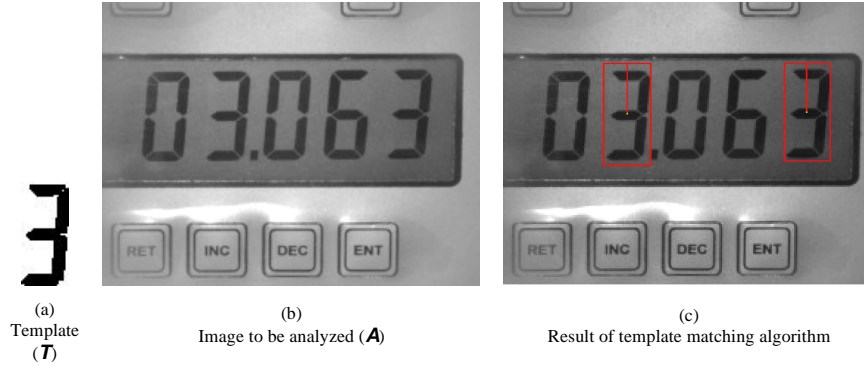


Figure 1. Example of template matching.

The simplest implementation of template matching algorithm, using global approach, invariant to rotation and scale is the “brute force” solution. It performs a series of template matchings between the image to be analyzed  $A$  and the instances of the query template image  $T$  rotated by every angles and scaled by every scale factors, considering some specified range of angles and scale factors (Kim & Araújo, 2007).

The brute force algorithm yields the most precise solution to this problem. Nevertheless, it is very time consuming and thus is not feasible to be used in practical applications. However, this technique can be very efficient if does not have the necessity of rotation and scale invariance (Araujo & Kim, 2011).

Normalized cross correlation (NCC) is widely used as a similarity measure in template matching and is the most robust correlation measure for determining similarity between points in two or more images (Hii et al., 2006).

Considering an image  $A$  of size  $M \times N$  pixels and a template  $T$  of size  $I \times J$  pixels, then the cross correlation is defined as:

$$C(m, n) = \sum_{i=-a}^a \sum_{j=-b}^b T(i, j) A(m+i, n+j) \quad (1)$$

where:  $a = (I - 1)/2$  e  $b = (J - 1)/2$ ,  $m=0, \dots, M-1$ ,  $n=0, \dots, N-1$ ,  $C$  is the result matrix in which the peaks indicating the occurrences of  $T$  in  $A$  (matching positions).

In many applications the brightness of the images can vary due to environment lighting conditions. In these cases, the images can be first normalized. Thus,  $NCC(A, T)$  can be defined as:

$$NCC(A, T) = C(m, n) = \frac{\sum_{i=-a}^a \sum_{j=-b}^b [T(i, j) - \bar{T}][A(m+i, n+j) - \bar{A}]}{\sqrt{\sum_{i=-a}^a \sum_{j=-b}^b [T(i, j) - \bar{T}]^2 \sum_{i=-a}^a \sum_{j=-b}^b [A(m+i, n+j) - \bar{A}]^2}} \quad (2)$$

The computation of  $NCC(A, T)$ , defined in Equation 2, generates a matrix  $C$  with the same size of  $A$ , in which the values range from -1 to +1. Thus, depending on the application, we can use either the absolute value  $|C(m, n)|$  to allow matching negative instances of the  $T$  in  $A$  or

the value of  $C(m, n)$  with signal for not allowing the matching of negative instances of  $T$ .

The Figure 2 depicts the result of NCC between  $T$  and  $A$ . The highest peaks indicate the occurrences of instances of  $T$  in  $A$ .

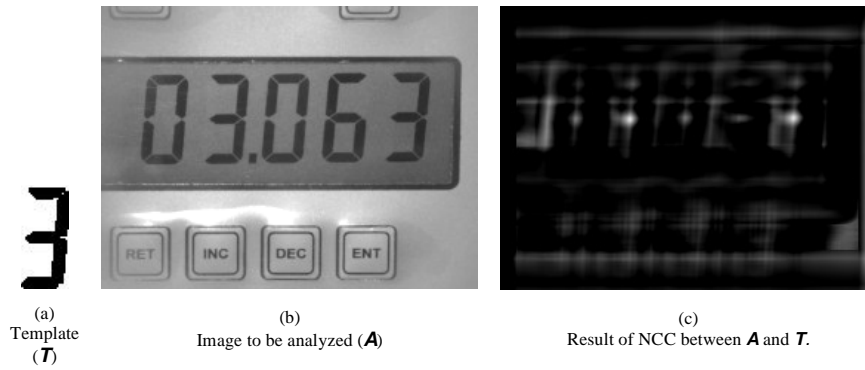


Figure 2. NCC between two images.

In the practice, template matching with NCC is performed by moving the center (or the origin) of the template  $T$  over each  $(m, n)$  point in the  $A$  and calculate the sum of products between the coefficients in  $A(m, n)$  and  $T(i, j)$  over the whole area spanned by  $T$ .

The NCC computation is high costly in time. However, the most available library implementation makes is optimized and makes use of Fast Fourier Transform (FFT), where the correlation sums are reduced to matrix products, accelerating the processing time. It is the case of the OpenCV library (Intel, 2007) used in our implementations.

### 3. The proposed algorithm

The algorithm proposed in this paper was implemented in C/C++ language using OpenCV and Proeikon librarys (Intel, 2007; Kim, 2010). It is divided in two main steps: i) Regions of Interest (ROIs) determination and ii) Recognition of the digits. After two steps, the ASCII data representing the digits are storage in a database.

#### 3.1 ROIs determination

In the first step of the algorithm a color image of size  $851 \times 638$  is acquired by a conventional low cost webcam, converted at 256 gray levels and reduced to 30% of its original size. Examples of input image are showed in the figures 1a and 2a.

Since the template matching technique may be an expensive operation concerning the time processing, the reduction of search area is important to accelerate the algorithm.

To detect the ROIs automatically, firstly NCC between input image and the template of digit 8 ( $T_8$ ), denoted by  $NCC(A, T_8)$ , is computed generating the image  $C$ , illustrated in the Figure 3a, in which the highest peaks give the coordinates of the line crossing the vertical center of the digits (Figure 3b).

In addition, by determining a threshold value  $t$ , it is possible to select the five highest peaks over the line (Figure 3c) that indicate the horizontal center of each digit.

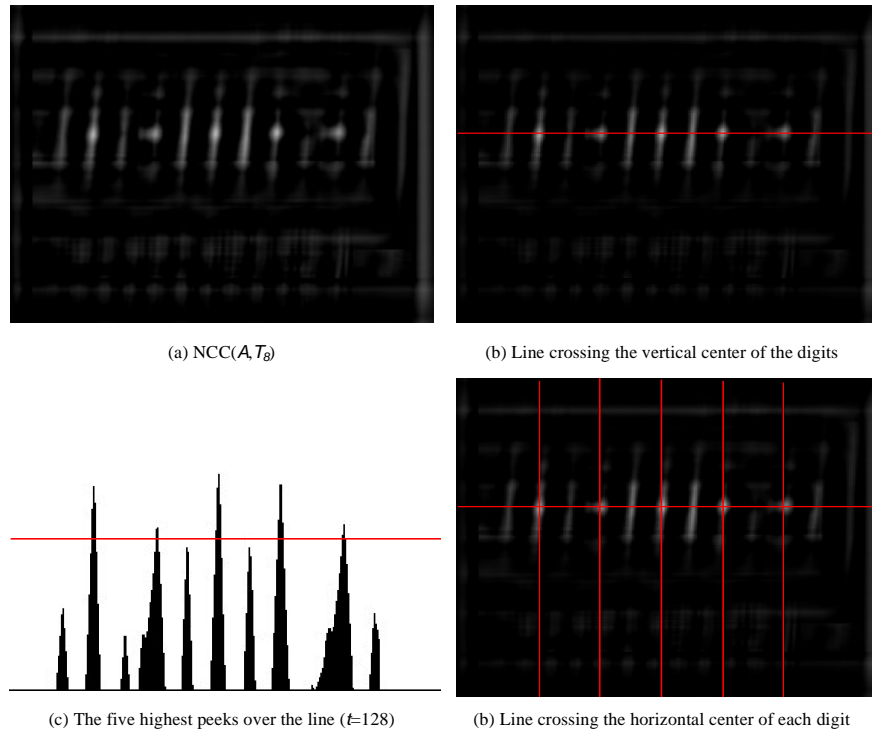


Figure 3. NCC between two images.

Using the coordinates that denote the center of the digits, the initial and final position of each subregion  $S_0, \dots, S_4$  is finally calculated, as shown in the Figure 4.



Figure 4. Detected subregions.

However, it is valid to remember that the decimal point need not be recognized because it is always fixed after the second digit.

### 3.2 Recognition of the digits

In this step, the digits of the display are recognized and stored in a database. For this purpose, first the NCC is calculated between each  $S_i$  (Figure 4) and each one of the templates  $T_i$  (Figure 5). From this operation is obtained the matrix  $P$  with 5 rows (subregions of  $A$ ) and 10 columns (templates) and is defined as:

$$P[l, c] = NCC(S_l, T_c) \quad (3)$$

where  $l=0, \dots, 4$  and  $c=0, \dots, 9$ .

The index  $c$  of highest value of each line in the matrix  $P$  indicating the value displayed in the analyzed instrument on the position  $l$ . Thus, the output data consists of an integer vector  $D$  containing 5 elements. This vector is obtained as follows:

$$D[l] = \underset{c=0}{\operatorname{ARGMAX}}^9 [P[l, c]] \quad (4)$$

where  $c$  is the argument that maximizes  $P[l, c]$ .

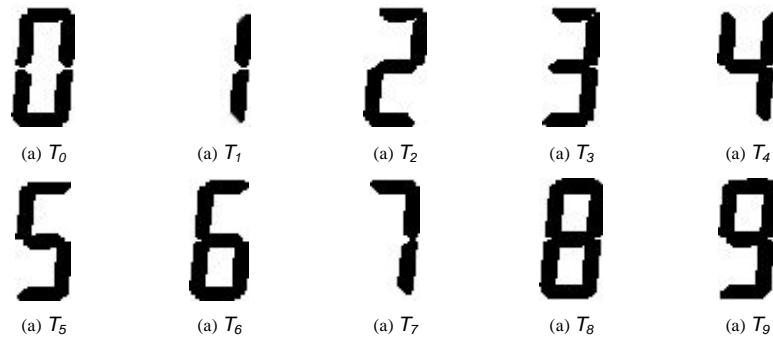


Figure 5. Query templates used in recognition step.

The Figure 6 illustrates the matrix  $P$  calculated to recognize the display digits depicted in the figures 1, 2 and 4.

	0	1	2	3	4	5	6	7	8	9
1	<b>215</b>	149	106	122	112	93	148	184	176	127
2	133	144	137	<b>219</b>	103	124	101	163	139	163
3	<b>223</b>	150	104	130	100	112	162	178	181	136
4	146	152	80	126	93	188	<b>227</b>	168	171	125
5	114	150	127	<b>225</b>	110	129	97	162	138	164

Figure 6. Matrix  $P$  calculated in the recognizing of the display containing the digits “0”, “3”, “0”, “6” and “3”. The correlation values ranging from -1 to 1 were converted to the interval [0, 255].

Finally, the ASCII values from vector  $D$  representing the displayed digits, a display image's copy and other data about the calibration process are storage in a database to be used in the instrument certificate or to identify some mistake occurred in the digits recognition.

#### 4. Experimental results

In order to evaluate the robustness of the proposed algorithm we have conducted experiments using a set of 50 images affected by illumination changes (Figure 7a and 7b); noise (Figure 7c) and JPEG compression (Figure 7d). In these experiments all 250 matchings were perfect, without any false positive or false negative, even in the presence of some faintly visible digits (e.g. Figure 7b).

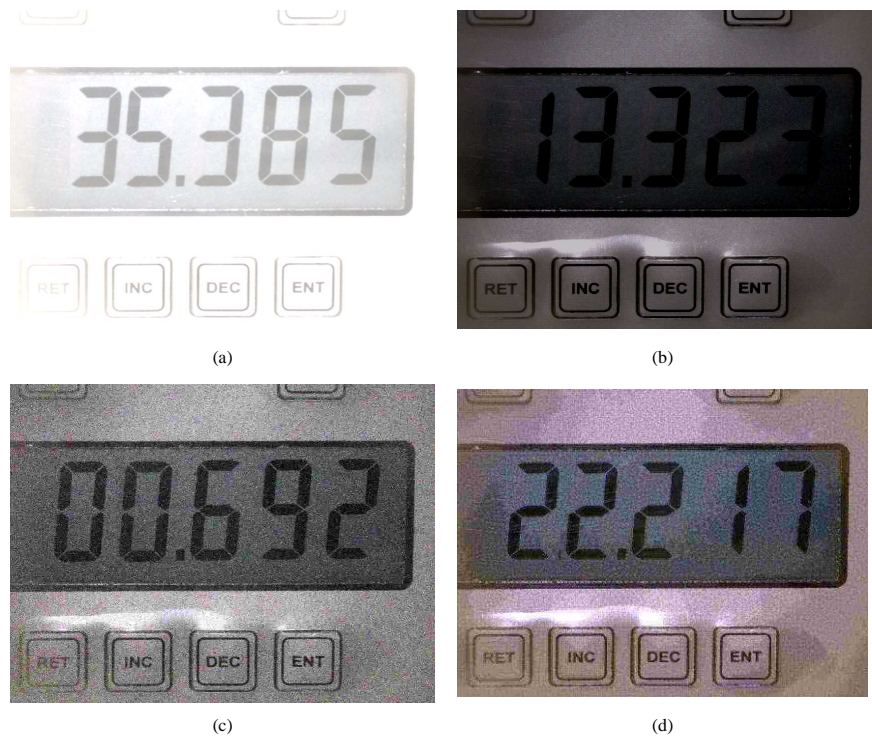


Figure 7. NCC between two images.

The distribution of the digits in the image set is showed in the Table 1. As can be seen all 10 digits appear at least 7.6% in the images.

Tabela 1. Distribution of the digits in the set of images.

Digit	0	1	2	3	4	5	6	7	8	9
%	14,4	13,2	10	10	7,6	7,6	9,6	9,2	9,6	8,8

The robustness of the proposed algorithm can be addressed to the fact that, differently from most used techniques like (Alegria & Serra, 2000), it does not discard the rich grayscale information through operations such as detection of edges or segmentation/binarization of the images. The main problem is that these image-simplifying operations throw away the rich grayscale information, are sensitive to noise and JPEG, decreasing the robustness of the matching (Kim & Araújo, 2007).

Concerning the time of processing, the algorithm takes, in average, 0.3 sec using a Pentium 4 2.8GHz to read the numbers appearing in the display image. Nevertheless, this time could be further reduced by performing the template matching operation, in the second step, only in a few pixels around the peaks representing the center of the digits instead all pixels inside each subregions Si. This reduction in processing time can allow the use of the algorithm in real-time computer vision systems applied to calibration processes.

## 5. Conclusions

This paper describes an algorithm using template matching technique with normalized cross correlation for reading the display of digital measuring instruments without built-in communication interface and storing the read ASCII values in a database used in the certification process.

In the experiments using a set of 50 images affected by illumination changes, noise, and JPEG compression all 250 digits were correctly recognized, showing the robustness of proposed algorithm. In addition, its performance could be further improved by reducing the ROIs. Besides, the algorithm could be used in several kinds of digital measuring instruments using a set of templates previously stored in a database.

In the future works we intend to improve the proposed algorithm making it feasible to be applied in a real-time automated calibration system and also studying and applying computer vision methods in the automation of other types of measuring equipment such as analog instruments.

## Acknowledgment

The authors would like to thank Uninove by financial support and CAPES for the scholarship granted to one of the authors.

## Referências

- Alegria F. C. & Serra, A. C.** (2000), *Automatic Calibration of Analog and Digital Measuring Instruments Using Computer Vision*, IEEE Transactions on Instrumentation and Measurement, 49 (1), 94-99.
- Andria, G. et al.** (2009), *Automatic Calibration System for Digital Instruments Without Built-In Communication Interface*, Anais do XIX IMEKO World Congress, 857-860, Lisbon, September 6-11.
- Araújo, S.A.** (2009), *Casamento de padrões em imagens digitais livre de segmentação e invariante sob transformações de similaridade* – Tese de doutorado, USP, São Paulo.
- Araújo, S.A. & Kim, H. Y.** (2011), *Ciratefi: An RST-invariant template matching with extension to color images*. Integrated Computer-Aided Engineering, 18, 75-90.
- Borges, R.O.; Silva, R.A.A. & Castro, S.S.** (2007), *Utilização da classificação por distância euclidiana no mapeamento dos focos de arenização no setor sul da alta bacia do Rio Araguaia*, Anais do XIII Simpósio Brasileiro de Sensoriamento Remoto, Florianópolis, Brasil, 21-26 abril 2007, INPE, p. 3777-3784.
- Fernández, E.V. et al.** (2009), *A machine vision system for the calibration of digital thermometers*, Measurement Science and Technology, 20 (6).
- Fracasso, P. T.; Silva, T. J. & Selvatici A. H. P.** (2010), *Bancada inteligente para calibração de hidrômetros*, Anais do XIV Congresso Internacional de Automação, Sistemas e Instrumentação, São Paulo, 1-10.
- Gonzalez, R. C. & Woods, R. E.** (2002), *Digital Image Processing*. Massachusetts: Addison-Wesley.
- He, W. et al.** (2007), *Intelligent reading method for analog meter based on computer vision*, Chinese Journal of Scientific Instrument, 28 (7), 1234-1239.
- Hii, A.J.H. et al.** (2006), *Fast normalized cross correlation for motion tracking using basis functions*, Computer Methods and Programs in Biomedicine, 82 (2), 144-156.

- Intel.** (2007), *OpenCV - Open Source Computer Vision Library*. Disponível em: <<http://www.intel.com/technology/computing/opencv/>>. Acesso em: 09 out. 2007.
- Kim, H.Y.** (2010) *ProEikon - Rotinas e programas em C++ para processamento de imagens e visão computacional*, São Paulo, 2006. Disponível em: <<http://www.lps.usp.br/~hae/software>>. Acesso em fev. 2010.
- Kim, H. Y. & Araújo, S. A.** (2007), *Grayscale Template-Matching Invariant to Rotation, Scale, Translation, Brightness and Contrast*. Lecture Notes in Computer Science, 4872, 100-113.

**ANEXO B – ARTIGO ACEITO PARA PUBLICAÇÃO NA REVISTA EXACTA, 2012.**

# Técnicas de visão computacional aplicadas no processo de calibração de instrumentos de medição com *display* numérico digital sem interface de comunicação de dados

*Computer vision techniques applied to the process of calibrating measurement instruments having numeric displays without built-in data communication interfaces*

Peterson Adriano Belan

Bacharel em Ciência da Computação e Mestrando em Engenharia de Produção – Universidade Nove de Julho (UNINOVE).  
São Paulo, SP [Brasil]  
peterson@gruponef.com.br

Sidnei Alves de Araújo

Doutor em Engenharia Elétrica pela Escola Politécnica – Universidade de São Paulo – USP, Professor e Pesquisador do Programa de Mestrado em Engenharia de Produção e do curso de Bacharelado em Ciência da Computação – Universidade Nove de Julho – UNINOVE.  
São Paulo, SP [Brasil]  
saraujo@uninove.br

André Felipe Henriques Librantz

Doutor em ciência de materiais – Universidade de São Paulo – USP, Diretor do Programa de Mestrado em Engenharia de Produção – Universidade Nove de Julho – UNINOVE.  
São Paulo, SP [Brasil]  
librantz@uninove.br

## Resumo

O processo de calibração de instrumentos que não possuem uma interface de comunicação de dados consome muito tempo e pode estar sujeito a erros humanos, pois envolve inúmeras leituras e anotações feitas pelo técnico que realiza a tarefa. Nesse contexto, os sistemas automatizados podem proporcionar maior precisão, repetibilidade, redução de custos e padronização de resultados. Neste trabalho, exploram-se técnicas de visão computacional para automatização do processo de calibração de instrumentos de medição com *display* numérico digital que não possuam uma interface de comunicação de dados. Utilizando-se uma câmera USB, a imagem do visor do equipamento é adquirida e, após sua análise, os dígitos são reconhecidos e os valores são armazenados para serem utilizados na emissão dos certificados de calibração. Para essa tarefa, foram descritas e avaliadas três diferentes abordagens e os resultados experimentais mostraram alta confiabilidade em duas delas, sugerindo que ambas podem ser aplicadas na automatização do processo de calibração.

**Palavras-chave:** Calibração. Instrumentos de medição digital. Visão computacional.

## Abstract

The process of calibrating instruments that do not have built-in data communication interfaces is very time-consuming and may be subject to human error since it involves many data readings and annotations by the technician who performs the task. In this context, automated systems play an important role since they can provide higher accuracy, repeatability, cost reduction and standardization of results. This paper explores the use of computer vision techniques in automating the process of calibrating measurement instruments with numeric digital displays that do not have data communication interfaces. Using a USB camera, the image of the instrument display is acquired; and, after its analysis, the digits are recognized and the corresponding values stored in a file to be used in issuing calibration certificates. For this task, three different approaches were described and evaluated. Experimental results showed a high reliability in two of them, suggesting that both can be applied in the automation of the calibration process.

**Key words:** Calibration. Computer vision. Digital measurement instruments.



## 1 Introdução

Um problema encontrado nos laboratórios de calibração nos dias de hoje é a produtividade de seus técnicos e a fidelidade dos resultados obtidos na calibração e na transcrição de seu certificado. Isso ocorre principalmente pelo fato de que, na maioria dos casos, o técnico que executou o serviço de calibração não emite o certificado, mas apenas o assina depois que um digitador preenche esse documento. Em alguns casos, o resultado expresso pode não ser o real obtido durante a calibração; situação ocasionada, sobretudo, por falhas humanas, tais como anotação incorreta do valor exibido no *display* do instrumento no momento da calibração ou cópia errada – emitida pelo digitador da planilha –, preenchida pelo técnico.

Dessa forma, os sistemas automatizados desempenham, nos dias atuais, um papel muito importante na calibração de instrumentos de medição. Eles proporcionam maior precisão, repetibilidade e redução de custos, além da diminuir os trabalhos monótonos e de tarefas complexas (ALEGRIA; SERRA, 2000).

O processo de calibração de instrumentos que não possuem uma interface de comunicação de dados consome muito tempo e pode estar sujeito a erros humanos, uma vez que envolve inúmeras leituras e anotações do técnico que realiza a tarefa. Nesse contexto, diversos autores têm proposto sistemas de calibração automática empregando técnicas de visão computacional (ALEGRIA; SERRA, 2000, ANDRIA et al., 2009, FERNÁNDEZ et al., 2009, HE et al., 2007, FRACASSO; SILVA; SELVATICI, 2010).

Neste trabalho, explora-se o emprego de técnicas de visão computacional para automatizar o processo de calibração de instrumentos de medição com indicador digital numérico que não possuam uma interface de comunicação de dados.

Especificamente, são apresentadas e avaliadas três abordagens para reconhecimento automático dos dígitos que aparecem no *display* do instrumento que está sendo analisado, a saber: reconhecimento com base na matriz contendo os valores dos *pixels*; reconhecimento usando histogramas de projeção e reconhecimento utilizando segmentos de retas adjacentes. Tais técnicas foram escolhidas por serem amplamente exploradas em tarefas semelhantes às descritas neste trabalho. A ideia é que os valores oriundos da leitura automática sejam inseridos em um arquivo usado na emissão dos certificados de calibração, diminuindo a possibilidade de erros na digitação.

Assim, pretende-se melhorar o desempenho do processo de calibração e emissão dos certificados, tendo em vista a eliminação da etapa de verificação dos dados digitados.

## 2 Materiais e métodos

Para realização dos experimentos, utilizou-se uma base constituída por 43 imagens adquiridas em um laboratório de calibração, contendo um total de 215 dígitos, distribuídos, conforme mostra a Tabela 1.

**Tabela 1: Distribuição dos dígitos utilizados para os experimentos**

Dígito	0	1	2	3	4	5	6	7	8	9	Total
Distribuição dos dígitos	31	29	20	18	18	17	21	19	23	19	215

A implementação dos algoritmos foi realizada em linguagem C++; utilizando a biblioteca de rotinas de processamento de imagens Proeikon (KIM, 2010), desenvolvida para trabalhar na plataforma Windows.



*pixels* que compõem os objetos e os que compõem o fundo da imagem em duas classes distintas. Normalmente esse processo consiste na conversão de imagens em escala de cinzas para binárias (Equação 1), em que valor 1 representa o branco; e 0, o preto, podendo ser 1 para representação do fundo; e 0, para os objetos ou vice-versa.

$$g(x, y) = \begin{cases} 1, & \text{se } f(x, y) > L \\ 0, & \text{se } f(x, y) \leq L \end{cases} \quad (1)$$

Em que  $g$  é a imagem binária de saída e  $f$  é a imagem de entrada em níveis de cinzas.

A grande dificuldade em processos de limiarização é estabelecer automaticamente o limiar que classifique os *pixels* de maneira adequada para uma determinada aplicação (GONZALES; WOODS, 2000). Uma técnica amplamente utilizada para limiarização global é o algoritmo de Otsu (OTSU, 1979), o qual permite estabelecer, de forma automática, um limiar  $L$  com base no histograma da imagem.

### 2.2.2 Componentes conexos

A conectividade entre *pixels* é um conceito importante usado no estabelecimento de bordas de objetos e componentes de regiões em uma imagem. Para determinar se dois *pixels* estão conectados, é preciso definir a noção de vizinhança e se seus níveis de cinza obedecem algum critério de similaridade. Considerando imagens binárias, que contêm apenas duas classes (0 e 1), dois *pixels* vizinhos estão conectados se eles pertencerem à mesma classe, obedecendo assim ao critério de similaridade (GONZALES; WOODS, 2000; PEDRINI; SCHWARTZ, 2007).

Existem duas formas comuns de conectividade para imagens bidimensionais: conectividade-de-4 e conectividade-de-8, conforme modelos apresentados na Figura 2. Dois *pixels* estão conectados de 4 se obedecem ao critério de similaridade

e se são vizinhos horizontais ou verticais, e estão conectados de 8 se obedecem ao critério de similaridade e são vizinhos horizontais, verticais ou diagonais. Os vizinhos horizontais de um *pixel*  $p$  de coordenada  $(x, y)$  são os *pixels* de coordenadas  $(x+1, y)$  e  $(x-1, y)$ , os verticais são de coordenadas  $(x, y+1)$  e  $(x, y-1)$ ; e os diagonais são de coordenadas  $(x+1, y+1)$ ,  $(x+1, y-1)$ ,  $(x-1, y+1)$  e  $(x-1, y-1)$  (GONZALES; WOODS, 2000).

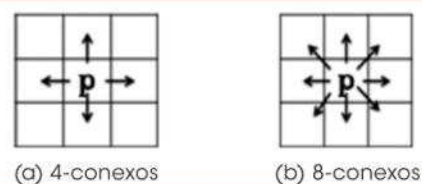


Figura 2: Exemplo de conectividade e vizinhança de *pixels*

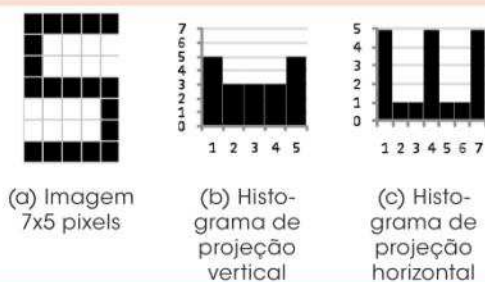
Componente conexo de uma imagem é o conjunto de *pixels* que estão conectados de 4 ou conectados de 8 entre si (GONZALES; WOODS, 2000).

### 2.2.3 Histogramas de projeção

O uso de projeções para extração de características é bastante comum, principalmente em aplicações que envolvem reconhecimento de caracteres. O histograma de projeção é um método que utiliza a projeção da imagem em uma determinada direção. As mais comuns são as direções vertical e horizontal. Para cada direção é associado um vetor no qual se armazena o número de *pixels* com uma determinada tonalidade de cinza, geralmente aqueles compondo o objeto que se deseja representar ou descrever (PEDRINI; SCHWARTZ, 2007).

Um histograma de projeção vertical, por exemplo, representa as quantidades de *pixels* pertencentes ao objeto em cada coluna, enquanto um histograma de projeção horizontal representa as quantidades de *pixels* pertencentes ao objeto em cada linha.

As Figuras 3b e 3c ilustram, respectivamente, o histograma de projeção vertical e o de projeção horizontal da imagem mostrada na Figura 3a.



**Figura 3: Exemplos de histogramas de projeção vertical e horizontal**

Também podem ser usados histogramas de projeção circular e radial para representação de objetos. No caso da projeção circular, uma vantagem é a invariância à rotação do objeto (KIM; ARAÚJO, 2007).

### 3 Leitura automática do display do instrumento de medição

O processo de leitura automática da informação mostrada no *display* envolve as etapas de determinação da região de interesse (RI), pré-processamento da RI e reconhecimento dos dígitos, as quais são descritas ao longo desta seção. Para o reconhecimento dos dígitos são descritas três abordagens na subseções 3.3 a 3.5.

#### 3.1 Determinação da RI

A região de interesse é a área da imagem em que será realizado o processamento. Para tanto, no primeiro passo do algoritmo, uma imagem colorida (padrão RGB) 640x480 *pixels* é adquirida via *webcam* (Figura 4a). Na sequência, a imagem adquirida é limiarizada por meio do algoritmo de Otsu, gerando uma imagem binária (Figura 4b).

Com base na imagem binária, calcula-se o histograma de projeção horizontal (Figura 4c) para, posteriormente, segmentar a região da imagem que contém os dígitos (Figura 4d). As linhas contendo os maiores picos indicam o início e o fim da região de interesse.

Cabe ressaltar que o histograma de projeção horizontal mostrado na Figura 4c foi propositalmente rotacionado, visando um melhor entendimento da segmentação. Nas etapas seguintes, procede-se o pré-processamento e reconhecimento dos dígitos. Para esta última, três abordagens são apresentadas.

#### 3.2 Pré-processamento da RI

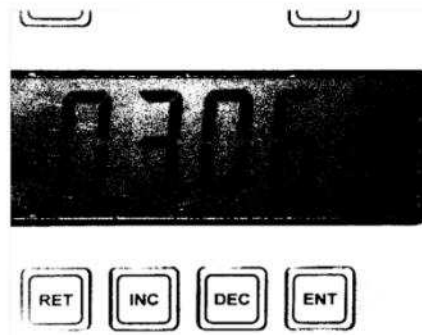
Nessa etapa, o primeiro passo é efetuar a limiarização da RI empregando o algoritmo de Otsu (OTSU, 1979; GONZALES; WOODS, 2000), resultando na imagem mostrada na Figura 5a. O próximo passo é efetuar a remoção de componentes conexos menores que 100 *pixels* (valor obtido empiricamente para ser utilizado especificamente no problema abordado), uma vez que tais componentes podem comprometer o processo de reconhecimento dos dígitos. A imagem após remoção dos pequenos componentes conexos é mostrada na Figura 5b. Feito isso, é calculado o histograma de projeção vertical (Figura 5c) da imagem mostrada na Figura 5b, para estabelecer o ponto inicial e final ( $X_1$ ,  $Y_1$ ,  $X_2$ ,  $Y_2$ ) da região de cada dígito (Figura 5d).

#### 3.3 Reconhecimento dos dígitos com base na matriz contendo os valores dos *pixels*

A região que representa cada dígito é, de fato, uma matriz contendo valores binários. Assim, para determinar qual o dígito representado pela matriz (ou região), deve-se compará-la com cada uma das matrizes-padrão que representam os dez dígitos, usando para tanto alguma medida de similarida-



(a) Imagem de entrada do algoritmo.



(b) Imagem limiarizada.



(c) Histograma de projeção horizontal da imagem mostrada na Figura 4b.



(d) Região de interesse da imagem capturada.

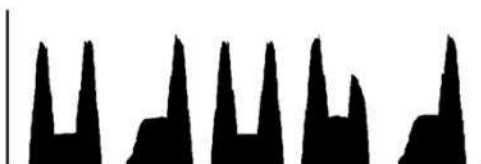
Figura 4: Passos do processo de determinação da RI



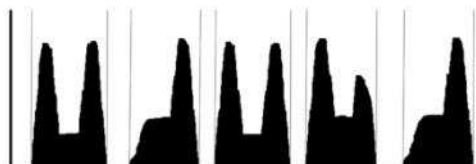
(a) RI limiarizada



(b) RI após limiarização e remoção dos componentes conexos pequenos



(c) Histograma de projeção vertical da RI



(d) Determinação da posição dos dígitos

Figura 5: Pré-processamento da RI

de. Entre as medidas de similaridade mais conhecidas estão a distância Euclidiana (Equação 2) e a Correlação (LONG; ZHANG; FENG, 2005). Devido à simplicidade do cálculo, em todas as comparações de similaridade realizadas neste trabalho, empregou-se a distância Euclidiana (Equação 2).

$$dE(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^N (x_i - y_i)^2} \quad (2)$$

Em que  $N$  é o número de elementos dos conjuntos de atributos  $x$  e  $y$ . Quanto menor a distância, mais parecidos são os conjuntos  $x$  e  $y$ .

A Figura 6 ilustra exemplos de avaliação de similaridade entre regiões que representam dígitos.

<table><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table> <p>(a) Matriz representando a região do dígito 5 (referência)</p>	0	0	0	0	0	0	1	1	1	1	0	1	1	1	1	0	0	0	0	0	1	1	1	1	0	1	1	1	1	0	0	0	0	0	0	<table><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr></table> <p>(b) dE=1,4</p>	1	0	0	0	0	0	1	1	1	1	0	1	1	1	1	0	0	0	0	0	1	1	1	1	0	1	1	1	1	0	0	0	0	0	1	<table><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table> <p>(c) dE=2,0</p>	0	0	0	0	0	1	1	1	1	0	1	1	1	1	0	0	1	1	1	0	0	0	0	0	0	1	1	1	1	0	1	1	1	1	0	0	0	0	0	0	<table><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr></table> <p>(d) dE=3,0</p>	0	1	1	1	0	0	1	1	1	0	0	1	1	1	0	0	0	0	0	0	1	1	1	1	0	1	1	1	1	0	1	1	1	1	0	1	1	1	1	0
0	0	0	0	0																																																																																																																																																					
0	1	1	1	1																																																																																																																																																					
0	1	1	1	1																																																																																																																																																					
0	0	0	0	0																																																																																																																																																					
1	1	1	1	0																																																																																																																																																					
1	1	1	1	0																																																																																																																																																					
0	0	0	0	0																																																																																																																																																					
1	0	0	0	0																																																																																																																																																					
0	1	1	1	1																																																																																																																																																					
0	1	1	1	1																																																																																																																																																					
0	0	0	0	0																																																																																																																																																					
1	1	1	1	0																																																																																																																																																					
1	1	1	1	0																																																																																																																																																					
0	0	0	0	1																																																																																																																																																					
0	0	0	0	0																																																																																																																																																					
1	1	1	1	0																																																																																																																																																					
1	1	1	1	0																																																																																																																																																					
0	1	1	1	0																																																																																																																																																					
0	0	0	0	0																																																																																																																																																					
1	1	1	1	0																																																																																																																																																					
1	1	1	1	0																																																																																																																																																					
0	0	0	0	0																																																																																																																																																					
0	1	1	1	0																																																																																																																																																					
0	1	1	1	0																																																																																																																																																					
0	1	1	1	0																																																																																																																																																					
0	0	0	0	0																																																																																																																																																					
1	1	1	1	0																																																																																																																																																					
1	1	1	1	0																																																																																																																																																					
1	1	1	1	0																																																																																																																																																					
1	1	1	1	0																																																																																																																																																					

**Figura 6: Exemplos de avaliação de similaridade entre regiões usando distância Euclidiana (dE)**

Dada a fragilidade dessa medida com relação a ruídos existentes nas imagens, essa abordagem mostrou-se menos robusta que as duas outras descritas a seguir.

### 3.4 Reconhecimento dos dígitos usando histogramas de projeção

Como pode ser visto na Figura 5d, o histograma é segmentado por valores nulos que indicam os espaços entre os dígitos. Assim, cada seg-

mento do histograma pode ser utilizado como um vetor de atributos descritor. Para verificar qual dígito o descritor representa basta compará-lo com vetores que representam os dígitos de 0 a 9, previamente armazenados em uma base, usando uma medida de similaridade. Por razões já explicadas empregou-se a distância Euclidiana.

Na Figura 7, é ilustrado um exemplo de comparação do primeiro segmento do histograma de projeção vertical mostrado na Figura 5d com os histogramas padrões dos dígitos 0 a 9.

### 3.5 Reconhecimento dos dígitos usando segmentos de retas adjacentes

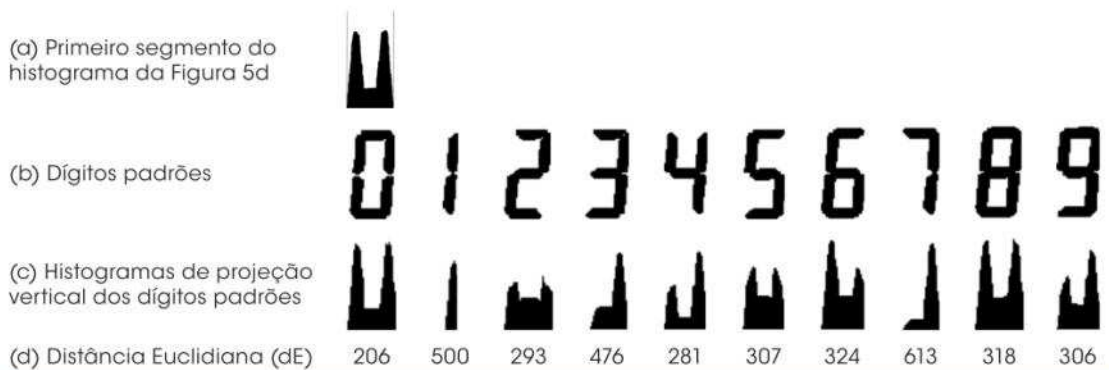
No reconhecimento usando o conceito de segmentos de retas adjacentes, verificam-se as posições dos segmentos de retas que aparecem na imagem analisado, como exemplificado na Figura 8.

Tendo-se o conhecimento de quais segmentos estão presentes na imagem analisada, basta consultar uma tabela com as configurações de todos os dígitos (Tabela 2), para decidir qual o dígito contido na imagem.

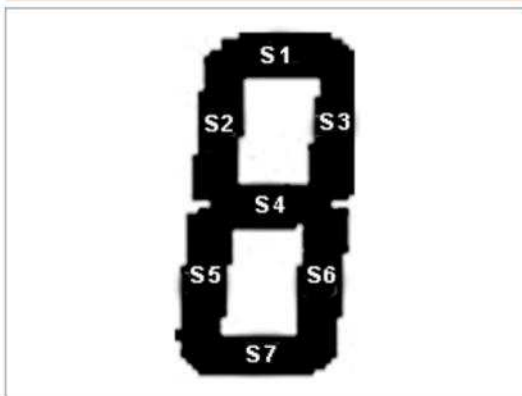
Como pode ser visto na imagem ilustrada na Figura 8, os segmentos identificados coincidem com os esperados para o dígito 8, conforme mostra a Tabela 2.

## 4 Resultados e discussão

O primeiro método empregado, baseado na matriz que representa o dígito (seção 3.3), apresentou um erro médio de 3%. Se for analisado apenas o erro médio, tem-se a impressão de que é um método funcional, mas quando são observados os erros cometidos para cada dígito verifica-se que esse método se mostra ineficiente para alguns dígitos como, por exemplo, “0”, “5” e “8”, conforme mostrado na Tabela 3.



**Figura 7:** Cálculo das distâncias entre o histograma de projeção vertical (figura 7a) e os histogramas de projeção vertical dos dígitos padrões



**Figura 8:** Segmentos de retas adjacentes presentes na imagem do dígito 8

**Tabela 2:** Segmentos de retas presentes em cada dígito

Dígito/ Segmento	S1	S2	S3	S4	S5	S6	S7
0	X	X	X	-	X	X	X
1	-	-	X	-	-	X	-
2	X	-	X	X	X	-	X
3	X	-	X	-	X	X	X
4	-	X	X	X	-	X	-
5	X	X	-	X	-	X	X
6	X	X	-	X	X	X	X
7	X	-	X	-	-	X	-
8	X	X	X	X	X	X	X
9	X	X	X	X	-	X	-

**Tabela 3:** Resultados dos testes realizados com base na distância dos valores dos *pixels* das regiões que representam os dígitos na imagem

Dígito	Distribuição dos Dígitos	Erros	% Erros
0	31	4	12%
1	29	0	0%
2	20	0	0%
3	18	0	0%
4	18	0	0%
5	17	1	6%
6	21	0	0%
7	19	0	0%
8	23	1	5%
9	19	0	0%
Total	215	6	3%

Como pode ser visto na Tabela 3, nessa abordagem o reconhecimento do dígito “0” é instável devido a sua aparência com o dígito “8”. Já os demais dígitos não apresentaram grandes problemas, visto que só se obteve um erro para os dígitos “5” e “8”. Esses erros mostraram a necessidade de buscar outras formas de reconhecimento e, por isso, foram testados alguns outros métodos, tais como a comparação de histogramas de projeção (Seção 3.4) e a análise de segmentos de retas adjacentes (Seção 3.5).

O método de comparação dos histogramas de projeção mostrou-se eficaz no reconhecimento dos dígitos, obtendo 100% de acertos nas 43 imagens testadas. Não obstante, a abordagem usando segmentos das retas adjacentes também não apresentou erros. Vale ressaltar que os desempenhos dessas duas técnicas se devem, principalmente, ao pré-processamento aplicado nas imagens. Dessa forma, se o pré-processamento não for bem-sucedido as técnicas de reconhecimento podem apresentar erros. Os resultados sugerem que ambas podem ser aplicadas na automatização do processo de calibração, com maior garantia de veracidade dos dados, podendo resultar em 30% de redução no tempo gasto no processo de calibração e emissão dos certificados.

Por fim, ressalta-se que o fato de as técnicas apresentadas demandarem pouca carga computacional viabiliza suas utilizações em tarefas que necessitem operar em tempo real.

## 5 Conclusão

Neste trabalho, explorou-se a utilização de técnicas de visão computacional para automatização do processo de calibração de instrumentos de medição sem interface de comunicação de dados, com vistas à redução dos tempos gastos na calibração e na emissão de certificado e aumento da produtividade. Para tanto, três diferentes abordagens para reconhecimento dos dígitos foram apresentadas e avaliadas.

Com base nos resultados obtidos, pode-se constatar que os métodos de reconhecimento baseados na comparação de histogramas de projeção e em segmentos de retas adjacentes mostraram ser confiáveis apresentando 100% de acertos.

Ainda com base nos resultados, foi possível inferir que o tempo médio gasto em cada calibração, incluindo a emissão do certificado, pode ser

reduzido em aproximadamente 30%, tornando-se uma alternativa viável para os laboratórios de calibração. Os valores de referência para obtenção do tempo médio gasto foram adquiridos em um laboratório de calibração, o qual realizou os testes comparando o método tradicional com o método proposto.

Pretende-se, em trabalhos futuros, estudar a aplicação de outras técnicas de visão computacional na automatização de outros tipos de equipamentos de medição, tais como termômetros do tipo TLV e instrumentos analógicos.

## Referências

- ALEGRIA, F. C.; SERRA, A. C. Automatic calibration of analog and digital measuring instruments using computer vision. *IEEE Transactions on Instrumentation and Measurement*, v. 49, n. 1, p. 94-99, 2000.
- ANDRIA, G. et al. Automatic calibration system for digital instruments without built-in communication interface. *XIX IMEKO World Congress*, Lisbon, September 6-11, p. 857-860, 2009.
- ARAÚJO, S. A. *Casamento de padrões em imagens digitais livre de segmentação e invariante sob transformações de similaridade*. Tese [Doutorado]—Universidade de São Paulo, USP, São Paulo, 2009.
- FERNÁNDEZ, E. V. et al. A machine vision system for the calibration of digital thermometers. *Measurement Science and Technology*, v. 20, n. 6, p. 1-7, 2009.
- FRACASSO, P. T.; SILVA, T. J.; SELVATICI, A. H. P. Bancada inteligente para calibração de hidrômetros. In: CONGRESSO INTERNACIONAL DE AUTOMAÇÃO, SISTEMAS E INSTRUMENTAÇÃO, 14., 2010. São Paulo, p. 1-10, 2010.
- GONZALEZ, R. C.; WOODS, R. E. *Processamento de imagens digitais*. São Paulo: Edgard Blucher, 2000. 509 p.
- HE, W. et al. Intelligent reading method for analog meter based on computer vision. *Chinese Journal of Scientific Instrument*, v. 28, n. 7, p. 1234-1239, 2007.
- INMETRO. *Regulamentação metrológica: resolução CONMETRO nº 11/88*. 3. ed. Rio de Janeiro: SENAI, 2007.
- KIM, H. Y. *ProEikon Biblioteca de processamento de imagens e visão computacional*. Disponível em: <http://www.lps.usp.br/~hae/software>. Acesso em: fev. 2010.



KIM, H. Y.; ARAÚJO, S. A. Grayscale template-matching invariant to rotation, scale, translation, brightness and contrast. *IEEE Pacific-Rim Symposium on Image and Video Technology, Lecture Notes in Computer Science*. v. 4.872, p. 100-113, 2007.

LONG, F.; ZHANG, H.; FENG, D. Multimedia information retrieval and management: technological fundamentals and applications. New York: Springer-Verlag, 2005. p. 1-26.

OTSU, N. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 9, n. 1, p. 62-66, 1979.

PEDRINI, H.; SCHWARTZ, W. R. *Análise de imagens digitais: princípios, algoritmos e aplicações*. São Paulo: Thomson Learning, 2007. 528 p.

Recebido em 26 ago. 2011 / aprovado em 26 set. 2011

#### Para referenciar este texto

BELAN, P. A.; ARAÚJO, S. A.; LIBRANTZ, A. F. H. Técnicas de visão computacional aplicadas no processo de calibração de instrumentos de medição com *display* numérico digital sem interface de comunicação de dados. *Exacta*, São Paulo, v. 10, n. 1, p. 1-300, 2012.

**ANEXO C – ARTIGO ACEITO PARA PUBLICAÇÃO NA REVISTA  
MEASUREMENT, 2012.**

## ARTICLE IN PRESS

Measurement xxx (2012) xxx–xxx



Contents lists available at SciVerse ScienceDirect

## Measurement

journal homepage: [www.elsevier.com/locate/measurement](http://www.elsevier.com/locate/measurement)

## Segmentation-free approaches of computer vision for automatic calibration of digital and analog instruments

P.A. Belan, S.A. Araujo\*, A.F.H. Librantz

Industrial Engineering Post Graduation Program, Universidade Nove de Julho (UNINOVE), São Paulo, Brazil

## ARTICLE INFO

## Article history:

Received 21 November 2011

Received in revised form 2 March 2012

Accepted 12 June 2012

Available online xxxx

## Keywords:

Calibration

Computer vision

Digital instruments

Analog instruments

Template matching

Radial projections

## ABSTRACT

A common problem found in the calibration laboratories is the reliability of the results obtained from calibration of instruments, especially when they do not have a built-in communication interface. In this case, the time consuming is increased significantly and the calibration may be subject to human error. Thus, many approaches based on computer vision have been proposed in the literature for automating calibration processes. However, most of them first simplify the images, usually using operations such as segmentation/binarization, for after proceeds the recognition of the digits. The issue lies in the fact that these simplifying operations throw away the rich grayscale information, decreasing the robustness of the algorithms, mainly when the images are affected by illumination changes, noise or JPEG compression. In this paper it was proposed two segmentation-free algorithms for automating the calibration process of digital and analog measuring instruments without built-in communication interface. The first one is based on template matching with normalized cross correlation for reading the display digits and the second uses radial projections and Bresenham algorithm to determine the pointer position in analog instruments. The experimental results showed that the proposed algorithms presented high accuracy and performance and can be used in real time systems applied to calibration processes.

© 2012 Elsevier Ltd. All rights reserved.

### 1. Introduction

A common problem found in the calibration laboratories nowadays is the productivity of their technicians, the reliability of the results obtained in the calibration process and the transcript of the calibration certificate [1]. This is mainly caused by the fact that, in most cases, the technician who performed the calibration service does not issue the certificate, but only signs it after the filling done by a typist. In some cases the result in the certificate can be different that one obtained during the calibration process, frequently caused by human errors in transcription data.

The calibration processes accuracy may be significantly affected when instruments without built-in communication interface are involved because the several manuals

readings and transcriptions of the data are more subject to human errors. Moreover, this task is time consuming and stressful [1,2].

Thus, computer vision systems play a very important role nowadays in the calibration of measuring instruments because they provide greater accuracy, repeatability and cost savings, beyond the reduction of monotonous and complex tasks [1–3].

Computer vision can be defined as a sub-area of image processing that studies the development of methods and techniques that enable a computer system to recognize objects in images imitating some capabilities of the human visual system, as the ability to describe a scene contained in a digital image.

Indeed, an efficient computer vision system must be able to extract a set of attributes that accurately describes a scene and small enough to reduce the spent processing time to be used in practical applications such as robot vision systems, autonomous vehicles, surveillance systems,

\* Corresponding author. Tel.: +55 11 36659347.

E-mail addresses: [pbelan@gmail.com](mailto:pbelan@gmail.com) (P.A. Belan), [saraujo@uninove.br](mailto:saraujo@uninove.br) (S.A. Araujo), [librantz@uninove.br](mailto:librantz@uninove.br) (A.F.H. Librantz).

## ARTICLE IN PRESS

2

P.A. Belan et al. / Measurement xxx (2012) xxx–xxx

automatic license plate recognition, industrial inspection and biometrics patterns recognition.

In the last years, many authors have proposed automatic calibration approaches using computer vision techniques [1–7].

Alegria et al. developed an automatic calibration system that can automatically determine the reading of the measuring instruments even if they do not have a digital interface that enables the computer to communicate with it. Image analysis algorithms were used to automatically determine the instrument reading. It could be used with analog and with digital displays [1,2].

Andria et al. proposed a solution for metrological calibration of digital instruments without built-in communication interface (DIWIs) which was based on the conversion in numerical data of video camera images of DIWI display [3].

Fernandez et al. proposed a system for the calibration of portable measurement devices that used machine vision to obtain the numerical values shown by displays, based on human perception of digits [4].

Hemming et al. presented a machine vision-based equipment for the calibration of dial indicators developed at the Centre for Metrology and Accreditation in Finland, with emphasis on the calculation of measurement uncertainty [5]. They also presented an updating of developed equipment into a calibration device for micrometers [6].

He et al. proposed a method based on computer vision for automatic reading of analog meter, which is able to deal with reading error caused by imaging distortion [7].

However, most of them, for example [1–5], firstly simplify the images using some operations to detection of edges, detection of interest points and segmentation/binarization for after proceeding the digits recognition or pointer position detection. The problem is that these image-simplifying operations throw away the rich grayscale information, decreasing the robustness of the technique mainly when images are affected by illumination changes, noise and JPEG compression. Thus techniques able to work in different conditions of image acquisition are required.

In this context, two free-segmentation approaches are presented in this work, in order to automate the process of calibration of digital and analog measuring instruments without built-in communication interface. For digital instruments it was proposed an algorithm using template matching with normalized cross correlation for reading the display digits whereas the algorithm for analog instruments is based on radial projections and Bresenham line drawing algorithm to locate the pointer position. Both algorithms store automatically the calibration values in a database, reducing the possibility of human errors in data reading/transcription and parallax view.

The remainder of the paper is organized as follows. Section 2 describes the used methodology. In Section 3, the proposed algorithm for digital instruments and the template matching technique, used for it, are described. Section 4 describes the proposed algorithm for analog instruments, including pointer detection and Bresenham algorithms. In Section 5 some experimental results are shown and finally, Section 6 concludes the paper and shows our plans for future works.

## 2. Methodology

The experiments involving digital instruments were performed using a standard digital manometer Zurich (model Z-10). Regarding analog instruments, it was used three different dial gage from distinct trademarks: Mitutoyo (model 1044F), Starrett (model 3025-481), Pantec (model 12307NJ-10) and one analog multimeter ICEL (model MA-60 200 MOHMS). To compare the result of the dial it was used a Mitutoyo Linear Gage (LGF Series), which shows the errors found on the dial gage.

The used images and videos have been acquired in a laboratory of calibration with different conditions of illumination. For digital instruments we captured 50 images. To evaluate the algorithm for analog instruments it were captured six videos, two of each dial gage, considering the full range of the instrument. Each video has about 30 s, with 30 fps (frames per second) totalizing 900 frames per video to be analyzed.

Besides the average times spent in calibration process used as reference were acquired in a laboratory of calibration, which conducted the tests comparing the traditional method with the proposed method.

The algorithms were implemented in C/C++ language using OpenCV and Proeikon libraries [8,9] and their working depends on the measuring instruments features such as shape and size of digits that appear in the display; shape, colors, range, diameter and length of the pointer of the dial indicator; number of laps that is necessary to contemplate the full range of instrument and the accuracy involved.

Thus, in the beginning of the calibration process, the technician must inform the model of the equipment that will be calibrated for setting the adequate parameters used by algorithms. This operation presupposes the use of a database.

This process of calibration for analog and digital instruments can be represented as shown in Fig. 1.

## 3. Proposed algorithm for digital instruments

The proposed algorithm for digits recognition is divided in two main steps: regions of interest (ROIs) determination and recognition of the digits. In both, template matching technique is used. Thus, we first present this technique.

### 3.1. Template matching with normalized cross correlation

One of the most known technique in computer vision systems is the template matching and it is used to find small image parts, called template, in another image that has been analyzed [10–12].

Let  $A$  be the image to be analyzed and  $T$  the query template. The goal of a template matching algorithm is to find all occurrences of  $T$  in  $A$ , as illustrated in Fig. 2.

Template matching can be done by two ways: feature-based or template-based. The first way uses the features of the  $T$  and  $A$  to find the best matching locations of  $T$  in  $A$ . The second, also called global approach, uses the entire template to determine the matching positions. In both

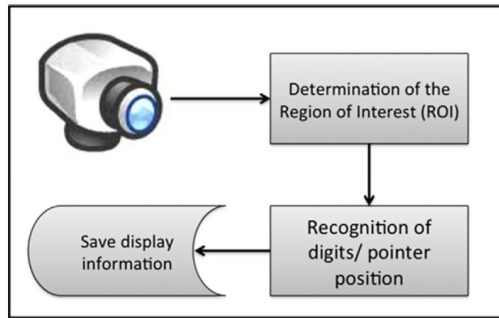


Fig. 1. Diagram for automatic calibration of analog and digital instruments.

cases, some similarity measure needs to be used. Template-based approaches can be conducted using, for example, sum of square differences (SSD), sum of absolute differences (SAD) or normalized cross correlation (NCC) measures [11].

In most cases the instances of  $T$  in  $A$  may appear rotated, scaled, shifted and with diverse brightness and contrast. Due to this fact template matching technique is well known to be an expensive operation and the execution time is proportional to the size of the images  $A$  and  $T$ .

The simplest implementation of template matching algorithm, using global approach, invariant to rotation and scale is the “brute force” solution. It performs a series of template matchings between the image to be analyzed  $A$  and the instances of the query template image  $T$  rotated by every angles and scaled by every scale factors, considering some specified range of angles and scale factors [11,13].

The brute force algorithm yields the most precise solution to this problem. Nevertheless, it is very time consuming and thus is not feasible to be used in practical applications. However, this technique can be very efficient if does not have the necessity of rotation and scale invariance.

Normalized cross correlation (NCC) is widely used as a similarity measure in template matching and it is the most robust correlation measure for determining similarity between points in two or more images [12].

Considering an image  $A$  of size  $M \times N$  pixels and a template  $T$  of size  $I \times J$  pixels, then the cross correlation between  $A$  and  $T$ , denoted by  $CC(A, T)$ , is defined as:

$$CC(A, T) = C(m, n) = \sum_{i=-a}^a \sum_{j=-b}^b T(i, j) A(m+i, n+j) \quad (1)$$

where  $a = (I-1)/2$ ,  $b = (J-1)/2$ ,  $m = 0, \dots, M-1$  and  $n = 0, \dots, N-1$ ,  $C$  is the result matrix, with the same size of  $A$ , in which the peaks indicating the occurrences of  $T$  in  $A$  (matching positions).

In many applications the brightness of the images can vary according to environment lighting conditions. In these cases, the images can be firstly normalized. Thus,  $NCC(A, T)$  can be defined as:

$$NCC(A, T) = C(m, n) = \frac{\sum_{i=-a}^a \sum_{j=-b}^b [T(i, j) - \bar{T}][A(m+i, n+j) - \bar{A}]}{\sqrt{\sum_{i=-a}^a \sum_{j=-b}^b [T(i, j) - \bar{T}]^2 \sum_{i=-a}^a \sum_{j=-b}^b [A(m+i, n+j) - \bar{A}]^2}} \quad (2)$$

The computation of  $NCC(A, T)$ , defined in Eq. (2), generates a matrix  $C$  in which the values vary from  $-1$  to  $+1$ . Depending on the application, we can use either the absolute value  $|C(m, n)|$  to allow matching negative instances of the  $T$  in  $A$  or the value of  $C(m, n)$  with signal for not allowing the matching of negative instances of  $T$ .

Fig. 3 depicts the result of NCC between  $T$  and  $A$ . The highest peaks indicate the occurrences of instances of  $T$  in  $A$ .

In practice, template matching with NCC is performed by moving the center (or the origin) of the template  $T$  over each  $(m, n)$  point in the  $A$  and calculate the sum of products between the coefficients in  $A(m, n)$  and  $T(i, j)$  over the whole area spanned by  $T$ .

The NCC computation is high costly in time. However, the most available library implementation is optimized

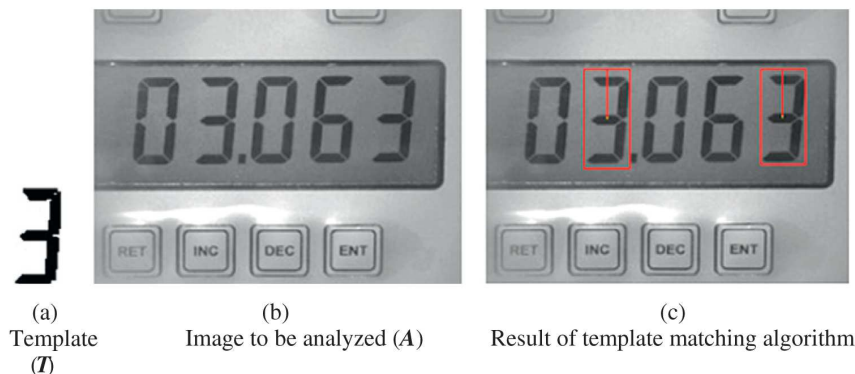


Fig. 2. Example of template matching.

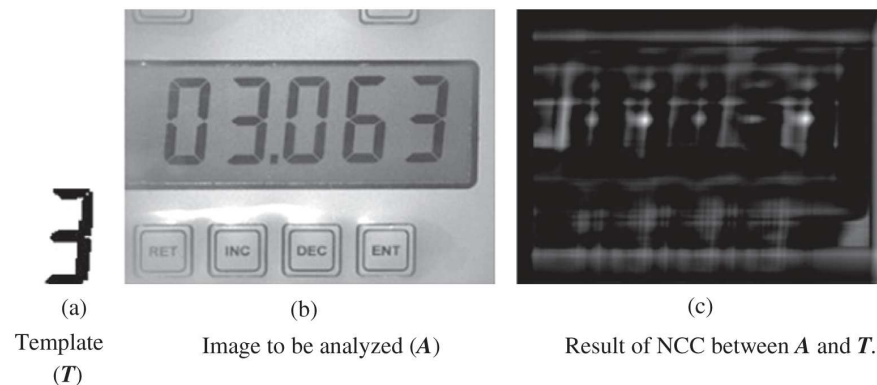


Fig. 3. NCC between two images.

and makes use of Fast Fourier Transform (FFT), where the correlation sums are reduced to matrix products, accelerating the processing time. That is the case of the OpenCV library [8], used in our implementations.

### 3.2. ROIs determination

In the first step of the algorithm a color image of size  $851 \times 638$  is acquired by a conventional low cost webcam, converted at 256 gray levels and reduced to 30% of its original size. One example of input image is shown in Fig. 3b.

Since the template matching technique may be an expensive operation regarding processing time, the reduction of searching area is important to accelerate the algorithm.

In order to detect the ROIs automatically, firstly NCC between input image and the template of digit 8 ( $T_8$ ), denoted by  $NCC(A, T_8)$ , is computed generating the image C, illustrated in Fig. 3a, in which the highest peaks give the coordinates of the line crossing the vertical center of the digits (Fig. 4b).

In addition, by determining a threshold value  $t$ , it is possible to select the five highest peaks (Fig. 4c), which indicate the horizontal center of each digit (Fig. 4d).

Using the coordinates that denote the center of the digits, the initial and final position of each subregion  $S_0, \dots, S_4$  is finally calculated, as shown in Fig. 5.

It is valid to remember that the decimal point does not need to be recognized because it is fixed by technician, according to the instrument that is been analyzed.

### 3.3. Recognition of the digits

In this step, the digits of the display are recognized and stored in a database. For this purpose, first the NCC is calculated between each  $S_i$  (Fig. 5) and each one of the templates  $T_i$  (Fig. 6) and the maximum value of correlation is stored in the matrix  $P$ , containing five rows (subregions of  $A$ ) and 10 columns (templates), as follows:

$$P[l, c] = \text{MAX}[NCC(S_l, T_c)] \quad (3)$$

where  $l = 0, \dots, 4$  and  $c = 0, \dots, 9$ .

The index  $c$  of the highest value of each line in the matrix  $P$  indicates the value displayed in the analyzed instrument on the position  $l$ . Thus, the output data consists of an integer vector  $D$  containing five elements. This vector is obtained as follows:

$$D[l] = \underset{c=0}{\text{ARGMAX}}[P[l, c]] \quad (4)$$

where  $c$  is the argument that maximizes  $P[l, c]$ .

Fig. 7 illustrates the matrix  $P$  calculated to recognize the display digits depicted in Figs. 2, 3 and 5.

Finally, the ASCII values from vector  $D$  representing the displayed digits, a display image's copy and other data about the calibration process are stored in a database to be used in the instrument certificate or to identify some mistake occurred in the digits recognition.

## 4. Proposed algorithm for analog instruments

The reading of an analog instrument can be obtained by determining the position of the pointer in relation to the scale limits [3]. The proposed algorithm uses projections on a set of radial lines and the Bresenham line drawing algorithm to detect the position of the pointer.

### 4.1. The Bresenham line drawing algorithm

The Bresenham algorithm is used to determine which points should be plotted in order to create an approximate straight line between two given points. It is common used to draw lines on the computer screen. The main purpose of this algorithm is to reduce the computational effort to draw a straight line, as well as reduce rounding errors and floating point operations. Thus, only integer additions and subtractions, which are very cheap operations in standard computer architectures, are used. It was one of the first developed algorithms in the field of computer graphics and it has been extended for drawing circles [14,15].

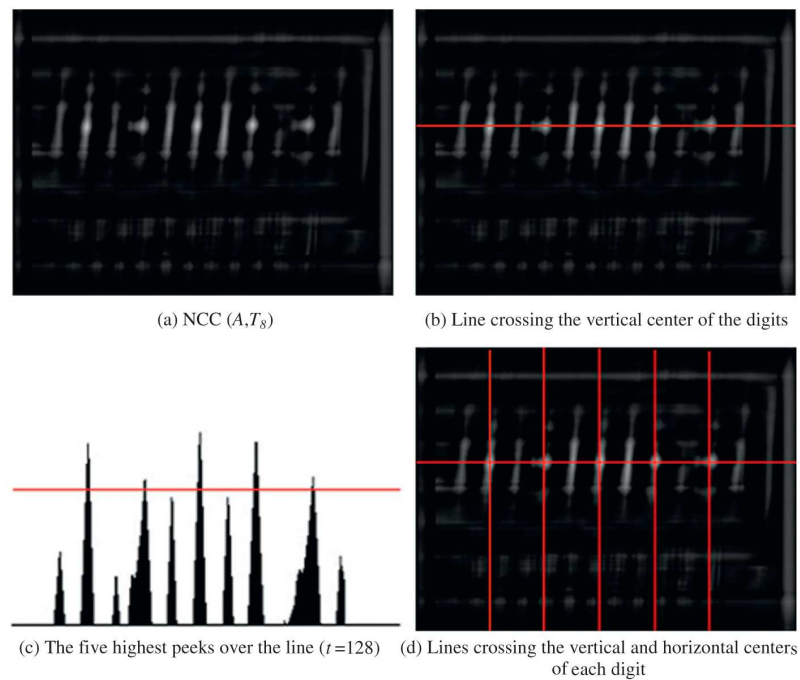


Fig. 4. ROIs determination process.



Fig. 5. Detected subregions.

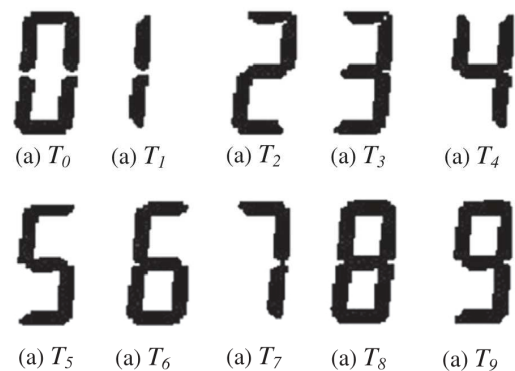


Fig. 6. Query templates used in the recognition step.

#### 4.2. Pointer detection algorithm

The analog equipment, such as the dial gages display usually contains a pointer, to indicate the measure that is being read. Thus, the algorithm needs to be able to detect the inclination angle of the pointer to determine the read value based on features of the analyzed instrument. For that, we used the idea of radial sampling proposed in [11,13], replacing the average grayscale by sum.

Given an image  $I$ , the radial sampling  $Ras_i^j(x, y, \alpha)$  can be defined as the sum of the pixels of  $I$  located on the radial line with one vertex at  $(x, y)$ , length  $\lambda$  and inclination  $\alpha$ :

	0	1	2	3	4	5	6	7	8	9
0	215	149	106	122	112	93	148	184	176	127
1	133	144	137	219	103	124	101	163	139	163
2	223	150	104	130	100	112	162	178	181	136
3	146	152	80	126	93	188	227	168	171	125
4	114	150	127	225	110	129	97	162	138	164

Fig. 7. Matrix  $P$  calculated in the recognizing of the display containing the digits "0", "3", "0", "6" and "3". The correlation values ranging from  $-1$  to  $1$  were converted to the interval  $[0, 255]$ .

$$Ras_j^{\lambda}(x, y, \alpha) = \int_0^{\lambda} I(x + t \cos \alpha, y + t \sin \alpha) dt \quad (5)$$

In practice, the integral must be replaced by a sum and the Bresenham line drawing algorithm can be used to find efficiently all the pixels that belong to a line.

Given length of  $\lambda$  (according to the instrument that is being analyzed) and a set of  $k$  angle inclinations  $(\alpha_0, \alpha_1, \dots, \alpha_{k-1})$ ,  $I$  is radially sampled, yielding a vector  $R_l$  with  $k$  values:

$$R_l[j] = Ras_j^{\lambda}(x_c, y_c, \alpha_j), \quad 0 \leq j < k \quad (6)$$

where  $(x_c, y_c)$  is the central pixel.

Then, the inclination angle of the pointer ( $a_p$ ) is defined by:

$$a_p = \underset{j=1}{\operatorname{ARGMIN}}[R_l[j]] \quad (7)$$

where  $j$  is the argument that minimizes  $a_p$ .

As mentioned before, using the inclination angle the indicator value can be determined and stored in a database to be used in the instrument certificate or to identify some occurred mistake.

Fig. 8 shows one example of the proposed methodology, where the radial sampling of three different angles where computed.

It is worth mentioning that, if the pointer is light and the background of the instrument is dark, the inclination angle will be determined by argument  $j$  that maximizes  $a_p$ . This information, well as the diameter of the instrument to determine  $(x_c, y_c)$  and  $\lambda$  need to be stored in a database.

As the previous algorithm, a dial image's copy and other data about the calibration process are stored in a database to be used in the instrument certificate or to identify some mistake occurred in the digits recognition.

## 5. Results and discussion

In this section we discuss details of experimental setup for evaluating the performance of the proposed algorithms and the results obtained by each one of them, as well.

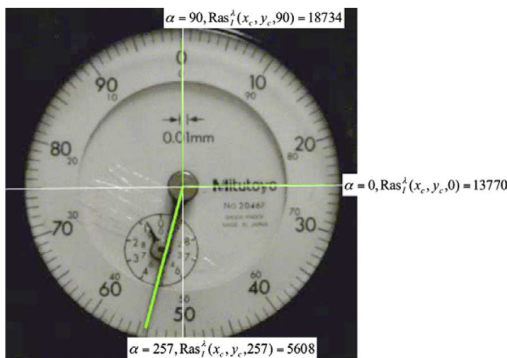


Fig. 8. Example of proposed methodology considering three values of  $\alpha$  (0, 90 and 257),  $\lambda = 150$ ,  $x_c = 175$  and  $y_c = 175$ . The image size is  $350 \times 350$  pixels.

### 5.1. Experimental results for digital instruments

In order to evaluate the robustness of the proposed algorithm we have conducted experiments using a set of 50 images affected by illumination changes (Fig. 9a and b); noise (Fig. 9c) and JPEG compression (Fig. 9d). In these experiments all 250 matching were perfect, without any false positive or false negative, even in the presence of some faintly visible digits (e.g. Fig. 9b). The distribution of the digits in the image set is shown in Table 1. As can be seen all 10 digits appear at least in 7.6% of the images.

Concerning the time of processing, the algorithm takes, in average, 0.3 s using a Pentium 4 2.8 GHz to read the numbers appearing in the display image. Nevertheless, this time could be further reduced by performing the template matching operation, in the second step, only in a few pixels around the peaks representing the center of the digits instead all pixels inside each subregion  $S_i$ . This reduction in processing time can allow the use of the algorithm in real-time computer vision systems applied to calibration processes. Results showed that this method could reduce the time spent in whole calibration process, could be reduced in approximately 30%.

### 5.2. Experimental results for analog instruments

To validate the proposed algorithm it were used four different dials. In all cases the central point of the gage was determined by using a support to fix the instrument and some information from the database. It is indicated by white crossed lines in Fig. 10a–d.

The answers returned by the algorithm are always integers since a reading of such equipments cannot be of another format. In all experiments we set  $k = 360$  angles.

The process does not require hard control of the environment once by this technique it's possible to detect the pointer in different lighting conditions. However, during the image acquisition shadows should be avoided since they can generate false positives. In Fig. 10 are illustrated some captured frames, each one showing the detected pointer position.

The tests involving dial gages were carried out with the dial gage pointer in motion and only the values multiples of 10 are considered. These values are compared to the values obtained by a Linear Gage and the found errors are stored in a database file to be used in the instrument certificate. A great advantage on computer vision is that it avoids the parallax error, which is natural in human vision given the difficulty of always to look for the instrument at same perspective point. As consequence, the average time spent in each calibration process, including the issuance of the certificate could be reduced in approximately 45%.

## 6. Conclusions

In this paper we have presented two segmentation-free algorithms in order to automate the process of calibration of digital and analog measuring instruments without built-in communication interface. Differently from most pro-

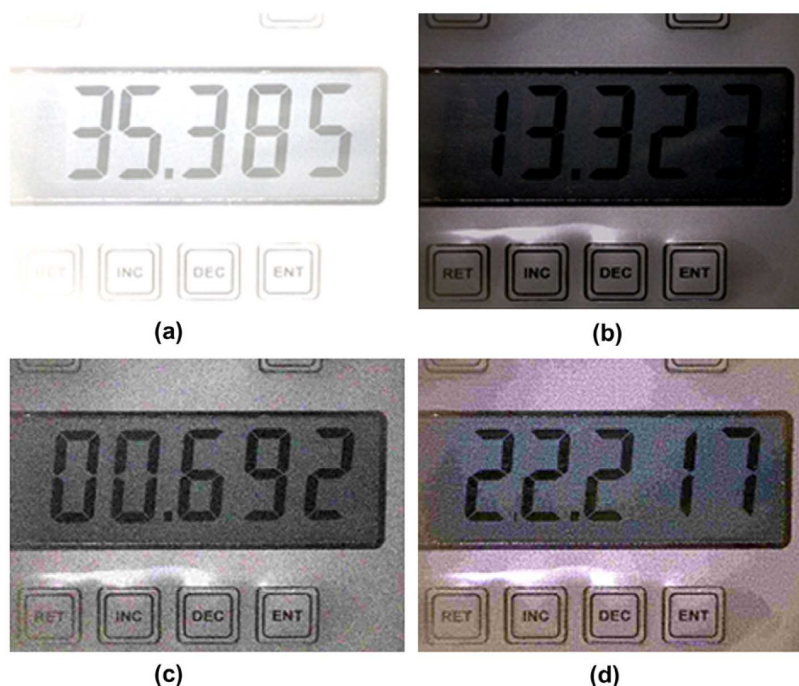


Fig. 9. NCC between two images.

**Table 1**  
Distribution of the digits in the set of images.

Digit	0	1	2	3	4	5	6	7	8	9
%	14.4	13.2	10	10	7.6	7.6	9.6	9.2	9.6	8.8

posed approaches found in the literature, the proposed algorithms do not discard the rich grayscale information through operations like detection of edges, detection of interest points and segmentation/binarization. Thus, their robustness is increased.

In the experiments regarding digital instruments, the set of digits in all 50 images were correctly recognized, even considering illumination conditions, noise, and JPEG compression. This algorithm took about 0.3 s, on average, for processing each image. Indeed, as described

before, its performance could be improved by reducing the ROIs.

With respect to algorithm for analog instruments, the tests were performed using 5.400 images or frames extracted from six videos. In this case, all analyzed frames have been correctly recognized. This algorithm demonstrated high accuracy and performance, once it was able to analyze with precision about 90 fps. The application of the proposed method allowed a reduction in the average spent time in the calibration process and its performance makes it suitable for applying in laboratories of calibration.

In future works we intend to aggregate the proposed algorithms in a real time system for dealing with a wide variety of digital and analog instruments, well as to develop specific hardware for automatic calibration of instruments.

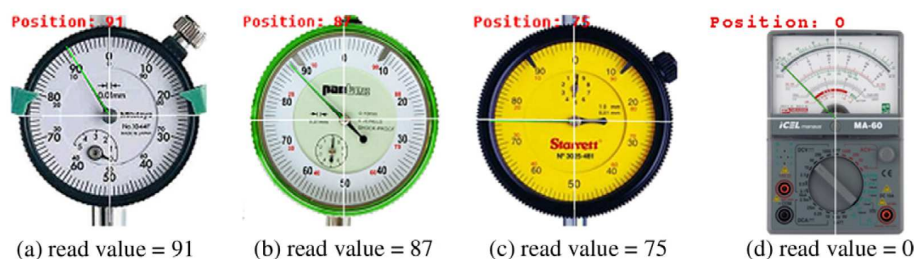


Fig. 10. Results of pointer detection algorithm.

## ARTICLE IN PRESS

8

P.A. Belan et al. / Measurement xxx (2012) xxx–xxx

**Acknowledgment**

The authors would like to thank Uninove by financial support and CAPES for the scholarship granted to one of the authors.

**References**

- [1] F.C. Alegria, A.C. Serra, Computer vision applied to the automatic calibration of measuring instruments, *Measurement* 28 (2000) 185–195.
- [2] F.C. Alegria, A.C. Serra, Automatic calibration of analog and digital measuring instruments using computer vision, *IEEE Transactions on Instrumentation and Measurement* 49 (2000) 94–99.
- [3] G. Andria et al., Automatic calibration system for digital instruments without built-in communication interface, in: *Anais do XIX IMEKO World Congress* (Lisbon), 2009, pp. 857–860.
- [4] E.V. Fernández et al., A machine vision system for the calibration of digital thermometers, *Measurement Science and Technology* 20 (2009) 1–7.
- [5] B. Hemming, H. Lehto, Calibration of dial indicators using machine vision, *Measurement Science and Technology* 13 (2002) 45–49.
- [6] B. Hemming, A. Fagerlund, A. Lassila, High-accuracy automatic machine vision based calibration of micrometers, *Measurement Science and Technology* 18 (2007) 1655–1660.
- [7] W. He et al., Intelligent reading method for analog meter based on computer vision, *Chinese Journal of Scientific Instrument* 28 (2007) 1234–1239.
- [8] Intel 2007, OpenCV – Open Source Computer Vision Library. <<http://www.intel.com/technology/computing/opencv/>>, (accessed 09.10.07).
- [9] H.Y. Kim, ProEikon – Library for Image Processing and Computer Vision. <<http://www.lps.usp.br/~hae/software>>, (accessed February, 2010).
- [10] R.C. Gonzalez, R.E. Woods, *Digital Image Processing*, Addison-Wesley, Massachusetts, 2002. p. 793.
- [11] H.Y. Kim, S.A. Araújo, Grayscale template-matching invariant to rotation, scale, translation, brightness and contrast, *Lecture Notes in Computer Science* 4872 (2007) 100–113.
- [12] A.J. Hii et al., Fast normalized cross correlation for motion tracking using basis functions, *Computer Methods and Programs in Biomedicine* 82 (2006) 144–156.
- [13] S.A. Araújo, H.Y. Kim, Ciratefi: an RST-invariant template matching with extension to color images, *Integrated Computer-Aided Engineering* 18 (2011) 75–90.
- [14] J.E. Bresenham, Algorithm for computer control of a digital plotter, *IBM Systems Journal* 4 (1965) 25–30.
- [15] J.R. Bresenham, A linear algorithm for incremental digital display of circular arcs, *Communications of the ACM* 20 (1977) 100–106.

**ANEXO D – ARTIGO ACEITO PARA PUBLICAÇÃO NA REVISTA  
INTERNATIONAL JOURNAL OF COMPUTER APPLICATIONS IN TECHNOLOGY,  
2012.**

---

## A comparative study of statistical methods for characterization of materials surface by means of texture analysis

---

Sidnei Alves de Araújo\*, Wonder Alexandre L. Alves,  
André F. H. Librantz and Peterson A. Belan

Industrial Engineering Post Graduation Program,  
Nove de Julho University (UNINOVE),  
Av. Francisco Matarazzo, 612, 05001-100, São Paulo, SP – Brazil  
E-mail: [saraujo@uninove.br](mailto:saraujo@uninove.br)

\*The author for correspondence

**Abstract:** Texture is an important attribute to distinguish objects and materials. Thus, along the decades many texture analysis methods have been proposed and utilized in a variety of application domains. Due to the fact there is not a generic method to describe a large variety of textures, comparative studies among the related methods became necessary. In this work it was performed a comparative study of the main statistical methods applied to materials surface characterization. In order to evaluate the performance of the compared methods, an unsupervised neural network was used to classify a set of 3000 textures images, divided in five categories, with different levels of details. Inferences from this work could assist those ones that intend to perform some tasks involving automatic inspection of texture, mainly in Materials Science context.

**Key words:** Texture, Statistical Descriptors, Materials Surface, Automatic Inspection, Artificial Neural Network, Moments of Histogram, Spatial Gray Level Dependence, Gray Level Difference, Sums and Differences Histograms and Gray Level Run Length Matrices.

**Reference:** to this paper should be made as follows: ...

**Biography notes:** Sidnei Alves de Araújo received his M.Sc. and his Ph.D. degrees in Electrical Engineering, respectively, from Mackenzie University in 2002 and from University of São Paulo (USP) in 2009. Since 2001 he has been teaching in Nove de Julho University and, currently, he is a postdoctoral student in Electrical Engineering at USP and Professor of the Industrial Engineering Post Graduation Program at Nove de Julho University. His research interests include image processing and analysis, computer vision, machine learning and processes optimization.

Wonder Alexandre Luz Alves received his B.Sc. degree in Computer Science (2005) from Nove de Julho University, his MSc degree in Computer Science (2010) from University of São Paulo (USP) and, currently, he is a Ph.D. candidate in Computer Science at USP. Since 2007 he has been teaching at Nove de Julho University, in Computer Science course. His research interests include image processing and analysis, computer vision, artificial intelligence and processes optimization.

André Felipe Henriques Librantz received his B. Sc. degree in Physics, and the M. Sc. And Ph.D. degrees in form the University of São Paulo (USP) in 1998, 2000 and 2004, respectively. In 2004 he was a postdoctoral research with the Nuclear and Energy Research Institute, IPEN. Since 2008 he has joined Industrial Engineering Post Graduation Program at Nove de Julho University. His current research interests include process optimization, numerical modeling and machine learning.

Peterson Adriano Belan received his B.Sc. in Computer Science (2004) from Nove de Julho University and currently he is a M.Sc. candidate in Industrial Engineering at the same university. His research interests include image processing and analysis, computer vision and artificial intelligence and processes optimization.

## 1 Introduction

The texture is a very important attribute to distinguish objects and materials. It can be found on walls and floors, metals, plastics, trees, rocks, skin of humans and animals, etc. Studies involving analysis of textures have been conducted for more than three decades (Zhang and Tan, 2002; Lazebnik, Schmid and Ponce, 2005) and until nowadays continues to be a topic of great interest in several areas of knowledge in which applications such as materials surface inspection, medical image processing, document processing, and remote sensing are explored.

Despite its importance, there are no universally accepted definition and categories of texture (Haralick, 1979; Gonzalez and Woods, 2002; Sang et al., 2010; Jucá and Mello, 2011). It can be addressed to the fact that we recognize texture when we see it but we have difficulties to define it (Tuceryan and Jain, 1998). Regarding the categories, in many works early reported, the textures are classified as smooth, rough and granular. These categories are also cited by Haralick (1979) and Gonzalez and Woods (2002). In addition, the following properties can be also found in texture description: uniformity, density, coarseness, roughness, regularity, linearity, directionality, frequency and phase (Tuceryan and Jain, 1998). Indeed, the properties of textures can cover ranges of possibilities from smooth to rough, coarse to fine, uniform to non-uniform, soft to hard, etc. (Mirmehdi, Xie and Suri, 2008).

According to Gonzalez & Woods (2002) the texture can be defined as a set of patterns existing in the physical surfaces that are noticeable to the human eye and have a lot of information about the nature of the surface such as smoothness, roughness and regularity.

In Materials Science area the texture can be seen as the preferential orientation of crystalline material in a polycrystalline material. The analysis of materials texture has been conducted for many objectives such as microstructural analysis of materials (Yan et al., 2009; Qaddoura and Belfield, 2010; Brokmeier, 2011; Yan et al., 2011; Zhang, Gong and Xu, 2012); prediction of tool wear in machining (Gadelmawla et al., 2009); corrosion detection (Kapsalas, et al. 2007; Zaidan et al., 2010) prediction of workpieces surface roughness (Ho et al., 2005; Ji et al., 2007) and texture reconstruction (Eisenlohr and Roters, 2008; Chen, Liu and Dong, 2010). In these cases, the characterization of texture is usually performed from images acquired by electronic microscopes and digital cameras.

Due to the fact there is no formal definition of texture, many methods to describe were proposed, each one more suitable to detect a particular type of feature. However, there are three main approaches to texture description: statistics, structural and spectral (Gonzalez and Woods,

2002). The statistical approach is based on histograms of first and second order, taking into account the distribution of gray levels of images. The structural approach takes into account primitives of textures, which are ordered according to syntactic rules to form patterns. In the spectral approaches, in general spectral properties of the Fourier Transform are used to describe the spatial frequency of patterns. Among the texture description methods, the statistical ones are the most well-known, easier to implement, and extensively used for describing texture features (Jucá and Mello, 2011; Tuceryan and Jain, 1998; Simoncelli and Portilla, 1998; Xie, 2008). In general, a statistical method generates a set of statistical measures also called descriptors.

Given the importance of the tasks involving analysis of textures and the existence of a variety of methods, in particular, the statistics, the following questions arise: (i) What is the power of statistical descriptors for characterization of material surfaces? (ii) Is there a more appropriate statistical method for a particular category of texture?

In order to answer these questions, it was conducted a study comparing the performance of the following statistical methods for texture analysis: (1) Moments of Histogram, (2) Spatial Gray Level Dependence, (3) Gray Level Difference, (4) Histogram of the Sums and Differences and (5) Gray Level Run Length Matrices, which are described in the next section. In the experiments we used a database containing 3000 images of textures equally divided into five categories: smooth, uniform granular, non-uniform granular, uniform rough and non-uniform rough.

It's worth mentioning that, in the last two decades, several works focusing on the comparison of methods for texture analysis were related in the literature. Among them, we can cite (Ohanian and Dubes, 1992; Chen, Nixon and Thomas, 1995; Smith and Burns, 1997; Chang, Bowyer and Sivagurunath, 1999; Singh and Singh, 2002; Varma and Zisserman, 2004; Zhao, Xu and Hong, 2009). However, none of them investigated the set of considered statistical methods and the texture categories proposed in this paper.

The remainder of this paper is organized as follows. In section 2, we described in details the five statistical methods used in this work to texture features extracting. These texture features will compose a set to build a classifier based on artificial neural networks, which is described in section 3. Materials and methods are described in section 4. Then, the comparative results of the five methods and the discussion about them are shown in section 5. Finally, we conclude this work in section 6 with some perspectives about the proposed study.

## 2 Statistical Descriptors for Texture Feature

### Extraction

The feature extraction through the statistical approach provides a description of images using statistical rules that govern the distribution and relationship between different gray levels. Several measures can be extracted from the distribution of gray levels in images. Among them we can mention: mean, variance, standard deviation, entropy, and others. In the statistical approach the descriptors may be first or second order. First order descriptors are extracted directly from the image histogram whereas the second order descriptors are extracted from gray level co-occurrence matrices. The problem of using the histograms of the first order is that, given two images with the same amount of gray levels differing only in terms of positioning, they present the same histogram whereas the second order approach, histograms can be different. This type of observation is important because the correct classification of any texture depends heavily on the descriptors used to characterize it.

Next subsections describe the statistical methods considered in this work. All descriptors are calculated from grayscale images whose definition is given by  $f: D_f \subset Z^2 \rightarrow \{0, 1, \dots, L-1\}$ , where  $L$  is  $2^b$  and  $b$  is the number of bits of image depth.

#### 2.1 Moments of Histogram (MH)

The MH is one of the simplest methods for describing textures and is based on statistics of the first order. Let  $h$  be a vector denoting the normalized histogram of an image  $f$ , the central moment  $f_{moment}$  of order  $n$  is given by

$$f_{moment}(n) = \sum_{k=0}^{L-1} (k - \mu)^n h(k), \quad (1)$$

where  $\mu$  is the average gray level of image  $f$ .

Varying  $n$  in Eq. 1, other important moments can be obtained. For example, for  $n = 2$  we obtain the variance that serves as a measure of contrast gray levels. Using  $n = 3$  and  $n = 4$ , can be obtained measures of skewness and kurtosis, and many others. The moments for  $n > 4$ , although providing information to describe the content of texture, they are not normally used (Gonzalez and Woods, 2002). For this reason, we use only the first four moments to describe texture.

#### 2.2 Spatial Gray Level Dependence (SGLD)

The SGLD, proposed by Haralick et al (Haralick, Shunmugan and Dinstein, 1973), extracts the textural information based on second order statistics, which are defined by features extracted from the co-occurrence matrices that consist of a count of how many different combinations of gray levels occur in an image in a certain

direction. To obtain these matrices, distance  $d$  and direction  $\theta$  between neighboring pixels were considered. These parameters are represented in the discrete case by  $\Delta x$  and  $\Delta y$ , as shown in table 1, and thus the co-occurrence matrix is defined as

$$M_{\Delta x, \Delta y}(i, j) = \sum_{(w, h) \in D_f} \begin{cases} 1, & \text{if } f(w, h) = i \text{ and } f(w + \Delta x, h + \Delta y) = j \\ 0, & \text{otherwise,} \end{cases} \quad (2)$$

where  $i$  and  $j$  are the indices of the co-occurrence matrix  $M_{\Delta x, \Delta y}$  obtained from the input image  $f$  and  $\Delta x$  and  $\Delta y$  are the parameters that correspond to the distance and direction of  $M_{\Delta x, \Delta y}$ , respectively.

Table 1. Parameters  $\Delta x$  and  $\Delta y$  that correspond to the distance  $d$  and direction  $\theta$ .

$\theta = 0 \text{ degrees}$	$\theta = 45 \text{ degrees}$	$\theta = 90 \text{ degrees}$	$\theta = 135 \text{ degrees}$
$\Delta x = d$	$\Delta x = d$	$\Delta x = 0$	$\Delta x = -d$
$\Delta y = 0$	$\Delta y = -d$	$\Delta y = -d$	$\Delta y = -d$

Based on the co-occurrence matrixes  $M_{\Delta x, \Delta y}$ , Haralick, Shunmugan and Dinstein (1973) determined various statistical measures known as Haralick descriptors. However, Ohanian and Dubes (1992) showed that only a small group of measures is sufficient to characterize the texture. For this reason, in this work, we used only eight measures: angular second moment (eq. 3), contrast (eq. 4), correlation (eq. 5), the inverse difference moment (eq. 6), maximum probability (eq. 7), means (eq. 8), dissimilarity (eq. 9) and entropy (eq. 10).

$$D_1 = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} M_{\Delta x, \Delta y}(i, j)^2 \quad (3)$$

The angular moment second measures the uniformity of the gray levels.

$$D_2 = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} (i - j)^2 M_{\Delta x, \Delta y}(i, j) \quad (4)$$

The contrast reflects the amount of gray level transitions in the region considered.

$$D_3 = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} \frac{(i - \mu_i)(j - \mu_j) M_{\Delta x, \Delta y}(i, j)}{\sigma_i \sigma_j} \quad (5)$$

The correlation measures the linear dependence of the gray levels in the combinations of pairs of pixels in a given direction. are the averages in the directions  $i$  and  $j$  whereas are the standard deviations.

2 S.A.Araújo, W.A.L. Alves, A.F.H. Librantz, -P. Belan

$$D_4 = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} \frac{1}{1+(i-j)^2} M_{\Delta x, \Delta y}(i, j) \quad (6)$$

The inverse difference moment measures the homogeneity of gray levels.

$$D_5 = \max \{M_{\Delta x, \Delta y}(i, j) : 0 \leq i \leq L-1 \text{ and } 0 \leq j \leq L-1\} \quad (7)$$

The maximum probability indicates the predominant orientation of the texture.

$$D_6 = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} i M_{\Delta x, \Delta y}(i, j) \quad (8)$$

Mean of co-occurrence matrix.

$$D_7 = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} |i-j| M_{\Delta x, \Delta y}(i, j) \quad (9)$$

The dissimilarity measures the degree of dissimilarity between pixels.

$$D_8 = - \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} M_{\Delta x, \Delta y}(i, j) \log_2 (M_{\Delta x, \Delta y}(i, j)) \quad (10)$$

The entropy provides the degree of dispersion of gray levels.

### 2.3 Spatial Gray Level Dependence (SGLD)

The GLD method consists of a texture analysis technique based on the difference between pairs of image pixels at a distance  $d$  and direction  $\theta$ . The resulting histogram  $Hgd$ , whose size is determined by the amount of gray level image  $L$ , indicates the probability of the absolute difference between the  $i$  pairs of image pixels. One can note that the  $i$  values of the absolute differences are between 0 to  $L-1$ .

$$Hgd_{\Delta x, \Delta y}(i) = \sum_{(w, h) \in D_f} \begin{cases} 1, & \text{if } |f(w, h) - f(w + \Delta x, h + \Delta y)| = i \\ 0, & \text{otherwise,} \end{cases} \quad (11)$$

where  $\Delta x$  and  $\Delta y$  are the parameters that correspond to the distance and direction, respectively.

The descriptors extracted from the histograms GLD are the same used for the histograms of the sums and differences, in other words, angular second moment (eq. 12), contrast (eq. 13), correlation (eq. 14), the inverse difference moment (eq. 15), variance (eq. 16), means (eq. 17) and entropy (eq. 18).

$$D_1 = \sum_{i=0}^{L-1} (Hgd_{\Delta x, \Delta y}(i))^2 \quad (12)$$

$$D_2 = \sum_{i=0}^{L-1} i^2 Hgd_{\Delta x, \Delta y}(i) \quad (13)$$

$$D_3 = \frac{\sum_{i=0}^{L-1} (i - \mu) Hgd_{\Delta x, \Delta y}(i)}{\sigma} \quad (14)$$

$$D_4 = \sum_{i=0}^{L-1} \frac{1}{1+i^2} Hgd_{\Delta x, \Delta y}(i) \quad (15)$$

$$D_5 = \sum_{i=0}^{L-1} (i - \mu)^2 Hgd_{\Delta x, \Delta y}(i) \quad (16)$$

$$D_6 = \sum_{i=0}^{L-1} i Hgd_{\Delta x, \Delta y}(i) \quad (17)$$

$$D_7 = - \sum_{i=0}^{L-1} \log_2 (Hgd_{\Delta x, \Delta y}(i)) Hgd_{\Delta x, \Delta y}(i) \quad (18)$$

### 2.4 Histogram of Sums and Differences (HSD)

The HSD, inspired in the co-occurrence matrices, was proposed by (Unser, 1986) as alternative to the co-occurrence matrices of Haralick, with the advantage of offering a low computational cost (Clausi and Zhao, 2003). The histogram of the sums  $Hs$  and histogram of differences  $Hd$  represent the amount of sums and absolute differences of pairs of image pixels, assuming a certain distance  $d$  and a direction  $\theta$ , resulting in a range of values related to the amount of gray levels  $L$  of the image. The sizes of  $Hs$  and  $Hd$  are  $2(L-1)$  and  $L-1$ , respectively and are defined as follow:

$$Hs_{\Delta x, \Delta y}(i) = \sum_{(w, h) \in D_f} \begin{cases} 1, & \text{if } f(w, h) + f(w + \Delta x, h + \Delta y) = i \\ 0, & \text{otherwise,} \end{cases} \quad (19)$$

$$Hd_{\Delta x, \Delta y}(i) = \sum_{(w, h) \in D_f} \begin{cases} 1, & \text{if } |f(w, h) - f(w + \Delta x, h + \Delta y)| = i \\ 0, & \text{otherwise,} \end{cases} \quad (20)$$

where  $\Delta x$  and  $\Delta y$  are the parameters that correspond to the distance and direction, respectively.

It should be noted that one disadvantage of using one-dimensional structures such as histograms of sums and differences is that not each of the terms extracted from the

co-occurrence matrices can be determined accurately. However, some of them such as angular second moment and entropy can be estimated (Clausi and Zhao, 2003). For this reason, we extracted from histograms of the sums and differences only 7 descriptors, as follow: angular second moment (eq.21), contrast (eq.22), correlation (eq.23), the inverse difference moment (eq.24), variance (eq.25), means (eq.26) and entropy (eq. 27).

$$D_1 = \sum_{i=0}^{2(L-1)} Hs_{\Delta x, \Delta y}(i)^2 \sum_{j=0}^{L-1} Hd_{\Delta x, \Delta y}(i)^2 \quad (21)$$

$$D_2 = \sum_{i=0}^{L-1} i^2 Hd_{\Delta x, \Delta y}(i) \quad (22)$$

$$D_7 = - \sum_{i=0}^{2(L-1)} Hs_{\Delta x, \Delta y}(i) \log_2(Hs_{\Delta x, \Delta y}(i)) - \sum_{j=0}^{L-1} Hd_{\Delta x, \Delta y}(j) \log_2(Hd_{\Delta x, \Delta y}(j)) \quad (27)$$

$$D_3 = \frac{1}{2} \sum_{i=0}^{2(L-1)} (i-2\mu)^2 Hs_{\Delta x, \Delta y}(i) - \sum_{j=0}^{L-1} j^2 Hd_{\Delta x, \Delta y}(j) \quad (23)$$

$$D_4 = \sum_{i=0}^{L-1} \frac{1}{1+i^2} Hd_{\Delta x, \Delta y}(i) \quad (24)$$

$$D_5 = \frac{1}{2} \sum_{i=0}^{2(L-1)} (i-2\mu)^2 Hs_{\Delta x, \Delta y}(i) + \sum_{j=0}^{L-1} j^2 Hd_{\Delta x, \Delta y}(j) \quad (25)$$

$$D_6 = \sum_{i=0}^{2(L-1)} Hs_{\Delta x, \Delta y}(i) \quad (26)$$

## 2.5 Gray Level Run Length Matrices (GLRLM)

The GLRLM is a method similar to the SGLD, whose goal is also to extract descriptive features of textures from auxiliary matrices, computed from the image gray level. This method takes into account the length of the primitive of gray levels. A primitive is a connected set of pixels, in a certain direction  $\theta$ , containing all the same gray level. The

length, the angle of inclination and gray level characterize these primitives.

A matrix of primitives can be defined as  $M_\theta(c, r)$ , where  $\theta$  is the angle of inclination of the primitives and each element  $(c, r)$  indicates amount that a primitive  $P$  of gray level  $c$  and length  $r$  occurs in image, that is

$$M_{\Delta x, \Delta y}(c, r) = \sum_{(w_1, h_1) \in D_f} \begin{cases} 1, & \text{if } \exists P(c, r, w_1, h_1) = ((w_1, h_1), (w_1 + \Delta x, h_1 + \Delta y), \\ & \dots, (w_{r-1} + \Delta x, h_{r-1} + \Delta y) : p \in P, f(p) = c) \\ & \text{and } P(r, c) \text{ is maximal} \\ 0, & \text{otherwise,} \end{cases} \quad (28)$$

where  $\Delta x$  and  $\Delta y$  are the parameters that correspond to direction obtained by the expressions listed in table 1, with distance  $d=L$ .

The size of the matrix is given by  $Nr \times L$ , where  $L$  is the amount of gray levels of image and  $Nr$  is the number of distinct primitive length. For this method, we consider five statistical descriptors that are extracted from the matrix of primitives. These descriptors are related to long primitive (eq. 29), short primitive (eq. 30), uniform gray level (eq. 31), primitive length uniformity (eq. 32), high gray level primitive (33), low gray level runs primitive (34) and percentage of primitives (35).

$$D_1 = \frac{1}{k} \sum_{c=0}^{L-1} \sum_{r=1}^{N_r} M_{\Delta x, \Delta y}(c, r) r^2 \quad (29)$$

Measures the amount of long primitive, where  $K$  is the number of primitives in  $M$ .

$$D_2 = \frac{1}{k} \sum_{c=0}^{L-1} \sum_{r=1}^{N_r} \frac{M_{\Delta x, \Delta y}(c, r)}{r^2} \quad (30)$$

Measures the amount of short primitive.

$$D_3 = \frac{1}{k} \sum_{c=0}^{L-1} \left( \sum_{r=1}^{N_r} M_{\Delta x, \Delta y}(c, r) r^2 \right)^2 \quad (31)$$

Measures the uniform of gray level of  $M$ .

2 S.A.Araújo, W.A.L. Alves, A.F.H. Librantz, -P. Belan

$$D_4 = \frac{1}{k} \sum_{r=1}^{N_r} \left( \sum_{c=0}^{L-1} M_{\Delta x, \Delta y}(c, r) r^2 \right)^2 \quad (32)$$

Measures the uniformity of the length of the primitive.

$$D_5 = \frac{1}{k} \sum_{c=1}^{L-1} \sum_{r=1}^{N_r} M_{\Delta x, \Delta y}(c, r) c^2 \quad (33)$$

Measures the distribution of high gray level values.

$$D_6 = \frac{1}{k} \sum_{c=1}^{L-1} \sum_{r=1}^{N_r} \frac{M_{\Delta x, \Delta y}(c, r)}{c^2} \quad (34)$$

Measures the distribution of short gray level values.

$$D_5 = \frac{k}{m} \quad (35)$$

Measures the percentage of primitives, where m is the number of distinct primitives in M.

### 3 Multi-Layer Perceptron (MLP)

Artificial neural networks (ANNs) consist of an artificial intelligence technique inspired by the structure and working of the human brain. An ANN is composed by interconnected elements, called artificial neurons, which are responsible for processing information (Haykin, 1999).

The Multi-Layer Perceptron, an ANN model widely known and used in pattern classification tasks (Sang et al., 2010; Ma et al., 2011; Moraga et al., 2011), is characterized by having, in addition to the input layer and output layer, one or more hidden layers (figure 1) that enable the network to map input patterns with similar structures for different outputs. To make this possible, the hidden layers act as feature detectors while the output layer has the function to receive the stimulus of the last hidden layer and build standard that will be the answer (Haykin, 1999).

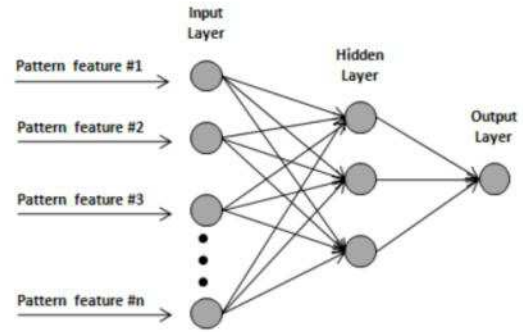


Figure 1 Example of MLP architecture.

Usually, for training MLP model, the supervised algorithm called backpropagation is used. It is a learning algorithm based on error correction in which the training is performed in two phases: first, when a pattern is presented to the network through the input layer, the activation signal is propagated layer by layer, until the response is produced by the output layer. In the second phase, the obtained output is compared to desired output for this particular pattern producing an error signal. This error signal is then propagated from the output layer to input layer and the synaptic weights are being adjusted so that the response of the network approximates the desired response (Haykin, 1999).

### 4 Materials and methods

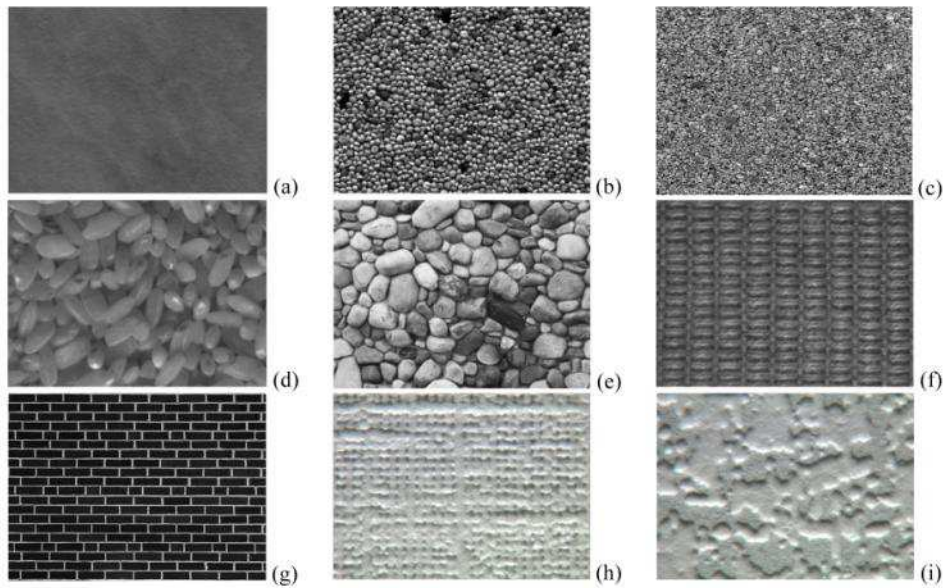
In this section it were described how the textures were gathered and how the experiments were conducted.

#### 4.1 Categorization of textures

As mentioned before there is no formal categorization of textures in the literature. Thus, we suggest five categories listed in table 2. The figure 2 shows some examples of textures for each category.

Table 2. Suggested categories of textures.

Category	Description
Smooth (SM)	Images with low roughness (figure 2a)
Uniform Granular (UG)	Images with similar grains in shapes and sizes (figures 2b and 2c)
Non-uniform Granular (NG)	Images with different grains in shapes and sizes (figures 2d and 2e)
Uniform Rough (UR)	Images with periodic rough patterns (figures 2f and 2g)
Non-uniform Rough (NR)	Images with non periodic rough patterns (figures 2h and 2i)



**Figure 2** Examples of textures of suggested categories. (a) Smooth, (b) and (c) Uniform Granular, (d) and (e) Non-uniform Granular, (f) and (g) Uniform Rough, (h) and (i) Non-uniform Rough.

#### 4.2 Experimental setup

In order to perform the experiments, it was used 100 images of metals, plastics, walls, floors, tissues, grains and rocks extracted from the three databases (Ojala et al., 2002; Dana et al. 1999; Franc and Domy, 2011), usually employed for evaluation of texture analysis algorithms. The sizes of considered images are:  $440 \times 640$ ,  $746 \times 538$  and  $1138 \times 493$ .

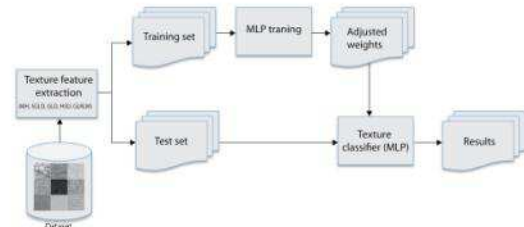
The well known Brodatz database (Brodatz, 1996) was not considered in this work once it is composed by images affected by geometric transformations such as scales and rotations, for which the evaluated statistical methods are not invariant.

From these 100 images, we compose a new database containing 3000 images of textures equally divided into five considered categories (subsets). To compose each subset, 20 images were used. Then we extract randomly 30 sub-images of size  $100 \times 100$  from each one, making a total of 600 images per category. Then,  $2/3$  of the images were used to training the ANN and the remaining  $1/3$  for test phase.

In the sequence we evaluate and compare the performance of five statistical methods for texture analysis (MH, SGLD, GLD, HSD and GLRLM) described in section 2. For this purpose it was used a MLP neural network from the popular suite of machine learning WEKA - Waikato Environment for Knowledge Analysis (Hall et al., 2009).

The experiments were comprised of the following steps: first we extracted from the image database all sets of descriptors for training and testing the ANN. For each method, there was a set for training the MLP and other for

final classification phase. For example, for the SGLD method there are two sets, with 8 columns (number of descriptors) each one while for the GLRLM method, each set has 5 columns. For each method, the MLP was trained and tested using the two sets of descriptors. A scheme of this procedure can be seen in figure 3.



**Figure 3** Proposed scheme to perform the experiments.

For evaluating each method it was used a MLP with different number of input neurons, according to the number of descriptors extracted by the statistical method.

## 5 Results and Discussion

The conducted experiments aimed to evaluate the performance of each statistical method in characterizing textures. The obtained results are summarized in table 3.

Table 3. Results of textures classification for each investigated statistical method.

Methods/Texture categories	SM	UG	NG	UR	NR	Average recall
Moments of Histogram (MH)	65.19	70.09	50.49	73.52	75.49	<b>66.96</b>
Spatial Gray Level Dependence (SGLD)	90.68	98.03	77.94	92.64	87.74	<b>89.41</b>
Gray Level Difference (GLD)	50.98	72.05	41.16	45.09	56.37	<b>53.13</b>
Histogram of sums and differences (HSD)	91.66	97.05	77.94	91.66	86.76	<b>89.01</b>
Gray Level Run Length Matrices (GLRLM)	92.15	94.60	76.96	89.21	81.86	<b>86.96</b>
<b>Average Recall</b>	<b>78.13</b>	<b>86.36</b>	<b>64.90</b>	<b>78.42</b>	<b>77.64</b>	<b>77.09</b>

Regarding the classification methods, one can infer that the SGLD method obtained the best results (89.41%), in terms of average recall, closely followed by HSD method (89.01%).

However, the GLD can be more suitable for most of applications once it can take less computational effort when compared to SGLD and GLRLM methods.

On the other hand, the worst results were obtained when the GLD and MH methods were applied in the classification. The weak hit rate of GLD could be addressed

to the low discriminatory power of the gradient information mainly in the case of images with few texture or large amount of non regular textures. It is because smooth regions have no gradient whereas regions with many small non regular textures can generate a large range of gradient values, decreasing the accuracy of the descriptors. The low efficiency of MH is probably caused by the fact that different images can have similar histograms, as illustrated in figure 4.

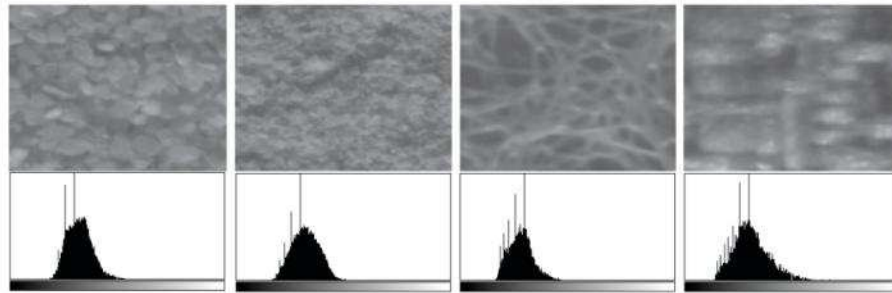
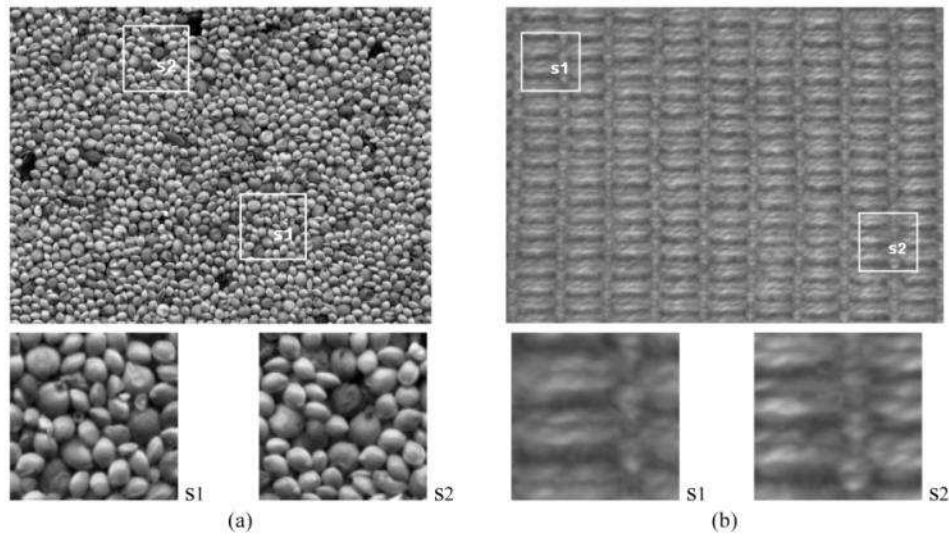


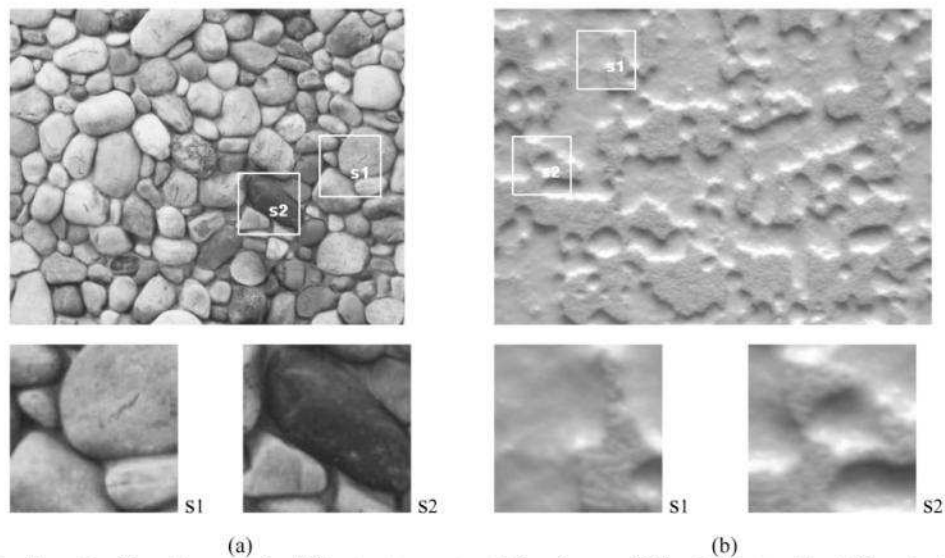
Figure 4 Examples of similar histograms of different images.

With respect to categories of textures, on average, the worst results are obtained when the classification methods were applied to non regular textures. On the other hand, best results occurred when the classification methods were applied to Uniform Granular (UG) and Uniform Rough (UR) textures, rather than other images categories. It can be

explained by the fact that two samples extracted from the same image usually have similar patterns (figures 5a and 5b) while for images from the categories NG and NR, it was observed the inverse situation, i.e. two samples extracted from the same image with very different patterns (figures 6a and 6b) is usual.



**Figure 5** Examples of samples containing similar patterns extracted from images of UG and UR categories. (a) Samples of UG image texture, (b) Samples of UR image texture.



**Figure 6** Examples of samples containing different patterns extracted from images of NG and NR categories. (a) Samples of NG image texture, (b) Samples of NR image texture.

As can be seen in figure 6, even in the case of sub-images size larger or smaller than  $100 \times 100$  the probability of occurring different patterns in them would be high. In addition, these situations become the classification a difficult task. Finally, we can also infer that for rich texture images the better descriptors could be used as similarity measures. However, this idea is not valid neither for smooth or non uniform images.

## 6 Conclusion

In this work, a performance comparison study of the main statistical methods for texture analysis was conducted. For this purpose, a neural network was used to classify textures of five proposed categories.

From the performed experiments we could not affirm that there is a more appropriate statistical method for a particular category of texture, but overall results indicates that Spatial Gray Level Dependence (SGLD) or Haralick's descriptors outperforms other evaluated methods confirming to be the one of the most efficient method for describing a wide variety of textures, in different application domains.

Nevertheless, SGLD performance was closely followed by Histogram of Sums and Differences (HSD), by indicating that this second method also should pay some attention from researchers. In summary, both statistical methods show to be appropriated for describing textures.

Regarding the categories of textures, best results occurred, on average, when the classification methods were applied to Uniform Granular (UG) and Uniform Rough (UR) images, rather than other images categories. This can be addressed to the fact that the images from these categories contain "well-behaved" texture patterns in terms of uniformity, roughness, regularity, linearity, directionality and frequency.

Despite the results obtained in this work, one can infer that in general, the performance of the statistical methods can be influenced by diverse factors such as the set of images, the parameterization of the methods and the size of the window from which texture features have been extracted. Thus, it is very important to know the descriptors for selecting a suitable subset for a determined situation.

In further works we intend to investigate more methods, including structural and spectral ones, aiming to determine a subset of optimal texture descriptors using Rough Sets Theory and Genetic Algorithms Technique, as well.

## Acknowledgement

The authors would like to thank Uninove by financial support and CAPES for the scholarship granted to one of the authors (P.A. Belan).

## References

- Broadatz, P. (1996) *A photographic Album for Artists & Designers*, Dover Publications, New York.
- Brokmeier, H. G. (2011) 'Non-destructive evaluation of strain-stress and texture in materials science by neutrons and hard X-rays', *Procedia Engineering*, Vol. 10, pp. 1657–1662.
- Chang, K. I., Bowyer, K. W. and Sivagurunath, M. (1999), 'Evaluation of Texture Segmentation Algorithms'. Paper Present at the *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 294–299.
- Chen, R., Liu, L. and Dong, G. (2010) 'Local resampling for patch-based texture synthesis in vector fields'. *International Journal of Computer Applications in Technology*, Vol. 38, No. 1/2/3, pp. 124–133.
- Chen, Y. Q., Nixon, M. S. and Thomas, D. W. (1995) 'Statistical Geometric Features for Texture Classification', *Pattern Recognition*, Vol. 28 No. 4, pp. 537–552.
- Choi, K. Y. and Kim, S. S. (2005) 'Morphological analysis and classification of types of surface corrosion damage by digital image processing', *Corrosion Science*, Vol. 47 No. 1, pp. 1–15.
- Clausi, D. and Zhao, Y. (2003) 'Grey Level Co-occurrence Integrated Algorithm (GLCIA): a Superior Computational Method to Rapidly Determine Co-occurrence Probability Texture Features', *Computers & Geosciences*, Vol. 29, pp. 837–850.
- Dana, K. J., Van-Ginneken, B., Nayar, S. K. and Koenderink, J. J. (1999) 'Reflectance and Texture of Real World Surfaces', *ACM Transactions on Graphics*, Vol. 18 No.1, pp. 1–34.
- Eisenlohr, P. and Roters F. (2008) 'Selecting a set of discrete orientations for accurate texture reconstruction', *Computational Materials Science*, Vol. 42 No. 4, pp. 670–678.
- Franc, M. and Domy, C. *Texture library*, Available at <http://textures.forrest.cz/>. (Access in 20 April 2011).
- Gadellmawla, E. S., Eladawi, A. E., Abouelatta, O. B. and Elewa, I. M. (2009) 'Application of computer vision for the prediction of cutting conditions in milling operations', *Journal of Engineering Manufacture*, Vol. 223 No. 7, pp. 791–800.
- Gonzalez, R. C. and Woods, R. E. (2002) *Digital Image Processing*, 2nd ed., Prentice Hall, New Jersey.
- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. and Witten, I. H. (2009) 'The WEKA Data Mining Software: An Update', *SIGKDD Explorations*, Vol. 11 No.1, pp. 10–18.
- Haralick, R. M., Shunmugan, K. and Dinstein L. (1973) 'Textural Features for Image Classification', *IEEE Transaction on Systems Man and Cybernetics*, SMC-3 No. 6, pp. 610–621.
- Haralick, R. M. (1979) 'Statistical and Structural Approaches to Texture', *Proceedings of The IEEE*, Vol. 67 No. 5, pp. 786–804.
- Haykin, S. (1999) *Neural networks: a comprehensive foundation*, 2nd ed., Prentice Hall.
- Ho, S. Y., Lee, K. C., Chen, S. S. and Ho, S. J. (2005) 'Accurate estimation of surface roughness from texture features of the surface image using an adaptive neuro-fuzzy inference system', *Precision Engineering*, Vol. 29 No. 1, pp. 95–100.
- Ji, S. M., Zhang, L. Q., Yuan, L., Jin, M. S. and Yuan, J. L. (2007) 'A novel ballonet polishing tool and its robot control system for polishing the curved surface of mould', *International Journal of Computer Applications in Technology*, Vol. 29, No. 2/ 3/ 4, pp. 212–215.
- Jucá, V. M. and Mello, C. A. B. (2011) 'Texture-Based Segmentation Using Image Fidelity Indexes', *IEEE Latin America Transactions*, Vol. 9 No. 3, pp. 416–420.
- Kapsalas, P., Maravelaki-Kalaitzaki, P., Zervakis, M. and Delegou, E. T. (2007) 'A. Moropoulou, Optical Inspection for Quantification of Decay on Stone Surfaces', *NDT&E International*, Vol. 40 No.1 , pp. 2–11.
- Lazebnik, S., Schmid, C. and Ponce (2005) 'A Sparse Texture Representation Using Local Affine Regions', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27 No. 8, pp. 1265–1278.
- Ma, J.W., Wang, F.J., Jia, Z.Y., Wei, W.L. (2011) 'Using support vector machine for characteristics prediction of hydraulic valve', *International Journal of Computer Applications in Technology*, Vol. 41, No.3/4, pp. 287–295.
- Mirmehdi, M., Xie, X. and Suri, J. S. (2008) *Handbook of Texture Analysis*, Imperial College Press.
- Moraga, R., Rabelo, L., Jones, A. and Vila, J. (2011) 'Using neural networks to monitor supply chain behaviour', *International Journal of Computer Applications in Technology*, Vol. 40, Nos. 1/2, pp. 53–63.
- Ohanian, P. P. and Dubes, R. C. (1992) 'Performance evaluation for four classes of textural features', *Pattern Recognition*, Vol. 25 No. 8, pp. 819–833.
- Ojala, T., Mäenpää, T., Pietikäinen, M., Viertola, J., Kyllönen, J. and Huovinen, S. (2002) 'Outex—New framework for empirical evaluation of texture analysis algorithms'. Paper Present at the *Proceedings of 16th International Conference on Pattern Recognition*, pp. 701–706.
- Qaddoura, M. A. and Belfield, K. D. (2010) 'Probing the Texture of the Calamitic Liquid Crystalline Dimer of 4-(4-Pentenyl)oxy benzoic Acid', *Materials*, Vol. 3, pp. 827–840.
- Sang, S., Liu, M., Liu, J. and An, Q. (2010) 'A new approach for texture classification in CBIR', *International Journal of Computer Applications in Technology*, Vol. 38, No. 1/2/3, pp. 34–39.

- Simoncelli, E. P. and Portilla, J. (1998), 'Texture Characterization via Joint Statistics of Wavelet Coefficient Magnitudes', Paper Present at the *Proceedings of Fifth International Conference on Image Processing*, Vol. I, pp. 4–7.
- Singh, M. and Singh, S. (2002) 'Spatial texture analysis: a comparative study' Paper Present at the *International Conference on Pattern Recognition*, pp. 676–679.
- Smith, G. and Burns, I. (1997) 'Measuring texture classification algorithms', *Pattern Recognition Letters*. Vol. 18 No. 14, pp. 1495–1501.
- Tuceryan, M. and Jain, A. K. (1998) 'Texture Analysis. in: C. H. Chen, L. F. Pau, P. S. P. Wang (eds.)', *The Handbook of Pattern Recognition and Computer Vision*, 2nd ed., World Scientific Publishing Co., pp. 207–248.
- Unser, M. (1986) 'Sum and difference histograms for texture classification', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8 No. 1, pp. 118–125.
- Varma, M. and Zisserman, A. (2004) 'Unifying statistical texture classification frameworks', *Image and Vision Computing*, Vol. 22 No. 14, pp. 1175–1182.
- Varma, M. and Zisserman, A. (2005) 'A Statistical Approach to Texture Classification from Single Images', *International Journal of Computer Vision*. Vol. 62 No. 1/ 2, pp. 61–81.
- Xie X. (2008) 'A Review of Recent Advances in Surface Defect Detection using Texture analysis Techniques', *Electronic Letters on Computer Vision and Image Analysis*, Vol. 7 No. 3, pp. 1–22.
- Yan, H., Bi, H., Li, X. and Xu, Z. (2009) 'Microstructure, texture and grain boundaries character distribution evolution of ferritic stainless steel during rolling process', *Journal of Materials Processing Technology*. Vol. 209 No. 5, pp. 2627–2631.
- Yan, H., Chen, R. S., and Han, E. H. (2011) 'A comparative study of texture and ductility of Mg–1.2Zn–0.8Gd alloy fabricated by rolling and equal channel angular extrusion', *Materials Characterization*. Vol. 62 No. 3, pp. 321–326.
- Zaidan, B. B., Zaidan, A. A., Alanazi, O. H. and Alnaqeib, R. (2010) 'Towards Corrosion Detection System', *International Journal of Computer Science Issues*. Vol. 7 No. 3, pp. 33–36.
- Zhang, J. and Tan, T. (2002) 'Brief review of invariant texture analysis methods', *Pattern Recognition*, Vol. 35 No. 3, pp. 735–747.
- Zhang, X.W., Gong, F., Xu, L.Z. (2012) 'Inspection of surface defects in copper strip using multivariate statistical approach and SVM', *International Journal of Computer Applications in Technology*, Vol. 43, No. 1, pp. 44–50.
- Zhao, H., Xu, Z. and Hong, P. (2009) 'Performance Evaluation for Three Classes of Textural Coarseness', Paper Present at the *2nd International Congress on Image and Signal Processing*, pp. 1–4.