

UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE CIÊNCIAS E TECNOLOGIA
DEPARTAMENTO DE ENG. ELÉTRICA

Relatório de Estágio

LOCAL : Laboratório de Redes Neurais e Automação Inteligente – NEUROLAB
Departamento de Sistemas e Computação – DSC
Centro de Ciências e Tecnologia – CCT
Universidade Federal da Paraíba – UFPb

ORIENTADOR : *JOSÉ HOMERO FEITOSA CAVALCANTI*

ESTAGIÁRIO : *JURACY EMANUEL MAGALHÃES DA FRANCA*
MATRÍCULA : 9221287-4
MODALIDADE DO ESTÁGIO : SUPERVISIONADO

CAMPINA GRANDE, SETEMBRO/1999



Biblioteca Setorial do CDSA. Fevereiro de 2021.

Sumé - PB

ÍNDICE

| | |
|---|-----------|
| Agradecimentos | 03 |
| Resumo | 04 |
| Introdução | 05 |
| NEUROMORFO – Detecção de Peças num Tabuleiro de Xadrez | 07 |
| NEUROMORFO – Extração de Contorno | 11 |
| Conclusão | 14 |
| Bibliografia | 15 |

AGRADECIMENTOS

Agradeço à minha família. Agradeço aos amigos que contribuíram diretamente, ou indiretamente para a realização deste trabalho, em particular, para o Professor Homero, pela orientação e incentivo. Agradeço a Deus por tudo.

RESUMO

O objetivo deste estágio foi contribuir para a inserção de visão computacional aos projetos desenvolvidos no NEUROLAB. Para este fim, foi desenvolvido um software, o NEUROMORFO, usando o compilador C++Builder, que realiza operações básicas de processamento de imagens como : cálculo de histograma, equalização, filtragem, binarização, extração de contorno.

O NEUROMORFO realiza tais operações sobre imagens armazenadas em arquivos do tipo BMP de 8 bits. Este formato foi escolhido devido à portabilidade do mesmo dentro do sistema operacional Windows, e de sua disponibilidade no software da câmara que usamos para a aquisição das imagens. Estamos utilizando para a aquisição uma câmara QuickCam B/P, que é conveniente devido ao seu baixo custo.

Para trabalhar num sistema de visão experimentalmente, desenvolvemos uma parte do software para realizar a detecção de peças, usando redes neurais artificiais, em um tabuleiro de xadrez, que gera uma matriz simbólica representando o estado da disposição das peças no tabuleiro para realimentar um software que controla um braço robótico utilizado para deslocar as peças. O sistema (braço, controle, e visão) foi denominado AutoXad.

A seguir faremos uma introdução ao propósito do laboratório, e apresentaremos o sistema AutoXad, e o NEUROMORFO. Depois descreveremos, através de estudos de caso, de forma sucinta o processamento de imagens realizado pelo NEUROMORFO.

INTRODUÇÃO

No NEUROLAB (Laboratório de Redes Neurais e Automação Industrial), são desenvolvidos robôs para pesquisas tomando como plataformas de testes a música e os jogos. Por enquanto esses robôs utilizam somente sensores óticos para detectar a posição dos eixos dos motores elétricos de corrente contínua. Obviamente, a não existência de sensores de visão limitou a operação destes robôs. Para suprir esta deficiência demos inicio ao AutoXad, que é um sistema robótico, com realimentação visual, desenvolvido para fazer a interface entre o computador e o tabuleiro do jogo de xadrez. A Fig. 1 apresenta uma foto do AutoXad, que é constituído de uma câmara QuickCam P/B (indicada pela seta), de um braço robótico, e de um microcomputador PENTIUM que executará as tarefas de visão do tabuleiro de xadrez, e as tarefas de controle do sistema.

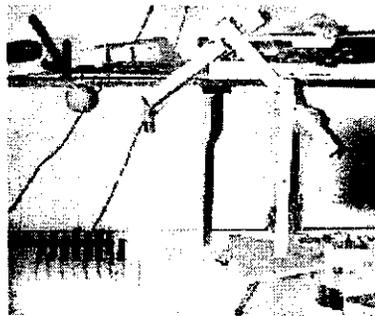


Fig.1 O AutoXad

O robô manipulador AutoXad é composto de uma base, três braços (S1,S2 e S3) e uma garra, conforme ilustrados na Fig. 3. As juntas dos braços são acionadas por motores de passo, sendo que a junta 1, responsável pelo movimento do braço S1, gira em torno da normal ao Eixo Base (ver Fig. 2), executando movimentos para a direita e esquerda. A junta 2, responsável pelo movimento do braço S2, executando movimentos para cima e para baixo e a junta 3, responsável pelo movimento do braço S3, executando movimentos também para cima e para baixo. A localização dos motores (M1,M2 e M3) é ilustrada na Fig. 3. O motor M1 movimenta o braço S1 através de um conjunto de roldanas. O motor M2 preso a base do braço S1 movimenta o braço S2 com um sistema de transmissão por cabos, e o motor M3 preso ao braço S2 movimenta o braço S3 também com um sistema de transmissão por cabos. O braço S3 possui uma garra eletromagnética (eletroímã) em sua extremidade.

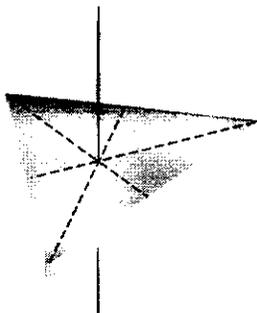


Fig.2 - O Eixo Base

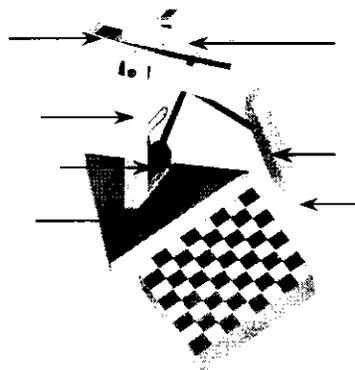


Fig.3 - O Desenho do AutoXad

A tarefa de visão do AutoXad foi implementada no NEUROMORFO, que tem como entrada arquivos do tipo BMP de 8 bits. Como ele foi desenvolvido para trabalhar com imagens capturadas por uma QuickCam, e esta possui uma matriz de 320x240 pixels, então o mesmo foi dimensionado para este formato de imagem, como pode ser visto na Fig. 4 abaixo.

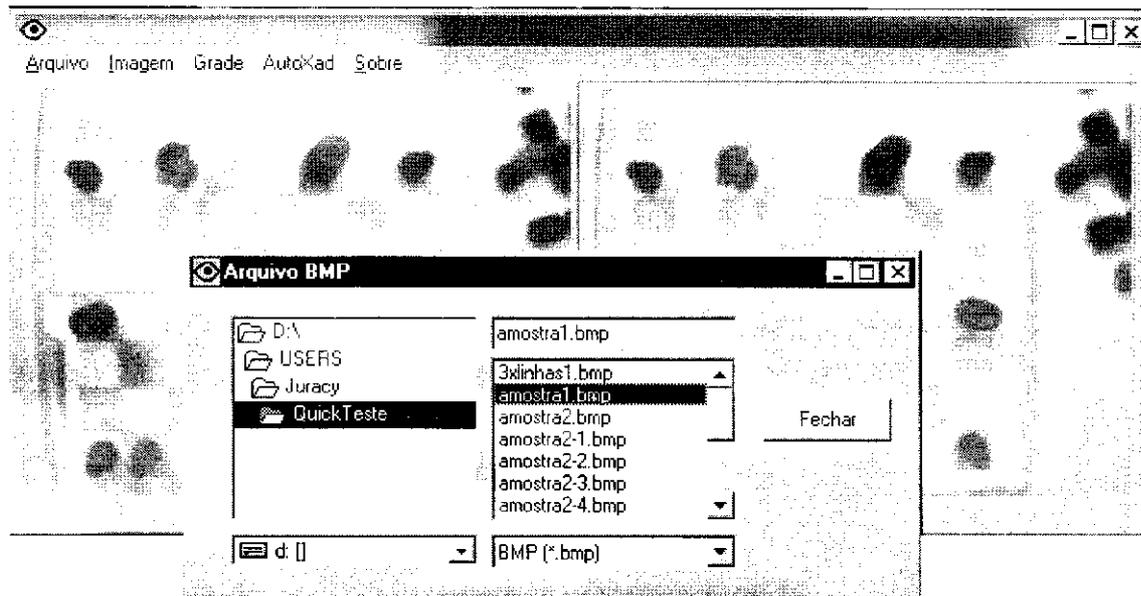


Fig. 4 Abrir Arquivo

O NEUROMORFO cria uma cópia do arquivo selecionado onde serão realizadas as operações (a esquerda contém a imagem original, e a direita a copia), embora algumas operações sejam realizadas sobre a imagem da esquerda também. Porém o NEUROMORFO salvará, no formato BMP, somente a imagem do lado direito, através da opção Salvar disponível no ítem Arquivo. O objetivo de colocar duas imagens lado-a-lado, é possibilitar a comparação do resultado do processamento com a imagem original capturada pela câmara.

Na Fig. 4 pode-se observar um Menu principal constituído por 5 opções que são : Arquivo, Imagem, Grade, AutoXad e Sobre. Ao longo do relatório não descreveremos todas as opções disponíveis no NEUROMORFO, apenas apresentaremos suas principais características através de exemplos. Em Arquivo, temos opções para abrir arquivos, salvá-los (em BMP, ou ASC), e recarregar as imagens quando for necessário. Em Imagem, temos diversas operações de PDI como Histograma, Equalizar, Binarização, Dilatação e Erosão binárias, extração de Contorno, Subtração, Ampliação, Redução, Filtros da média, Convolução de uma, ou duas, máscaras 3x3 cujos pesos são especificados pelo usuário. A opção Grade foi desenvolvida para a tarefa de detecção de peças no tabuleiro, no entanto ela pode ser utilizada para extrair pedaços específicos da imagem. A opção AutoXad é utilizada para realizar a detecção das peças no tabuleiro. Finalmente, Sobre, provê informações sobre o software.

NEUROMORFO – Detecção de Peças num Tabuleiro de Xadrez

O objetivo do NEUROMORFO é detectar as peças no tabuleiro e transferir esta informação (presença, ou ausência de peças em cada casa do tabuleiro), na forma de uma matriz simbólica, para o sistema de controle do robô. O reconhecimento das peças (pião, torre, cavalo....) não está sendo feito já que o mesmo poderá ser obtido através de comparações com as situações anteriores, deixando a interpretação do conteúdo do tabuleiro para um sistema especialista.

Para isto temos que considerar vários fatores : não foi realizado um controle sobre a iluminação, conseqüentemente há uma variação na mesma ao longo do tabuleiro; as peças são relativamente altas, conseqüentemente temos a sombra das mesmas no tabuleiro; as peças possuem cores diferentes (brancas – marrom de fato - e pretas); o tabuleiro possui duas cores; a câmara gera uma distorção do tipo barril na imagem, possui ruído, e tem apenas 6 bits de dados, ou seja 64 níveis. Considerando todos estes fatores a utilização de um processo simples de limiarização não foi possível, usamos então redes neurais para classificar os níveis de cinza, pois são robustas ao ruído, e a variações. Além disto, através de treinamento, o sistema poderá se adaptar a diversas condições de iluminação diferentes.

Nosso primeiro passo foi separar o tabuleiro do restante da imagem. Considerando que o tabuleiro uma vez fixado não terá sua posição alterada, nós encontramos sua localização, no interior da imagem, fazendo o ajuste de uma grade de 8x8 com um tamanho similar ao mesmo, até que a posição da grade seja correspondente à posição do tabuleiro. Na Fig. 5 está o formulário utilizado para ajustar a grade, e na Fig. 6 a grade sobre o tabuleiro. Para fazer o ajuste da grade é necessário equalizar a imagem de trabalho antes.

| | | | |
|-------------------|----------------|------------------|------------|
| Posição Inicial X | 57 | Imagem Inicial | |
| Y | 13 | Passo u/ Inicial | |
| Dimensão da Grade | 8 | Coloca Grade | |
| Cor da Borda | 0 | Pega Tabuleiro | |
| Tamanho da Casa | 27 | Pega Casa | |
| Linha da Casa | 0 | Volta Tabuleiro | |
| Coluna da Casa | 0 | | |
| Esquerda Ponto | Topo Ponto | Esquerda Casa | Topo Casa |
| Direita Ponto | Base Ponto | Direita Casa | Base Casa |
| Salvar Dados txt | Estado / 4 / 6 | X4 X6 | OTSU 000 |
| Posição Atual X | 28 / 2 / 3 | X2 X3 | Histograma |
| Y | 14 | | |
| Filtrar 5x5 | Filtrar 3x3 | / 3 | X9 |
| | | | Fechar |

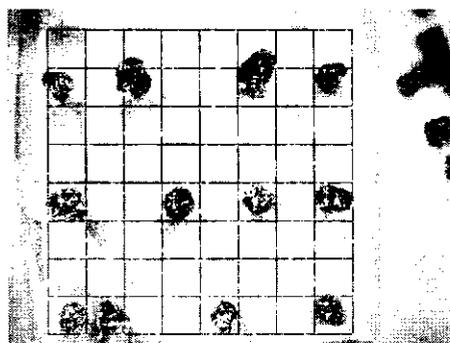


Fig. 5 Formulário para Ajuste da Grade Fig. 6 Grade Ajustada sobre o Tabuleiro

No formulário da Fig. 5 temos a posição do primeiro vértice da grade (em Posição Atual X e Y), que será utilizada para fazermos o processamento sobre o mesmo. O formulário possibilita alterar a cor do gradeado de ajuste, sua dimensão, o tamanho de cada casa, a quantidade de casas (Dimensão da Grade – sempre será um quadrado), sua posição inicial, e o seu deslocamento no interior da imagem, além de outras operações.

Tendo a coordenada X, Y (neste caso X=28, Y=14, como pode ser visto no formulário da Fig.5), podemos varrer, casa por casa, o interior do tabuleiro, aplicando as redes neurais para classificar os padrões de pixels. Mas, antes temos que preparar tais padrões. Considerando que as peças sempre estarão no centro da casa, e que, portanto, lá está a informação de interesse, nós decidimos avaliar apenas uma amostra (de 9x9 pixels) de cada casa (que tem 27x27 pixels) do tabuleiro. No entanto, 81 níveis de cinza ainda é uma quantidade grande de informação para a rede processar, além do que

certamente teremos ruído presente nos mesmos. Para resolver este problema decidimos realizar uma média aritmética sobre cada linha da amostra de 9x9 de cada casa, ficando portanto, uma amostra de 9 níveis como entrada da rede neural para uma dada casa do tabuleiro. A Fig. 7(A) mostra a região de 9x9 dentro de uma das casas do tabuleiro da Fig.6. A Fig. 7(B) mostra o conjunto de todas as amostras do tabuleiro, onde cada linha corresponde a um vetor de 9 valores, como pode ser visto nas Fig. 7 (C) e (D).

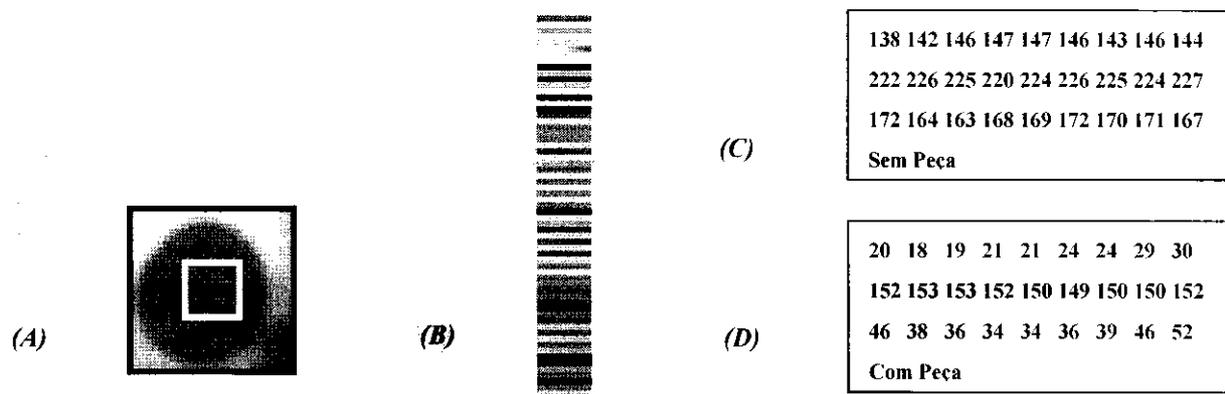


Fig. 7 (A) Região da Amostra; (B) Conjunto de Amostras do Tabuleiro; (C) Amostras das 3 Primeiras Casas do Tabuleiro; (D) Amostras das 3 Primeiras Casas da Segunda Linha do Tabuleiro.

Nós utilizamos uma rede Perceptron Multi-Camadas, com 3 camadas, tendo 9 neurônios na entrada, 8 na camada escondida, e 1 na saída. Nos escolhemos este tipo de rede porque está entre as formas mais versáteis e populares formas de redes neurais para classificação. Estas redes, com uma camada escondida simples, e a função de ativação não linear, são classificadores universais, ou seja, tais redes podem aproximar fronteiras de decisão de complexidade arbitrária.

Os 9 neurônios da camada de entrada possuem função de ativação linear, servindo apenas para transformar os valores da entrada, em partes por unidade (dividindo por 255), os neurônios da camada escondida possuem função de ativação sigmóide, dada na equação 1, e o neurônio da saída é hiperbólico, equação 2. O treinamento dos parâmetros da rede (pesos, β , θ , σ) foi realizado usando o algoritmo da Retropropagação do Erro.

$$S(X_i, \Theta, \beta, \delta) = \frac{\Theta}{[1 + \exp(-\beta(\sum X_i * Win_i + \delta))]} \quad (1)$$

$$T(X_i, \beta, \delta) = 2 * S(.) / \Theta - 1 = \frac{[1 - \exp(-\beta(\sum Y_i * Wout_i + \delta))]}{[1 + \exp(-\beta(\sum Y_i * Wout_i + \delta))]} \quad (2)$$

O conhecimento necessário para mapear os padrões de entrada (o vetor da amostra) na classe correspondente (presença - 1 - ou ausência da peça - 0) está embutido nos pesos da rede neural. Inicialmente, os pesos apropriados são desconhecidos, esta situação mudará após o treinamento, quando encontraremos um conjunto útil de pesos.

A rede irá aprender com o conjunto de treinamento apresentado à mesma (conjunto de vetores representando padrões), de forma que, se o mesmo for bom, e se o algoritmo de treinamento for adequado, através de um processo chamado generalização, a rede poderá classificar entradas que não pertençam ao conjunto de treinamento. Na

fase operacional, os pesos estarão fixos, e quando forem apresentadas as entradas de teste, ou do mundo real, a rede efetuará a classificação adequada. A escolha de um conjunto de treinamento que represente de forma adequada o problema visado, e tenha um tamanho adequado, é de fundamental importância para o bom desempenho da rede.

Pensando nisto utilizamos um conjunto de quatro imagens básicas para realizar o treinamento da rede. Para isto, inicialmente precisamos coletar as amostras de cada uma das imagens usando a opção AutoXad->RedeNeural->Amostras cujo formulário está na Fig. 8.

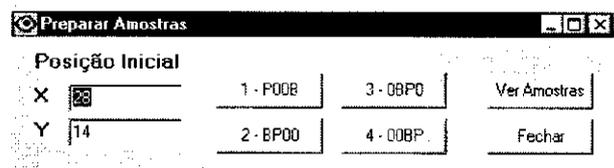


Fig. 8 Amostras para o Treinamento



Fig. 9 Conjunto de Treinamento

As variações P00B, BP00, 0BP0 e 00BP acima correspondem ao deslocamento de todas as peças, a partir da posição inicial padrão, ao longo do tabuleiro, gerando portanto, várias formações. Isto possibilitará que para cada casa do tabuleiro tenhamos padrões de treinamento correspondentes aos estados possíveis que são : presença de peça branca (B), ou preta (P), e ausência de peça (0). Finalmente, obteremos o conjunto de amostras que será utilizado para o treinamento das redes neurais, como pode ser visto na Fig. 9, cada linha, que é o conjunto de treinamento para a rede de cada casa, contém quatro vetores de 9 elementos (intercalados por um vetor de 9 elementos nulos).

Agora que temos o conjunto de amostras poderemos realizar o treinamento, para isto como foi dito acima utilizaremos o algoritmo da retropropagação do erro, que utiliza o erro quadrático médio como medida do erro na rede. Cada peso pode ser visto como uma dimensão no espaço N-dimensional do erro. No espaço do erro, os pesos são variáveis independentes e a forma da superfície de erro correspondente é determinada pela função erro em combinação com o conjunto de treinamento. O gradiente negativo da função erro em relação aos pesos, aponta para a direção na qual irá ocorrer a redução mais rápida da função erro. Movendo-se ao longo deste vetor no espaço dos pesos, será alcançado um ponto de mínimo (que poderá ser local) no qual o gradiente se igualará a zero.

No Formulário de Treinamento da Fig. 10 temos que entrar dois parâmetros : Iteração e Passo/10. Tais elementos são necessários fazer os pesos convergirem para os valores adequados, já que o processo de aprendizagem é realizado de forma iterativa.

A taxa de aprendizagem, ou o Passo, determina a distância percorrida, na direção do gradiente negativo, quando é dado um passo no espaço dos pesos. O gradiente descendente ocorrerá em passos infinitesimais ao longo da direção estabelecida pelo gradiente, mas como este procedimento não é prático utiliza-se a taxa de aprendizagem, que, tendo um tamanho adequado, possibilitará uma convergência mais rápida da rede, sem introduzir oscilações.

Finalmente, após o treinamento podemos realizar a classificação de uma imagem genérica, como aquela que está na Fig. 11 (repetição da Fig. 6 sem a grade), utilizaremos o Formulário de Classificação, reproduzido já com o resultado na Fig. 12, obtido pela opção AutoXad -> Rede Neural -> Classifica.

Treinamento

Iteração: 100 Passo /10: 5 >>

Treinar Salvar Saída

Resultados: POOB BPOO OBPO OOBP Fechar

Total: 00

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 1 | X | X | X | X | X | X | X | X |
| 2 | X | X | X | X | X | X | X | X |
| 3 | X | X | X | X | X | X | X | X |
| 4 | X | X | X | X | X | X | X | X |
| 5 | X | X | X | X | X | X | X | X |
| 6 | X | X | X | X | X | X | X | X |
| 7 | X | X | X | X | X | X | X | X |
| 8 | X | X | X | X | X | X | X | X |

Fig. 10 Formulário de Treinamento

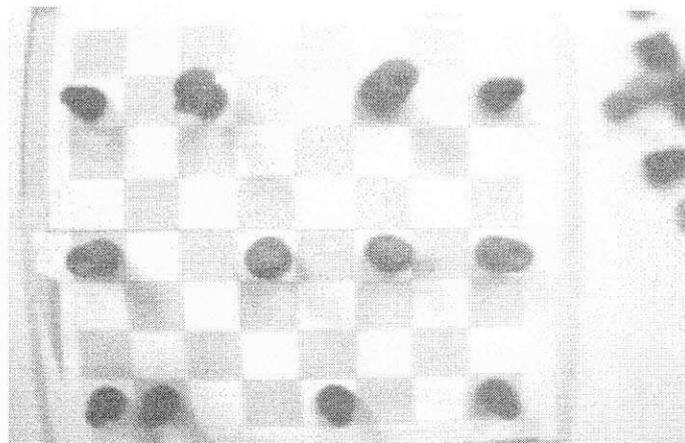


Fig. 11 Imagem de Teste

AutoXade - Neural

Posição Inicial

X: 28 Pegar Amostras Saída - Texto

Y: 14 Ler Rede

Resultado: Classificar Fechar

Total: 12

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |

Fig. 12 Resultado da Classificação

| | | | | | | | |
|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| -0.787086 | -0.793450 | -0.786588 | -0.779336 | -0.795030 | -0.782523 | -0.787864 | -0.781539 |
| 0.738623 | -0.777579 | 0.726370 | -0.786500 | -0.787293 | 0.700230 | -0.773411 | 0.744054 |
| -0.781156 | -0.793402 | -0.798148 | -0.727621 | -0.776937 | -0.801652 | -0.740820 | -0.803437 |
| -0.788560 | -0.794916 | -0.795301 | -0.770369 | -0.800606 | -0.786058 | -0.795728 | -0.793714 |
| 0.732672 | -0.800960 | -0.831056 | 0.707515 | -0.832019 | 0.756361 | -0.770219 | 0.785726 |
| -0.729758 | -0.547746 | -0.794810 | -0.720359 | -0.599317 | -0.721111 | -0.714197 | -0.779647 |
| -0.786210 | -0.855628 | -0.801919 | -0.806388 | -0.788119 | -0.816123 | -0.801428 | -0.786284 |
| 0.792787 | 0.746237 | -0.793409 | -0.773841 | 0.799145 | -0.767855 | -0.786919 | 0.751608 |

O Formulário de Classificação, Fig. 12, tem uma opção para obter o resultado numérico da classificação, como pode ser visto na tabela acima.

Finalmente, observando tanto a tabela quanto o formulário, e comparando com a imagem de teste, percebemos que o sistema consegue detectar as peças do tabuleiro.

NEUROMORFO – Extração de Contorno

A visão está para a máquina, da mesma forma que para o homem, é um recurso para interação com o ambiente que o envolve. A visão é necessária para avaliação geométrica, reconhecimento, determinação de posição e rastreamento de objetos. Tarefas fundamentais para o controle de qualidade da produção e para controle dos mecanismos usados na manufatura.

Num sistema de visão a extração do contorno dos objetos possui particular importância, pois através desta pode-se avaliar as dimensões do objeto, e até mesmo identificá-lo. Por isto, o NEUROMORFO possui várias operações úteis para este fim, e nesta seção faremos um estudo de caso de sua aplicação.

Para exemplificar algumas de suas rotinas iremos seguir um exemplo de extração de contorno de uma talhadeira, que está na Fig. 13, e compararemos o resultado obtido com a imagem original. Para economizar espaço iremos apresentar somente as imagens resultantes de cada operação.

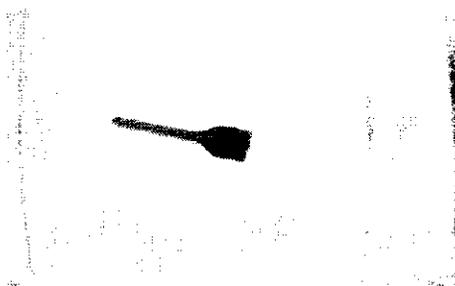


Fig. 13 Foto da Talhadeira

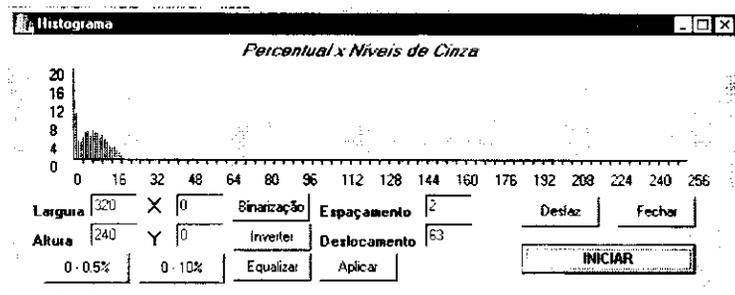


Fig. 14 Histograma da Talhadeira

A QuickCam possui apenas 6 bits para representar a imagem o que nos dá no máximo 64 níveis de cinza, no entanto para possibilitar a visualização da mesma no formato BMP de 8 bits, o software para aquisição de suas imagens gera BMPs cujas paletas não se apresentam de forma linear, ou seja, nestes arquivos o nível de cinza de valor 60 não corresponde àquele obtido numa escala de 0 a 256 correspondentes ao formato armazenado em 8 bits. Além disto há uma inversão do histograma fazendo com que o nível mais baixo (zero) corresponda à cor mais clara (branco) e o mais alto à escura (preto). Para observar o que foi dito observe na Fig. 14 o histograma da imagem acima. Ele nos dá o percentual em que cada nível (entre 0 e 256) aparece na imagem da Fig. 13.

Neste caso pode-se observar claramente que os pixels apresentam níveis menores que 64, e que a maior parte da imagem (quantidade de pixels) apresenta níveis com valores baixos (próximos do zero), no entanto observando a Fig. 13 acima, percebemos que os pixels escuros da talhadeira representam uma pequena porção da imagem, em relação aos pixels claros da bancada sobre a qual está. Isto mostra que os níveis estão invertidos, pois o zero deveria estar representando o preto.

Na foto da Fig. 13 observamos que há dois grupos de pixels : um grupo claro, que corresponde à mesa, ou seja ao fundo da imagem; e um grupo escuro, que corresponde à talhadeira, que é o objeto de interesse. Estes dois grupos deveriam estar refletidos no histograma através da presença de duas modas no histograma, no entanto devido à escala presente na Fig.14, a moda correspondente à talhadeira, que é menor, não ficou visível. Para visualizar a pequena porção de pixels correspondente à talhadeira precisamos clicar na opção 0-0.5% do formulário da Fig. 14, que restringirá a escala do percentual a 0.5% da imagem (320x240 pixels). Desta forma obtemos o histograma apresentado na Fig. 15, no qual podemos visualizar a moda correspondente à talhadeira, que está entre os níveis 28 e 36 (cada traço corresponde a 4 níveis).

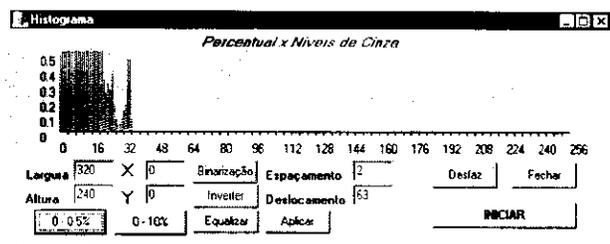


Fig. 15 Moda da Talhadeira

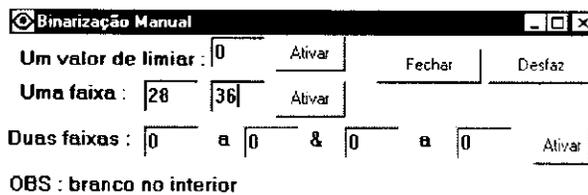


Fig. 16 Binarização Manual

O primeiro passo antes da extração do contorno é separar o objeto de interesse, a talhadeira neste caso, do restante da imagem. A este processo denominamos Binarização, que resultará em uma imagem constituída por apenas dois níveis simbólicos : 0 (nível 0 – cor preta) representando o fundo da imagem, que não nos interessa; e 1 (nível 255 – cor branca) representando o objeto de interesse. Ou seja, fazendo a binarização da imagem da Fig. 13, desejamos obter uma imagem onde a talhadeira irá aparecer totalmente branca e o restante da imagem totalmente preto. No NEUROMORFO, a binarização pode ser realizada tanto manualmente quanto automaticamente.

Na binarização manual escolhemos uma região do histograma que será convertida para a cor branca, e o resto ficará preto. Para isto, usamos o formulário que está na Fig. 16. Neste exemplo, como foi dito anteriormente, percebemos que os pixels da talhadeira possuem níveis entre 28 e 36, então selecionaremos esta faixa. O resultado pode ser visto na Fig. 17, abaixo.

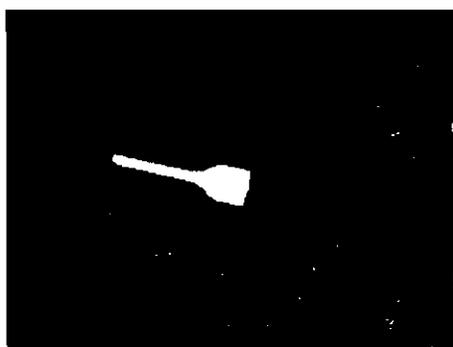


Fig. 17 Faixa de 28 a 36



Fig. 18 Limiarização de Kittler-Illingworth

No entanto, na prática é necessário que o processo seja automático, ou seja, dada a imagem, deve-se dispor de um método para extrair do histograma a faixa correspondente ao objeto. Para imagens do tipo que estamos analisando (contendo apenas duas modas – uma para o objeto, e a outra para o fundo), sendo o objeto escuro e o fundo branco, existem dois métodos estatísticos, disponíveis no NEUROMORFO, para determinar o valor de limiar que irá separar os níveis para realizar a binarização : método de Otsu, e o método de Kittler-Illingworth. Invertendo o histograma para que os níveis correspondentes aos objetos sejam os mais baixos, e aplicando o método de Kittler-Illingworth, obtivemos a imagem da Fig. 18 acima.

Finalmente, para realizar a extração do objeto, obtido pela Limiarização de Kittler-Illingworth (Fig. 18), o NEUROMORFO possui diversos algoritmos : Sobel, Prewitt, Laplaciano, e um algoritmo usando erosão binária com duas opções quanto o elemento estruturante usado. Nos utilizamos o operador Morfológico usando o elemento estruturante cruz, depois aplicamos a dilatação binária para realçar o contorno. no intuito de facilitar sua visualização e comparação com a imagem original, Fig. 19. Para possibilitar a comparação equalizamos a imagem original da Fig. 13, e extraímos a região da imagem da Fig 19 , obtendo a imagem da Fig. 20.

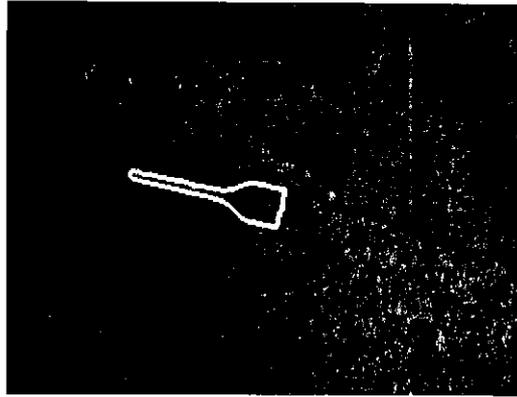


Fig. 19 Contorno Dilatado

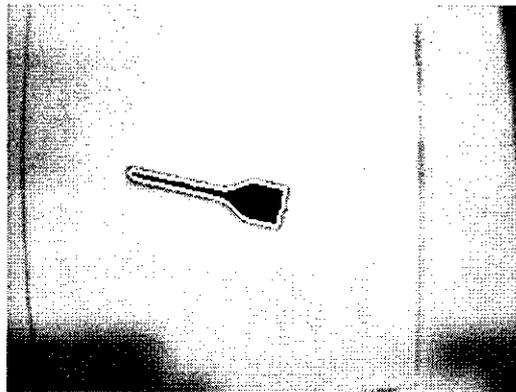


Fig. 20 Comparação com a Original Equalizada

CONCLUSÃO

Neste trabalho implementamos um software, para o sistema Windows, para realizar tarefas de visão computacional, tendo como propósito contribuir para os sistemas desenvolvidos no NEUROLAB.

Embora tenhamos realizado o objetivo especificado de detectar as peças em um tabuleiro de xadrez, o sistema ainda precisa de diversos acréscimos sendo particularmente importante a implementação de um driver para captura da imagem da QuickCam diretamente a partir da porta paralela.

Finalmente, embora a área de visão computacional seja recente, ela tem um papel fundamental em diversas aplicações na sociedade, e em particular na automação industrial, que requer inspeção e produção em altas taxas para compensar os custos. Já existe uma empresa no Brasil, a Polux, que produz vários sistemas baseados no ambiente Windows, que é útil devido à boa relação custo/desempenho. Portanto, este trabalho cumpriu com o seu papel para a minha formação profissional.

BIBLIOGRAFIA

José Homero Feitosa Cavalcanti, *Sistemas Inteligentes, Redes Neurais Artificiais e Lógica Fuzzy*, 1998, UFPb.

Jacques Facon, *Morfologia Matemática*, 1996, Editora Universitária Champagnat da PUC do Paraná.

Louis J. Galbiati Jr., *Machine Vision and Digital Image Processing Fundamentals*, 1990, Prentice Hall, Inc.

Robert M. Haralick, e Linda G. Shapiro, *Computer and Robot Vision*, 1992, Addison-Wesley Publishing Company.

Abhijit S. Pandya, e Robert B. Macy, *Patter Recognition with Neural Networks in C++*, 1995, CRC Press, e IEEE Press.