Proceedings of the VI Brazilian Conference on Neural Networks - VI Congresso Brasileiro de Redes Neurais
pp. 223–228, June 2–5, 2003 – Centro Universitário da FEI, São Paulo - SP - Brazil

Reconhecimento de Peças para Robô de Montagem utilizando Redes Neurais Artificiais - abordagem baseada em Invariância Geométrica

Mário Luiz Tronco[1], Arthur José Vieira Porto[2]

[1] Universidade Estadual Paulista - Instituto de Biociências, Letras e Ciências Exatas - Departamento de Ciências de Computação e Estatística
Rua Cristóvão Colombo, 2265 - CEP 15054 - 000 - São José do Rio Preto - SP - Brasil
[2] Universidade de São Paulo - Escola de Engenharia de São Carlos - Departamento de Engenharia Mecânica
Av. do Trabalhador São-carlense, 400 - Centro - CEP 13.566-590 - São Carlos- SP
E-mails: mariot@dcce.ibilce.unesp.br, ajvporto@sc.usp.br

Abstract

Automated Assembly Systems, based on simple and repetitive tasks of manipulation, with high level of tolerance, in controlled environment, uses artificial vision to manage tolerances and imperfect parts, and errors of positioning. This paper describes a Neural Network based Pattern Recognition System, that can recognize patterns even they are deformed by transformation of rotation, scaling and translation. Geometric Invariance is the main characteristic presented by the proposed system. Another important characteristic is the system's ability to work with noisy images. The final structure was designed to enable a parallel implementation, specially designed to run on the CPAD [3] Architecture.

1. Introdução

As operações de montagem, uma das principais aplicações de robôs, situam-se em uma área onde a automação total é rara, devido aos altos custos envolvidos, além de existirem poucos sistemas comerciais implementados com êxito. A evolução de sistemas mecânicos de montagem, os quais baseiam-se em tarefas simples e repetitivas de manipulação, com alto grau de tolerância, num meio cuidadosamente controlado, leva aos sistemas de montagem automatizada. Tais sistemas empregam visão artificial para trabalhar com problemas relacionados à tolerância das peças, peças defeituosas, além de erros de posicionamento das mesmas [6].

A utilização de Sistemas de Visão Artificial para o reconhecimento de objetos vem despertando bastante interesse em diversas áreas de aplicação. Avanços tecnológicos vêm possibilitando a implementação de algoritmos mais complexos de análise de imagens. Processadores mais rápidos e paralelos, grandes quantidades de memória disponíveis a baixo custo e desenvolvimento de técnicas de processamento e reconhecimento de imagens são os principais agentes deste processo.

A abordagem tradicional para o reconhecimento de padrões utiliza técnicas de processamento baseadas em algoritmos convencionais de processamento de imagem [7]. Com o crescente interesse em torno de Redes Neurais Artificiais, sistemas de reconhecimento de padrões baseados em modelos neurais de processamento vem sendo investigados, implementados e testados [5]. Tais sistemas, incorporados aos robôs manipuladores, resultam em sistemas inteligentes os quais podem operar em meios estruturados e não estruturados, através do uso de mecanismos avançados de realimentação sensorial, e tomando decisões usando algoritmos de aprendizado e raciocínio.

Redes Neurais vem sendo utilizadas, como mostrado por Handels [4], nas diversas etapas do processamento de imagens: *Pré-Processamento/Filtragem*, envolvendo técnicas de reconstrução, restauração e realce de imagens, *Extração de Características/Redução de Dados*, envolvendo operações de compressão de dados e extração de atributos de imagens, *Segmentação de Imagens*, envolvendo operações de partição da imagem original de acordo com algum critério, *Detecção e Reconhecimento de Objetos*, envolvendo a determinação de posição, orientação e escala de possíveis objetos em uma determinada cena, e sua classificação, *Análise de cena*, na qual são obtidos conhecimentos de alto nível (semânticos) daquilo que a imagem mostra e *Otimização*, na qual se utiliza a minimização de alguma função utilizada no processamento da imagem.

Neste artigo, descreve-se a implementação de um Sistema de Reconhecimento de Peças, baseado em Redes Neurais, para uso em Robôs empregados em operações de montagem automatizada. A principal característica apresentada pelo sistema é a de Invariância Geométrica, possibilitando ao mesmo trabalhar com imagens com transformações de translação, escala e rotação em relação às imagens previamente aprendidas, além de trabalhar com imagens ruidosas. O diagrama de blocos do sistema implementado é mostrado na figura 1, a seguir.

Após as operações de aquisição da imagem, incluindo filtragem de ruído, o padrão é Pré-Processado por um estágio que garante a invariância geométrica. O padrão resultante é então processado por um bloco gerador de *coarse coading*, no qual é implementado um esquema de compressão de dados. Tais campos (*coarse*) são classificados por uma Rede Neural Artificial, gerando um vetor representando o padrão reconhecido. As sessões a seguir trazem o detalhamento dos blocos implementados no sistema mostrado na figura 1.

2. Sistema de Reconhecimento de Peças

O Sistema de Reconhecimento de Padrões proposto foi implementado, como mostrado na figura 1, utilizando-se diversos módulos. Os módulos de Aquisição de Dados e de Segmentação/Extração de Características não serão detalhados neste artigo por envolverem apenas operações básicas nas imagens (leitura de *Frame Grabber*, Filtragem e Binarização da imagem). Exemplos de imagens resultantes destes blocos são mostrados na figura 2, a seguir.

O módulo de Reconhecimento, que será detalhado neste trabalho, foi implementado utilizando a abordagem de Normalização dos Padrões [1], em conjunto com uma técnica de redução de dados da imagem original, a qual envolve a geração de *coarse coading* [2]. Tal módulo é composto por dois sub-módulos:

- Sub-módulo Pré-Processador;
- Sub-Módulo Classificador.

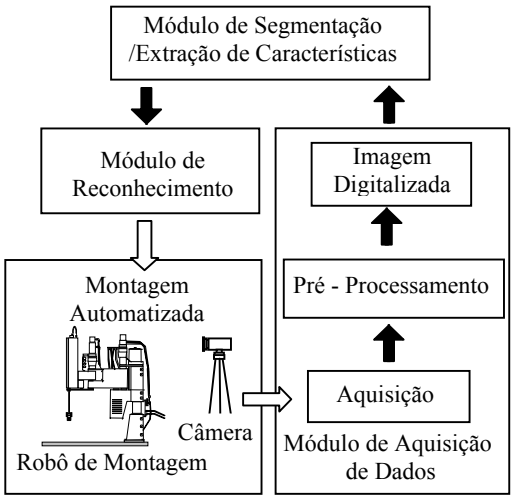

Figura 1. Sistema de Reconhecimento de Peças para Robô de Montagem.

2.1 Sub-Módulo Pré-Processador

O sub-módulo Pré-Processador tem como objetivo proporcionar correções de rotação, escala e translação antes do processo de classificação, proporcionando características de invariância geométrica ao sistema [1]. A figura 3, a seguir, mostra a estrutura do Sub-Módulo de Pré-Processamento.

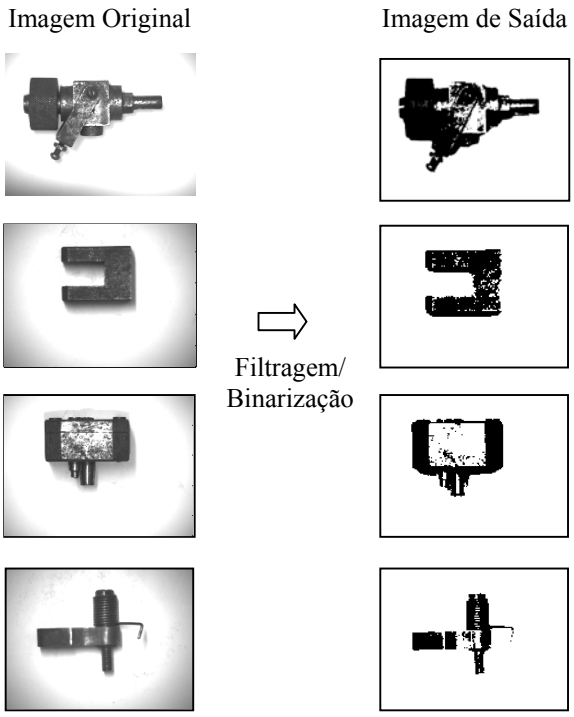

Figura 2. Imagens Resultantes dos Processos de Filtragem/Segmentação.

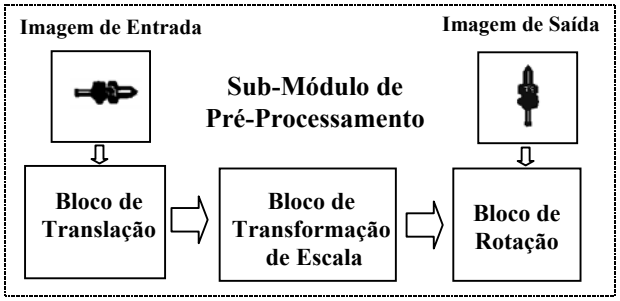

Figura 3. Sub-Módulo de Pré-Processamento

O Bloco de Rotação mantém invariância rotacional, o Bloco de Escala mantém invariância às modificações de escala e o Bloco de Translação mantém invariância translacional. A ordem na qual o blocos são cascateados é determinada pelas dependências funcionais entre os blocos, sendo o primeiro bloco o de translação, seguido pelo de escala e finalmente pelo de rotação. Como as operações na escala e de rotação necessitam de um ponto (pivô) apropriado, o Bloco de Translação é posicionado antes dos dois outros blocos. A origem da imagem de saída deste bloco será o ponto pivô para os blocos de Escala e Rotação. A seguir, é feita uma descrição dos blocos.

Para a implementação do Bloco Translação foi utilizada a seguinte formulação:

Seja P o número de pixels "1" na imagem,

$$P = \sum_{i=1}^{N} \sum_{j=1}^{N} f(x_i, y_j) \qquad (1)$$

Assim, o Centro de Gravidade (x_{av},y_{av}) será dado por:

$$x_{av} = \frac{1}{P}\sum_{i=1}^{N}\sum_{j=1}^{N} f(x_i, y_j).x_i \quad , \quad y_{av} = \frac{1}{P}\sum_{i=1}^{N}\sum_{j=1}^{N} f(x_i, y_j).y_j \tag{2}$$

onde *f(x,y)* fornece o valor do pixel para as coordenadas (x,y), ou seja, 0 ou 1.

A função de mapeamento, para que a invariância translacional possa ser obtida é dada por:

$$f_T(x_i, y_j) = f(x_i + x_{av}\ , y_j + y_{av}) \tag{3}$$

O bloco de Escala mantém a invariância (de escala) modificando a escala da imagem de entrada (proveniente do bloco de translação) de tal maneira que o raio médio para os pixels "1" seja igual a uma fração pré-determinada da grade, ou seja, qualquer que seja o tamanho da figura presente na imagem de entrada, ela será remapeada para uma imagem de saída do bloco, a qual conterá a figura da imagem de entrada com um tamanho igual a uma fração da grade. Neste trabalho, esta fração foi fixada em um quarto do tamanho da grade (128/4).

O raio de um determinado pixel é definido como o tamanho da reta que o liga à origem (ponto central da grade). Assim, o raio médio pode ser calculado como segue.

$$r_{av} = \frac{1}{\sum_{i=1}^{N}\sum_{j=1}^{N} f_T(x_i\ , y_j)} \sum_{i=1}^{N}\sum_{j=1}^{N} f_T(x_i\ , y_j).\sqrt{(x_i^2 + y_j^2)} \tag{4}$$

O fator de escala, *s*, é calculado como:

$$s = \frac{r_{av}}{R} \tag{5}$$

onde R é a fração da grade (128/4 neste trabalho).

A função de mapeamento, ou seja, aquela que produz a nova imagem, com a figura original com sua escala modificada, é mostrada a seguir.

$$f_{TS}(x_i, y_j) = f_T(s.x_i\ , s.y_j) \tag{6}$$

Através desta equação, os pixels da imagem de saída são mapeados levando em conta os pixels correspondentes na imagem de entrada (proveniente do bloco de translação). Assim, um pixel da imagem original pode gerar um conjunto de pixels na imagem de saída (no caso de aumento de tamanho da imagem original). Por outro lado, diversos pixels da imagem de entrada podem dar origem a um número menor de pixels ou mesmo um único pixel da imagem de saída (no caso de redução de tamanho da imagem original). O bloco de rotação mantém invariância rotacional rotacionando a imagem de tal maneira que a direção de máxima variância coincida com o eixo *x*. Esta transformação, baseada na transformação de Karhunen-Loéve utiliza os seguintes conceitos [1]: dado um conjunto de vetores, o *eigenvector* que corresponde ao maior *eingenvalue* da matriz covariância calculada do conjunto de vetores, aponta para a direção de máxima variância. Tal propriedade pode ser utilizada para manter invariância rotacional já que a detecção da direção de máxima variância também revela o ângulo de rotação. Utilizando-se vetores 2D formados pelas coordenadas dos pixels "1" na imagem, os *eigenvalues* podem ser calculados como segue .

Sendo:

$$\begin{bmatrix} m_x \\ m_y \end{bmatrix} = \frac{1}{\sum_{i=1}^{N}\sum_{j=1}^{N} f_{TS}(x_i, y_j)} \sum_{i=1}^{N}\sum_{j=1}^{N} f_{TS}(x_i, y_j) \begin{bmatrix} x_i \\ y_j \end{bmatrix} \tag{7}$$

$$P = \sum_{i=1}^{N} f_{TS}(x_i, y_j) \tag{8}$$

$$T_{xx} = \sum_{i=1}^{N}\sum_{j=1}^{N} f_{TS}(x_i, y_j).x_i^2 \tag{9}$$

$$T_{yy} = \sum_{i=1}^{N}\sum_{j=1}^{N} f_{TS}(x_i, y_j).y_j^2 \tag{10}$$

$$T_{xy} = \sum_{i=1}^{N}\sum_{j=1}^{N} f_{TS}(x_i, y_j).x_i\, y_j \tag{11}$$

A matriz covariância pode então, após simplificações, ser definida como:

$$C = \left(\frac{1}{\sum_{i=1}^{N}\sum_{j=1}^{N} f_{TS}(x_i, y_j)} \sum_{i=1}^{N}\sum_{j=1}^{N} f_{TS}(x_i, y_j)\, x \begin{bmatrix} x_i \\ y_j \end{bmatrix}\begin{bmatrix} x_i \\ y_j \end{bmatrix}^T\right) - \begin{bmatrix} m_x \\ m_y \end{bmatrix}\begin{bmatrix} m_x \\ m_y \end{bmatrix}^T \tag{12}$$

Como foi mantida a invariância à translação nos blocos anteriores, m_X e m_Y são zero. Após eliminar o termo médio antes da matriz, a matriz covariância torna-se:

$$C = \begin{bmatrix} T_{xx} & T_{xy} \\ T_{xy} & T_{yy} \end{bmatrix} \tag{13}$$

Em função do que foi definido anteriormente e, levando em consideração as simplificações feitas, os valores de seno e cosseno do ângulo de rotação da figura presente na imagem de entrada (imagem de saída do bloco de escala) podem ser definidos como segue.

$$\text{sen}\theta = \frac{(T_{yy} - T_{xx}) + \sqrt{((T_{yy} - T_{xx})^2 + 4.T_{xy}^2)}}{\sqrt{(8.T_{xy}^2 + 2.(T_{yy} - T_{xx})^2 + 2.(T_{yy} - T_{xx})\sqrt{((T_{yy} - T_{xx})^2 + 4T_{xy}^2)})}} \tag{14}$$

$$\cos\theta = \frac{2.T_{xy})}{\sqrt{(8.T_{xy}^2 + 2.(T_{yy} - T_{xx})^2 + 2.(T_{yy} - T_{xx})\sqrt{((T_{yy} - T_{xx})^2 + 4T_{xy}^2)})}} \tag{15}$$

Em função destes valores, de seno e cosseno, a função que faz o mapeamento da imagem de entrada para a imagem de saída, do bloco de rotação, pode ser descrita como segue.

$$f_{TSR}(x_i, y_j) = f_{TS}(\cos\theta.x_i - \text{sen}\theta.y_j\ ,\ \text{sen}\theta.x_i + \cos\theta.y_j) \tag{16}$$

A figura 4, a seguir, mostra exemplos de imagens processadas pelo Sub-Módulo de Pré-Processamento.

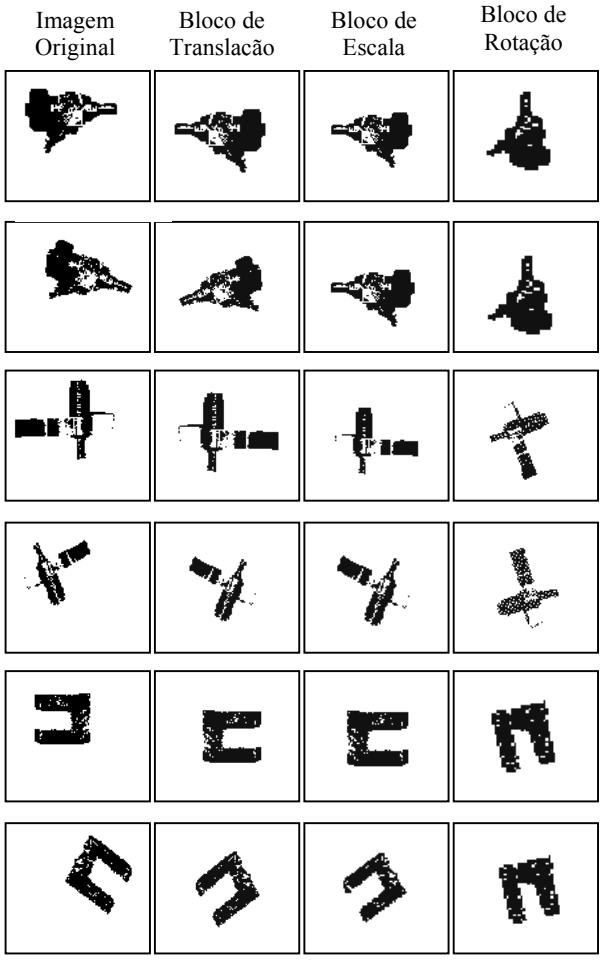

Figura 4. Imagens resultantes do Pré-Processamento.

2.2 Sub-Módulo Classificador

O Sub-Módulo Classificador, cuja função é a classificação das imagens resultantes do sub-módulo de pré-processamento, é composto por um bloco codificador (gerador de *coarse coading*) e um bloco de reconhecimento, baseado em uma Rede Neural, como mostrado na figura 5. O bloco gerador de *Coarse Coading* foi incorporado ao sistema com o objetivo de viabilizar a utilização de imagens com maiores resoluções (512 x 512, 1024 x1024, 2048x2048 pixels, etc.) as quais estão presentes em tarefas práticas de manipulação de peças reais. A técnica de *Coarse Coading* [2] é uma variação da técnica de representação distribuída, na qual cada característica é representada por um padrão de atividade sobre muitas unidades. Usando unidades que são muito grosseiramente ajustadas, poucas unidades podem codificar muitas características precisamente. O número máximo de caraterísticas é determinado pela densidade e grau de sobreposição das unidades do campo receptivo.

Codificar uma imagem com um conjunto de imagens geradas através de *Coarse Coading* incrementa significativamente o tamanho do campo de entrada possível em um sistema de classificação de imagens.

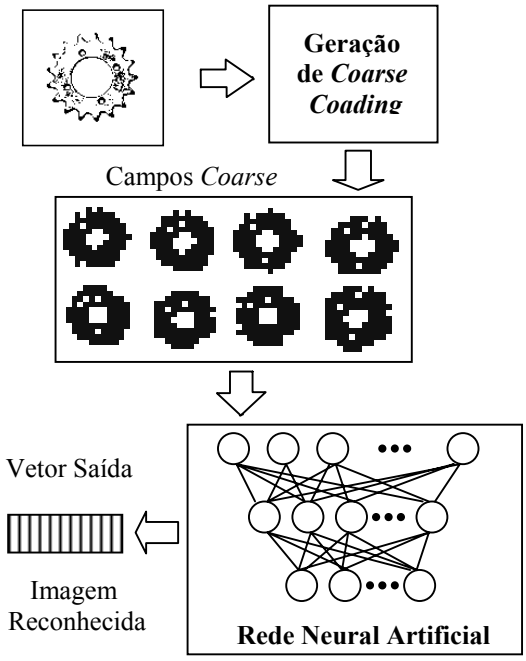

Figura 5. Sub-Módulo Classificador.

O algoritmo de *coarse coading* utiliza campos sobrepostos para representar um campo de entrada composto de pixels menores. Esta técnica funciona de forma análoga aos campos da retina, os quais permitem uma hiperacuracidade controlada. Através da sobreposição de campos de *coarse* pixels, é possível obter a imagem originalmente codificada. Usando esta técnica, a relação entre o tamanho do campo de entrada (IFS), tamanho do campo *coarse* (CFS) e o número de campos *coarse* (n) em cada dimensão é dada por [2]: IFS = (CFS * n).

No presente trabalho, como foram utilizadas imagens de 128 x 128 pixels, optou-se pela geração de 8 campos de 16 x 16 *coarse* pixels. A figura 6, a seguir, mostra exemplos da geração de *coarse coading*.

Os campos gerados através de *coarse coading*, para cada imagem, são apresentados à Rede Neural para treinamento. Foram utilizadas diversas imagens geometricamente transformadas (posição, escala e rotação) além de imagens com ruído e imagens padrão para o treinamento da Rede Neural. Assim, o funcionamento do sistema pode ser descrito como segue. Um conjunto de imagens (128 x 128 pixels) é apresentado ao sistema, para treinamento com imagens conhecidas. Cada uma destas imagens irá gerar 8 imagens de 16 x 16 pixels (campos *coarse*) as quais são ensinadas à Rede Neural. Uma vez que imagens geometricamente transformadas são pré-processadas e geram imagens que são muito próximas das imagens originais não transformadas, estas imagens (transformadas) gerarão campos *coarse* muito parecidos com os gerados pelas imagens não transformadas e já ensinados à Rede.

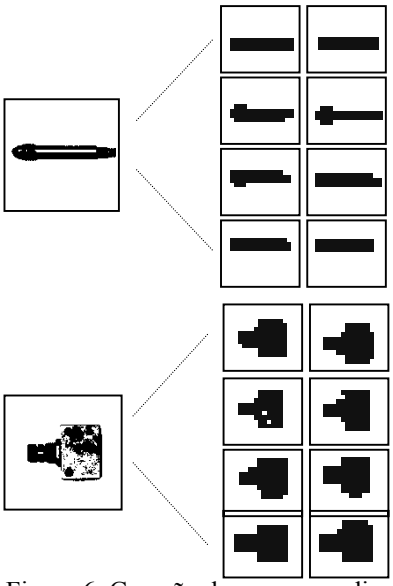

Figura 6. Geração de coarse coading.

Durante a fase de teste, sempre que uma imagem é apresentada ao sistema, o bloco de pré-processamento gera uma imagem a qual é então apresentada ao bloco de geração de *coarse coading*. Neste bloco, oito imagens são geradas e apresentadas à Rede para Reconhecimento.

Com conhecimento prévio dos campos *coarse* pertencentes às imagens de treinamento, a Rede Neural classifica a imagem.

Para teste do sistema proposto, as imagens geradas pelo gerador de *Coarse Coading* foram utilizadas para treinamento de diversos modelos de Redes Neurais, incluindo GSN [8], *Multilayer Perceptron* com *Backpropagation* e *Radial Basis Function*. Os resultados experimentais obtidos são mostrados a seguir.

3. Resultados Experimentais

Testes utilizando Modelo GSN de Rede Neural

Utilizando uma estrutura de pirâmides GSN (figura 7) foram feitos testes utilizando um conjunto de imagens para treinamento (20% do total) e imagens transformadas geometricamente para teste. Os padrões de entrada são as imagens de campos *coarse* geradas. Assim, cada padrão é representado pelos seus 8 campos *coarse* de 16x16 pixels. Cada lote testado (de um total de 100 lotes) é composto por 100 padrões de imagens (campos coarse).

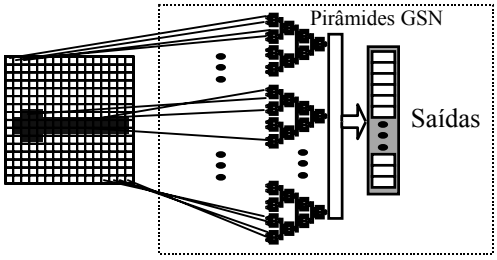

Figura 7. Configuração com GSN [8].

A figura 8 mostra os resultados experimentais obtidos.

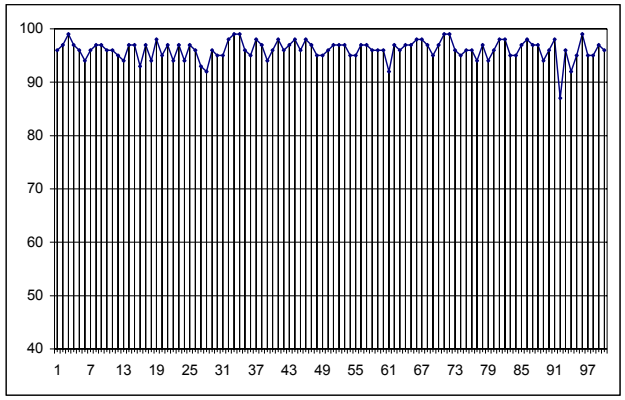

Figura 8. Taxa de Reconhecimento obtida utilizando modelo GSN.

Testes utilizando Modelo RBF de Rede Neural

Foram realizados testes utilizando RBF com algoritmo DDA, em uma estrutura empregando 256 neurônios na camada de entrada, 25 neurônios na camada intermediária e 15 na camada de saída, como mostrado esquematicamente na figura 9.

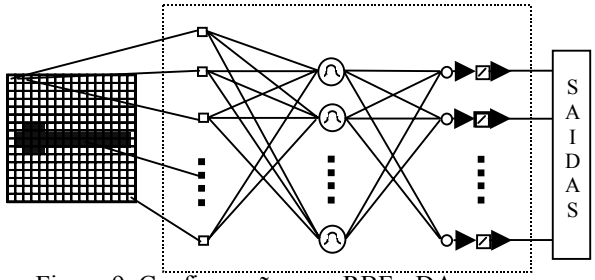

Figura 9. Configuração com RBF - DA.

Para o treinamento foram definidos 39 padrões representando 15 classes de peças. A figura 10 mostra os resultados experimentais obtidos.

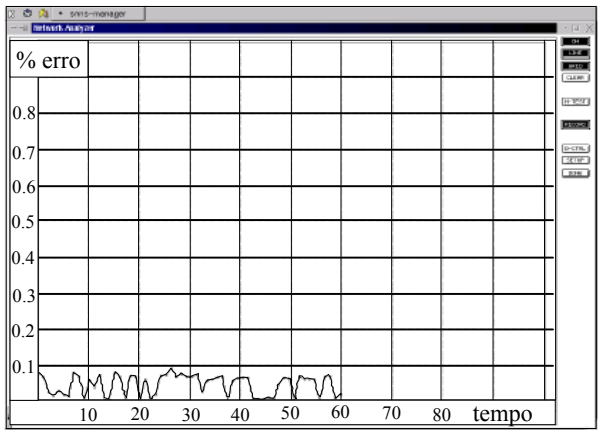

Figura 10. Erro médio obtido durante a fase de teste utilizando RBF-DA

Testes utilizando *Multilayer Perceptron* com *Backpropagation*

Foram realizados testes utilizando Backpropagation em uma estrutura empregando 256 neurônios na camada de entrada, 12 neurônios na

camada intermediária e 15 na camada de saída, como mostrado na figura 11.

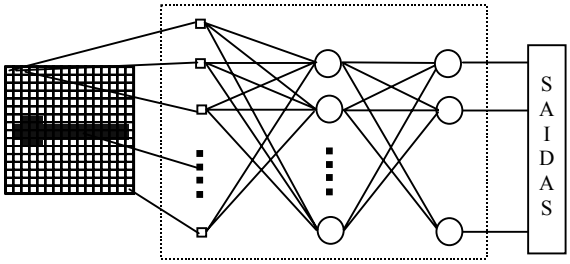

Figura 11. Configuração com Backpropagation.

A figura 12 mostra os resultados experimentais obtidos utilizando *Backpropagation*.

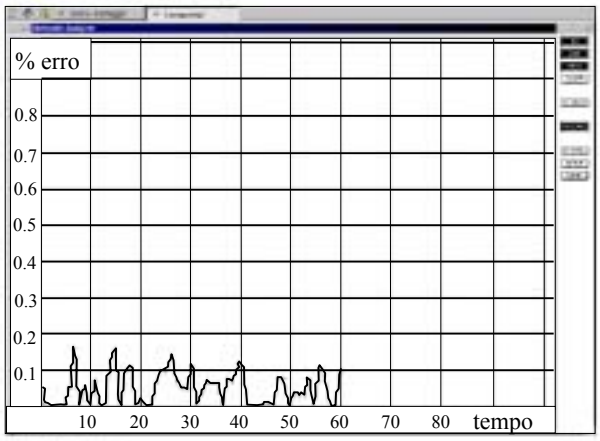

Figura 12. Erro médio obtido durante a fase de teste utilizando Backpropagation.

Tempos de Processamento

Os tempos de processamento descritos a seguir são diretamente relacionados à velocidade do processador utilizado no sistema. Os resultados foram obtidos através de um Microcomputador Pentium - 133MHz. O tempo de pré-processamento é diretamente relacionado ao número de pixels "1" presentes na imagem, e equivalente a: 9P+2G adições+5P multiplicações+P cálculos de Raiz onde: P = número de pixels "1" na imagem; G = número de pixels na imagem. Para as imagens utilizadas neste trabalho os tempos (médios) de pré-processamento obtidos foram: Bloco de translação: 17ms por padrão; Bloco de escala: 15ms por padrão; Bloco de rotação: 18ms por padrão. O tempo de geração de campos *coarse* foi de 5ms por campo. Portanto, para cada padrão o tempo de geração de campos coarse foi de 8x5ms = 40ms. O tempo de treinamento obtido para os padrões utilizados neste trabalho variou de 5 a 33 segundos, em função dos parâmetros utilizados nas configurações das Redes Neurais utilizadas. O tempo de classificação obtido, para cada padrão foi de 5ms em media.

4. Conclusões

Os tempos de processamento envolvidos demonstraram a viabilidade de utilização do sistema em aplicações de tempo real. Através das taxas de reconhecimento obtidas para os campos *coarse* gerados, pode-se garantir o reconhecimento de 100% das imagens apresentadas ao sistema, já que a classificação final é baseada no esquema *winner takes all* sobre os oito campos *coarse* testados. O sistema apresenta a característica de invariância geométrica ao tratar com as imagens de entrada, ou seja, tolera imagens de peças transladadas, escaladas e rotacionadas em relação à posição original dos exemplos utilizados para treinamento. Isto possibilita a operação em meios não controlados onde as peças a serem manipuladas podem apresentar-se ao sistema de forma diferente daquela originalmente aprendida. Para tarefas de manipulação em tempo real, esta característica representa um grande passo em relação à automação total da montagem. Imagens de entrada de 128x128, 256x256, 512x512, 1024x1024, 2048x2048 pixels, etc. podem ser utilizadas, sendo que a limitação está apenas na sensibilidade da filtragem inicial, a qual toma um maior tempo de processamento para imagens mais complexas. O esquema de codificação *coarse*, implementado no sistema, mostrou-se bastante eficaz em sua tarefa de compressão de dados, ou seja, através desta técnica, os tamanhos de imagens citados anteriormente passam a ser factíveis em sistemas de tempo real, já que estes são os tamanhos normalmente requeridos em tarefas práticas envolvendo manipulação precisa de diversas famílias de peças. O sistema pode trabalhar com diversas classes de peças, de diversos tamanhos e formas, além de poder trabalhar com imprecisões de posicionamento das mesmas.

5. Referências Bibliográficas

[1] C. Yuceer, K. Oflazer, "A rotation, scaling and translation invariant pattern classification system", *Pattern Recognition*, Vol. 26, Nr. 5, 1993.

[2] L. Spirkovska, M. B. Reid, "Coarse Coading Higher-Order Neural Networks for PSRI Object Recognition", *IEEE Transaction on Neural Networks,* Vol 4, Nr. 2, 1993.

[3] O. T. Oshiro. "Uma arquitetura paralela para o controle de máquina-ferramenta de ultra precisão. Tese de Doutorado. Escola de Engenharia de São Carlos. Universidade de São Paulo - EESC - USP, 1998.

[4] Egmont-Petersen, M.; Ridder, D.; HandelsS, H. Image Processing with neural networks - a review. Pattern Recgnition, Vol 35 (2002), pp 2279 - 2301.

[5] Pai, N.R.; Pai, S.K. A review on image segmentation techniques. Pattem Recognition 26 (9) (1993) 1277-1294.

[6] Davies**,** J. L.; Gill, K. F. Machine Vision and Automated Assembly. *Mechatronics*, (1993)Vol 3, n.4.

[7] Costa, J. A. F. *Sistema de Reconhecimento de Padrões Visuais Invariante a Transformações Geométricas Utilizando Redes Neurais Artificiais de Múltiplas Camadas.* (1996) Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Universidade de São Paulo.

[8] Carvalho, A. C. P. L. F. *Towards an integrated boolean neural network for image recognition*. (1994) PhD thesis. University of Kent at Canterbury.