
SISTEMA DE RECONHECIMENTO DE IMAGENS PARA ROBÔ DE MONTAGEM: IMPLEMENTAÇÃO E ESTUDO DE CASO

Mário Luiz Tronco[†]
mariot@dce.ibilce.unesp.br

Aleardo Manacero Júnior[†]
aleardo@dce.ibilce.unesp.br

Arthur José Vieira Porto[‡]
ajvporto@sc.usp.br

[†]Universidade Estadual Paulista - Departamento de Ciências de Computação e Estatística
Rua Cristóvão Colombo, 2265 - CEP 15054 - 000 - São José do Rio Preto - SP

[‡]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

ABSTRACT

This paper introduces a fast pattern recognition system that can be applied for parts classification on automatic assembly systems. This system is based on the use of *coarse coding* and neural networks, implemented by a low cost parallel machine. The use of *coarse coding* reduces the volume of processing needed to the recognition of parts in the assembly line, while the dedicated, low cost, parallel machine, which is built using customized processors, accelerates even more the recognition process. We present some results of parts classification using this environment, achieving always an impressive ratio of correct matchings.

KEYWORDS: Artificial Neural Networks, Robot Vision Systems, Automated Assembly Systems, Parallel Machine, Coarse Coding.

RESUMO

Este artigo apresenta um Sistema de Reconhecimento de Padrões, o qual pode ser utilizado em aplicações de classificação de peças em tarefas de montagem automatizada. O sistema, baseado em Redes Neurais, utiliza *coarse coding* e pode ser implementado em uma máquina paralela de baixo custo. O uso de codificação *coarse* reduz o volume de processamento necessário para o reconhecimento de peças em um sistema de montagem automatizada e a máquina paralela de baixo custo acelera o processo de reconhecimento. Resultados experimentais de classificação de peças, obtidos utilizando o sistema são apresentados.

PALAVRAS-CHAVE: Redes Neurais Artificiais, Visão de Robô, Montagem Automatizada, Máquina Paralela, *Coarse Coding*.

1 INTRODUÇÃO

Sistemas de Visão Artificial para o reconhecimento de objetos vem sendo empregados em diversas áreas, em especial em sistemas de montagem automatizada, utilizando robôs. Em tais sistemas, o uso de visão artificial permite contornar

problemas relacionados à tolerância e imperfeições nas peças a serem manipuladas, além de erros em seu posicionamento [6].

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 empregados em operações de montagem automatizada, resultam em sistemas que podem operar em meios estruturados e não estruturados, através do uso de mecanismos avançados de realimentação sensorial, tomando decisões usando algoritmos de aprendizado e raciocínio.

Redes Neurais vem sendo utilizadas nas diversas etapas do processamento de imagens [4]:

- *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;
- *Deteção 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;
- *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, para uso em Robôs manipuladores envolvidos em tarefas de montagem automatizada. Tal sistema, baseado em Redes Neurais, pode trabalhar com imagens com transformações de translação, escala e rotação em relação às imagens previamente aprendidas, além de trabalhar com imagens ruidosas. A descrição da estratégia de implementação paralela do sistema é também apresentada, através do uso da Arquitetura CPAD [3].

2 DESCRIÇÃO DO SISTEMA

O sistema é composto por três Módulos: Módulo de Aquisição de Dados, Módulo de Segmentação/Extração de Características e Módulo de Reconhecimento, como mostrado na figura 1, a seguir.

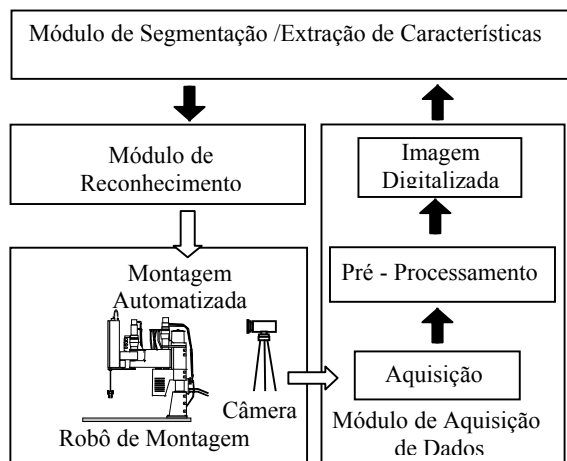


Figura 1. Sistema de Reconhecimento de Peças para Robô de Montagem.

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.

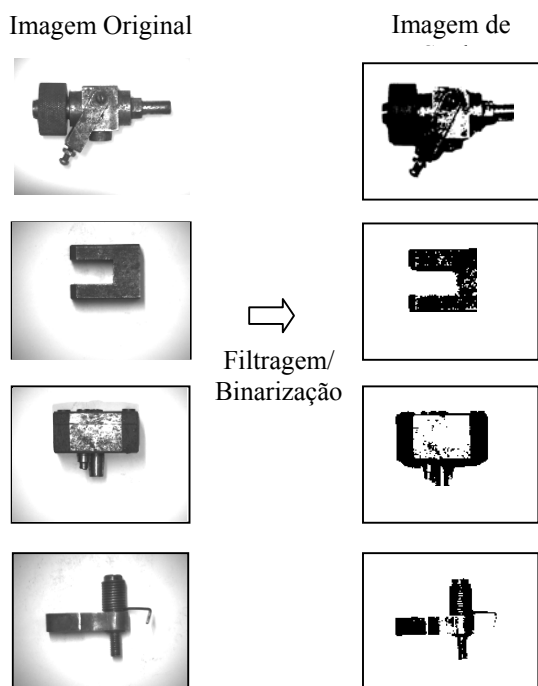


Figura 2. Imagens Resultantes dos Processos de Filtragem/Segmentação.

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 compressão de dados da imagem original, a qual envolve a geração de *coarse coding* [2]. Tal módulo é composto por dois sub-módulos:

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

A operação do sistema pode ser descrita como segue.

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 coding*, 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.

2.1 Sub-Módulo Pré-Processador

O sub-módulo Pré-Processador proporciona 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.

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 os 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

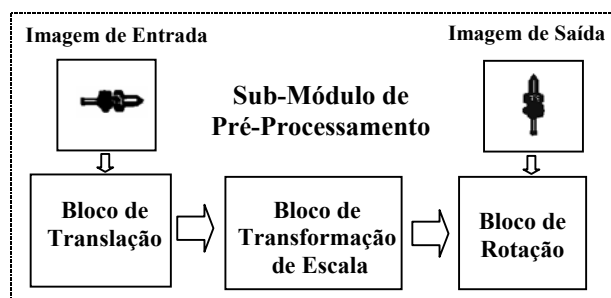


Figura 3. Sub-Módulo de Pré-Processamento

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

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

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

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

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

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:

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

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 re-mapeada 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.

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

O fator de escala, s , é calculado como:

$$s = \frac{r_{av}}{R} \quad (5)$$

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.

$$f_{TS}(x_i, y_j) = f_T(s \cdot x_i, s \cdot y_j) \quad (6)$$

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 "1" da imagem original pode gerar um conjunto de pixels "1" na imagem de saída (no caso de aumento da área representada pelos pixels "1" da imagem original). Por outro lado, diversos pixels "1" da imagem de entrada podem dar origem a um número menor ou mesmo um único pixel "1" na imagem de saída (no caso de redução da área representada pelos pixels "1" 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-Loève [11] utiliza os seguintes conceitos [1]: dado um conjunto de vetores, o autovetor que corresponde ao maior autovalor 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 autovalores podem ser calculados como segue.

Sendo:

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

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

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

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

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

A matriz covariância, definida em (12), pode ser simplificada [1].

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

Como foi mantida a invariância à translação nos blocos anteriores, m_x e m_y são zero. O termo médio da equação (12)

$$\frac{1}{\sum_{i=1}^N \sum_{j=1}^N f_{TS}(x_i, y_j)}$$

pode ser eliminado, já que não altera a direção dos autovetores. Assim, a matriz covariância torna-se:

$$C = \begin{bmatrix} T_{xx} & T_{xy} \\ T_{xy} & T_{yy} \end{bmatrix} \quad (13)$$

Levando em consideração as simplificações realizadas, 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 [11].

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

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

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.

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

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

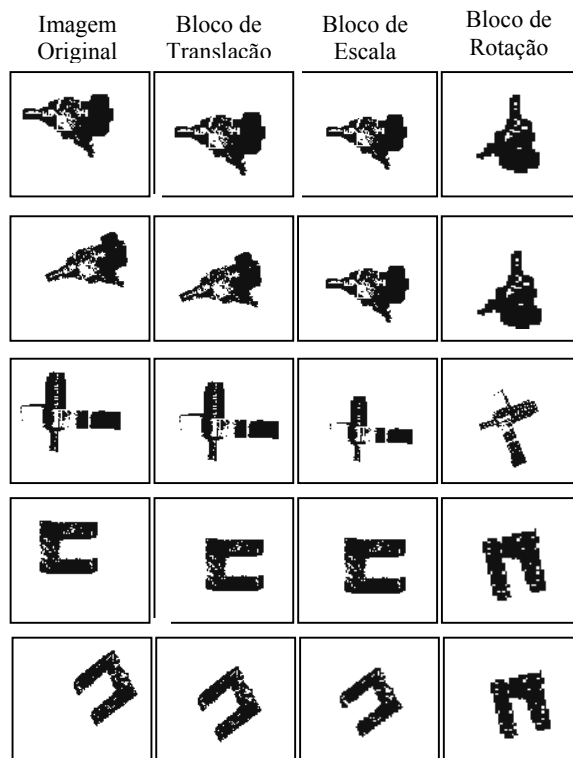


Figura 4. Imagens resultantes do Pré-Processamento.

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 coding*) e um bloco de reconhecimento, baseado em uma Rede Neural, como mostrado na figura 5.

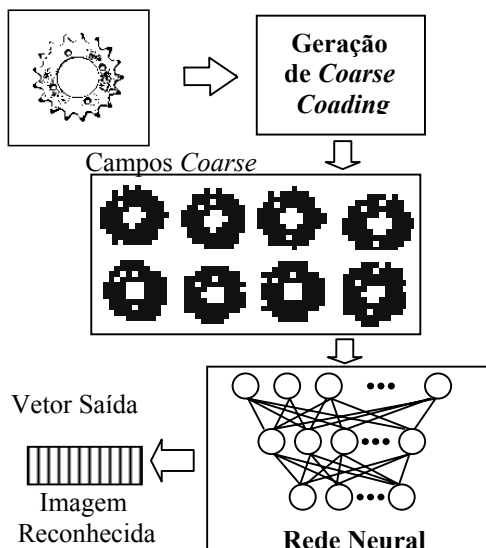


Figura 5. Sub-Módulo Classificador.

O bloco gerador de *Coarse coding* foi incorporado ao sistema com o objetivo de viabilizar a utilização de imagens com maiores resoluções (512 x 512, 1024 x 1024, 2048x2048 pixels, etc.) as quais estão presentes em tarefas práticas de manipulação de peças reais.

A técnica de *Coarse coding* [2] é uma variação da técnica de representação distribuída, na qual cada característica é representada por um padrão de atividade sobre muitas unidades. Ela utiliza campos sobrepostos para representar um campo de entrada composto de pixels menores. A figura 6, a seguir, mostra esta técnica.

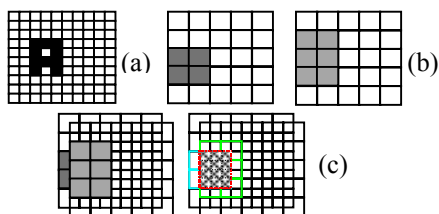


Figura 6. Técnica de *Coarse Coding*. (a) Padrão Original; (b) Campos *Coarse*; (c) Sobreposição de Campos;

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 7, a seguir, mostra exemplos da geração de *coarse coding*.

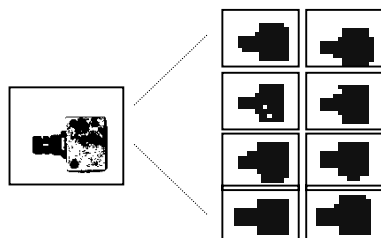


Figura 7. Geração de *coarse coding*.

Quando um objeto é apresentado ao sistema, o bloco de pré-processamento gera uma imagem a qual é apresentada ao bloco de geração de *coarse coding*. Neste bloco, oito imagens são geradas e apresentadas à Rede Neural para Reconhecimento. Com conhecimento prévio dos campos *coarse* pertencentes às imagens de treinamento, a Rede Neural classifica a imagem.

2.3 Resultados Experimentais

Para teste do sistema proposto, as imagens geradas pelo gerador de *Coarse coding* 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.

Testes utilizando Modelo GSN de Rede Neural

Utilizando uma estrutura de pirâmides GSN (figura 8) 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. 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*).

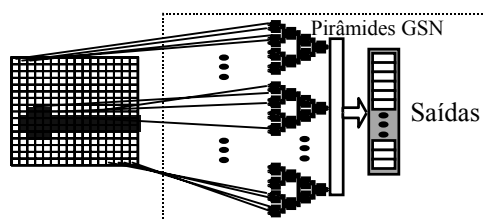


Figura 8. Configuração com GSN [8].

A figura 9 mostra os resultados experimentais obtidos

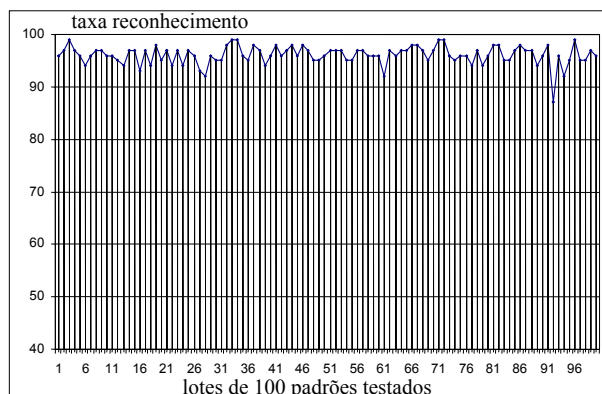


Figura 9. Taxa de Reconhecimento obtida utilizando modelo GSN.

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 10.

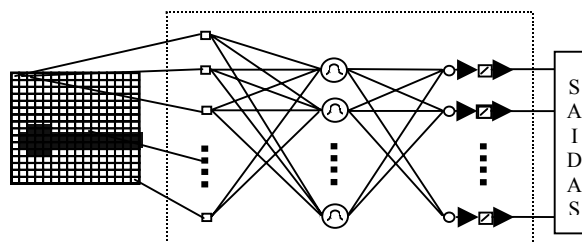


Figura 10. Configuração com RBF - DA.

Para o treinamento foram definidos 39 padrões representando 15 classes de peças. A figura 11 mostra os resultados experimentais obtidos.

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 12.

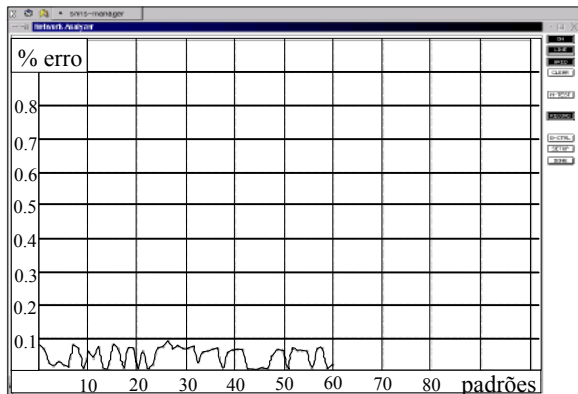


Figura 11. Erro médio obtido durante a fase de teste utilizando RBF-DA

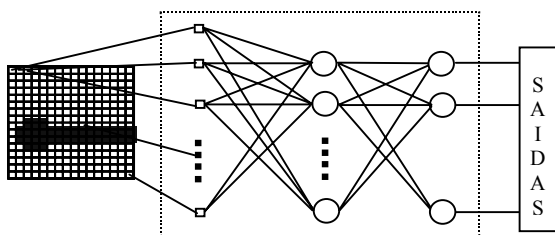


Figura 12. Configuração com Backpropagation.

A figura 13 mostra os resultados experimentais obtidos utilizando *Backpropagation*.

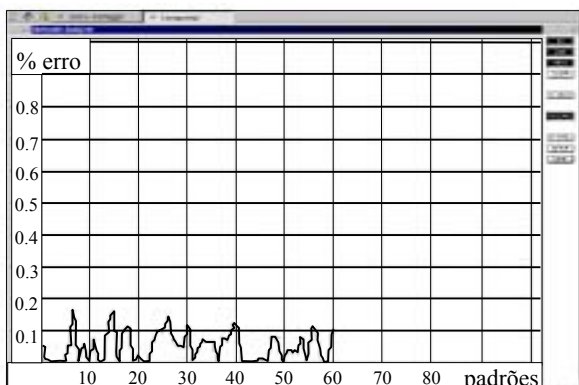


Figura 13. Erro médio obtido durante a fase de teste utilizando Backpropagation.

3 IMPLEMENTAÇÃO PARALELA

O sistema desenvolvido foi projetado para ser implementado na Arquitetura CPAD [3], a qual utiliza processadores TMS320C40. Este processador oferece bom desempenho de processamento, canais paralelos de comunicação e uma estrutura de multibus interna. Uma característica que contribui para seu alto desempenho é o

pipeline associado à concorrência de operações de entrada/saída e da CPU [9]. Uma estrutura composta por diversos 'C40 conectados em paralelo apresenta desempenho apropriado para as tarefas relacionadas às operações de montagem automatizada utilizando visão artificial. A presença de bus interno múltiplo, Coprocessador DMA, seis portas de comunicação e um conjunto de instruções que contém níveis de paralelismo possibilita a ocorrência de múltiplas operações simultâneas [10].

3.1 Arquitetura do sistema

O Módulo Básico da Arquitetura CPAD [3] é constituído por um Nó Pai e dois Nós Filhos. A hierarquia entre módulos é determinada pela ligação do nó filho de um módulo ao nó pai de outro. A figura 14, a seguir, mostra o Módulo Básico.

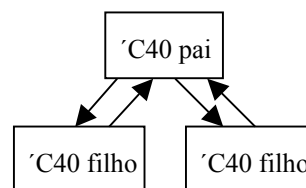


Figura 14. Módulo Básico da Arquitetura CPAD [3].

A implementação do Sistema de Reconhecimento de Imagens proposto pode ser realizada como mostrado na figura 15, abaixo.

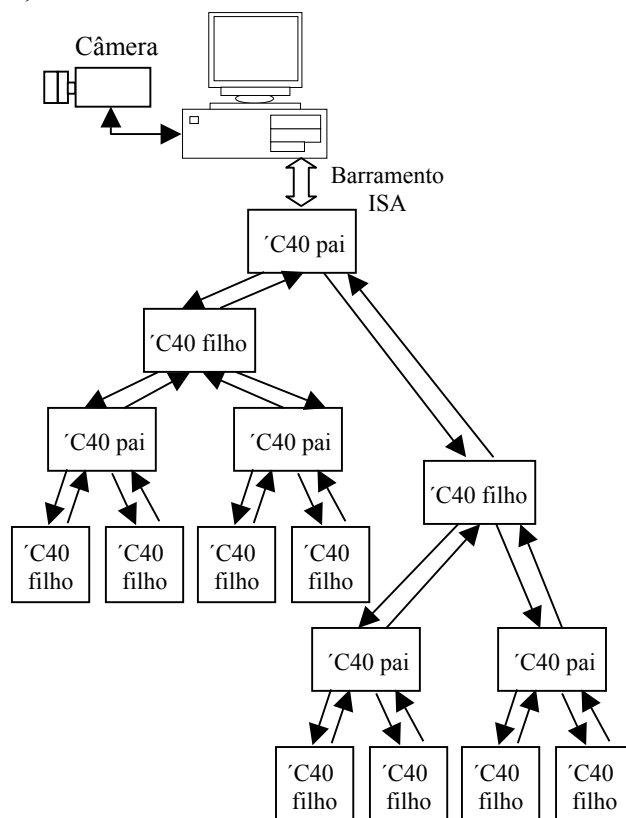


Figura 15. Sistema de Reconhecimento de Padrões utilizando a Arquitetura CPAD [3].

3.2 Paralelização do Sistema

O uso da arquitetura CPAD para a aceleração do processo de reconhecimento das imagens ocorre em dois momentos diferentes: primeiro no sub-módulo pré-processador e depois

no sub-módulo classificador. Em ambos os casos, o uso do conjunto de processadores 'C40 ocorre através de um processo de particionamento dos dados.

No sub-módulo pré-processador, o paralelismo acontece em quatro pontos do processamento, mais precisamente na determinação do centro de gravidade (equação 2), da função de mapeamento (equação 3), do raio médio (equação 4) e dos componentes da matriz covariância (equações 7 a 11). Em todos os casos o volume de processamento necessário é da ordem do número de pixels da imagem (ou N^2 , para imagens $N \times N$). Assim, mesmo considerando-se um pequeno número de operações para cada iteração (algo entre 3 operações para o cálculo da função de mapeamento e 13 operações para a matriz de covariância), tem-se um ganho acentuado de velocidade quando se particiona uma imagem de 16384 pixels (128x128), que geraria no mínimo cerca de 50000 operações aritméticas, em conjuntos de 1024 pixels para uma máquina com 16 'C40 filhos. Essa aceleração se torna ainda mais significativa para imagens mais reais, de 256x256 ou 512x512 pixels, que necessitariam executar cada ciclo respectivamente por 65536 e 262144 vezes. Como cada ciclo tem um comportamento regular e é repetido por um número de vezes que é potência de dois, assim como o número esperado de processadores filhos, então é possível acelerar um pouco mais o processamento através do desdobramento do ciclo mais interno, que executaria em cada iteração um número maior de operações, aqui definidas como oito. Assim, em vez de cada processador executar 1024 iterações (no caso de 16 'C40 e imagens de 128x128), teríamos apenas 128 iterações, reduzindo o total de instruções de máquina executadas para o controle da execução do ciclo.

Do outro lado, o paralelismo do sub-módulo classificador aparece na forma adotada para a classificação de uma imagem pela rede neural. Como cada imagem original gera oito campos coarse, o processo de reconhecimento é feito por eleição, isto é, a imagem é identificada como sendo a que mais vezes foi indicada pela rede neural a partir dos campos coarse. Assim, aqui o paralelismo usa cada processador 'C40 para realizar a classificação de um dos campos coarse. O aparente desperdício de metade dos processadores 'C40, considerando-se a arquitetura de 16 processadores até agora mencionada, pode ser facilmente compensado pelas características de concorrência dos mesmos. Desse modo, como a operação de pré-processamento é sensivelmente mais rápida, pode-se fazer com que metade dos processadores fiquem dedicados ao pré-processamento e a outra parte ao trabalho das redes neurais. Isso cria uma estrutura de pipeline em que, enquanto as redes neurais classificam uma primeira imagem usando metade dos 'C40, a outra metade dos processadores estaria fazendo o pré-processamento de uma segunda imagem, permitindo então que o sistema pudesse controlar duas (ou mais) estações de montagem automatizada.

4 CONCLUSÕES

O sistema implementado, como mostrado, pode trabalhar com imagens que sofreram alterações (transformações) em relação ao formato originalmente aprendido (ensinado à Rede Neural) e também com imagens degradadas (ruidosas). O uso de *coarse coading* proporcionou uma técnica bastante eficaz de redução de dados, permitindo ao sistema trabalhar com imagens da ordem de 256x256, 512x512 ou mesmo 1024x1024 pixels, habilitando-o à manipulação de peças de geometrias variadas com alto grau de sofisticação. Foram apresentados

resultados experimentais de reconhecimento de peças utilizando configurações diferentes de Redes Neurais, os quais mostraram uma alta taxa de acerto. Apresentou-se a Arquitetura CPAD, a qual é apropriada para a implementação paralela do sistema, e foi feito um estudo das atividades a serem paralelizadas e o potencial ganho desta estratégia. Pode-se concluir que a implementação do sistema na Arquitetura CPAD deve acelerar substancialmente tanto o pré-processamento quanto à classificação, permitindo também o controle de mais do que uma estação de montagem, separando-se os processadores em grupos dedicados a cada um dos sub-módulos. A continuação deste trabalho envolverá testes com a Arquitetura CPAD em tarefas reais de reconhecimento de padrões, em uma estação de montagem automatizada.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] C. Yuceer, K. Oflazer (1993). "A rotation, scaling and translation invariant pattern classification system", *Pattern Recognition*, Vol. 26, No. 5.
- [2] L. Spirkovska, M. B. Reid, (1993). "Coarse coding Higher-Order Neural Networks for PSRI Object Recognition", *IEEE Transaction on Neural Networks*, Vol 4, No. 2.
- [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.; Handels, H. (2002). Image Processing with neural networks - a review. *Pattern Recognition*, Vol 35, pp 2279 - 2301.
- [5] Pai, N.R.; Pai, S.K. (1993). A review on image segmentation techniques. *Pattern Recognition* 26 (9) pp 1277-1294.
- [6] Davies, J. L.; Gill, K. F. (1993). Machine Vision and Automated Assembly. *Mechatronics*, 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.
- [9] Dick, C.; Harris, F. (1998). Virtual signal processors. *Microprocessors and Microsystems*. vol 22, pp 291-292.
- [10] Harvey, D. M.; Kshirsagar, S. P.; Hobson, C. A. Low cost scaleable parallel image processing system. *Microprocessors and Microsystems*, Vol 25 (2001) 143 - 157.
- [11] Malki, H. A.; Moghaddamjoo, A. Using the Karhunen-Loève transformations in the back-propagation training algorithm. *IEEE Transactions on Neural Networks*, Vol 2(1), pp 162-165, January, 1991.