

UENF – Universidade Estadual do Norte Fluminense Darcy Ribeiro

CCT – Centro de Ciência e Tecnologia

LCMAT – Laboratório de Ciências Matemáticas

Relatório do período 10/200 à 10/2010

Bolsista: Herond Robaina Salles

Matrícula: 00107111452

Orientadora: Annabell Del Real Tamariz

Título do Plano de Trabalho: Desenvolvimento de uma interface de visão Computacional para o robô PAMDA

Fonte Financiadora da Bolsa: CNPq/UENF

1. Introdução

A utilização de artefatos explosivos em confrontos abertos, ou terrorismo, caracteriza um cenário clássico aonde a substituição do ser humano por um autônomo, capaz de detectar, interagir e desarmar tais artefatos, permitindo a preservação da vida, e é apontado visto como área de grande interesse tecnológico, suscitando um grande esforço em pesquisa. O projeto PAMDA (Protótipo Autônomo Móvel para Desarme de Artefatos) foi proposto com o intuito de criação de um autônomo capaz de interagir em um ambiente desconhecido, que utilizando um conjunto de sensores apoiado em uma rede neural, detecte e desarme um grupo de artefatos disposto aleatoriamente em uma área. Sua maior contribuição é a implementação de uma plataforma capaz de suportar as mais diversas implementações de autônomos. Para que o protótipo tenha um alto desempenho é necessário que o mesmo possua um conjunto de sensores e atuadores bem desenvolvidos, reativos e em tempo real.

Este relatório tem por objetivo descrever parte do sistema reconhecimento de padrões e visão computacional utilizado no projeto PAMDA, assim como as justificativas que levaram as escolhas de determinadas técnicas adotadas neste sistema.

2. Resumo das etapas anteriores

As etapas já desenvolvidas foram:

- Estudo e implementação de mecanismos de tratamento e simplificação de imagens, como filtros gaussianos e detectores de borda de Canny;
- Definição da plataforma do robô. No caso a plataforma SERB (*Arduino Controlled Servo Robot*) foi escolhida;
- Aquisição dos materiais para a montagem da plataforma.

3. Descrição comportamental do robô PAMDA

Antes de detalhar os algoritmos e métodos utilizados, é necessário discorrer sobre como o robô irá se comportar no ambiente, e como buscará desempenhar as tarefas da maneira mais eficiente possível, dentro das limitações técnicas impostas.

3.1. O ambiente de navegação e os objetivos

O robô PAMDA navegará sobre uma arena de MDF branca fosca, muretas laterais delimitam as bordas. Para simular declives e elevações a arena possui rampas e patamares. Bombas estão espalhadas aleatoriamente pela arena. Uma bomba é um bloco de madeira com 10x10x10cm e está sobre uma base de 5cm de elevação, cada bomba possui em suas laterais uma seqüência de fios para serem desarmados, totalizando sempre três fios, de cores verde, vermelho e preto. A ordem de desarme está bem definida no início da navegação, porém, o robô deve estar preparado para identificar outras combinações.

O robô deve desarmar todas as bombas, obedecendo à seqüência de desarme dos fios, no menor tempo possível.

3.2. Estratégia de alto nível e custos computacionais

Estando em algum lugar do ambiente, o robô não sabe a priori quais os tipos de objetos que pode encontrar, se são simples obstáculos ou bombas que devem ser desarmadas; não sabe que posição deve seguir, pois pode bater em algum objeto. Em virtude destes fatos, mecanismos de sensoriamento são utilizados para dotar o robô de um sentido simples, contudo fundamental, de examinar se em sua frente há algum obstáculo suficientemente próximo, e caso exista, o robô deve desviar-se para evitar uma colisão. O algoritmo abaixo encerra esta estratégia:

Listagem 3.2.1

```
if existir obstáculo and distância do obstáculo < D then  
    Desviar-se;
```

Contudo, percorrer o ambiente sem bater não é o propósito do robô, o objetivo maior é encontrar bombas e desarmá-las. Por isso, uma verificação antes de se desviar de um obstáculo é necessária, essa verificação deve indicar se o obstáculo é uma bomba, para dar fim ao propósito do robô. O algoritmo fica desta forma então:

Listagem 3.2.2

```
repeat
    if existir obstáculo them
        if obstáculo = bomba them
            Ir para bomba;
            Desarmar bomba;
        else distância do obstáculo < D them
            Desviar-se;
        end;
    end;

    buscar bomba;
until desarmar todas as bombas.
```

Verificar se o robô está prestes a bater em um obstáculo é computacionalmente simples, ou seja, não requer demasiado processamento. Contudo, caracterizar um obstáculo requer certo esforço computacional.

Um sonar é capaz de executar com suficiente acurácia a tarefa de buscar obstáculos, este mecanismo é acoplado a parte física do robô e pode se comunicar com ele diretamente pelo hardware. O sonar basicamente emite ultra-sons pelo ambiente, ao refletirem, são captados por um receptor, que registra a variação de tempo entre a emissão e a recepção do som. Um determinado limite de distância ativa o sistema de identificação.

Já o sistema de identificação precisa obter mais informações do obstáculo, é necessária uma foto. Esta foto deve ser trabalhada por mecanismos de reconhecimento de padrões capazes de dizer se esse obstáculo é uma bomba ou não. Em caso de resposta afirmativa, deve requisitar que o robô se dirija até a bomba para que ela seja desarmada.

Podemos dividir então o comportamento do robô em duas camadas, uma ligada diretamente ao hardware e que se destina a tarefas como locomoção, localização de obstáculos e determinadas tarefas reativas. E outra ligada a reconhecimento de padrões, que se destina a caracterizar bombas, identificar a sequência de desarme e manter alguns

estados globais do robô. A primeira camada exige menos processamento computacional, ao contrário da segunda. Por isso, é sempre preferível que os problemas sejam resolvidos preferencialmente pela primeira camada, adiando sempre que possível uma chamada a segunda.

A interação entre as duas camadas ocorre basicamente como indicado pela Figura 3.2.1:

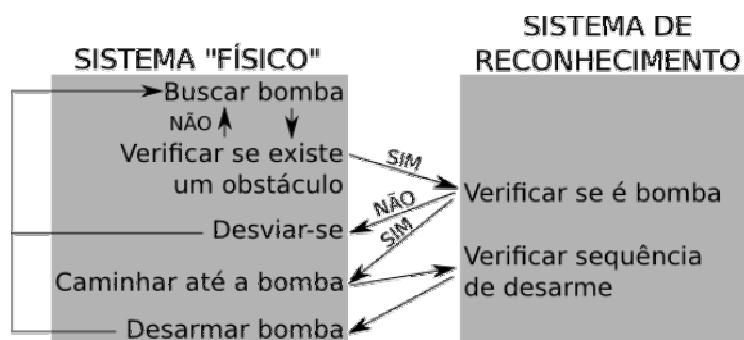


Figura 3.2.1: Interação entre o Sistema Físico e o Sistema de Reconhecimento.

O que foi descrito até aqui não encerra toda estratégia de funcionamento do robô. Como evitar que o robô tente desarmar duas vezes a mesma bomba ou fique andando em círculos, ou até mesmo como que ele vai de fato desarmar a bomba são questões que não estão explicadas neste relatório. Este tópico é um plano de fundo para a exposição do sistema de reconhecimento, o tema central deste resumo.

4. Materiais e métodos

O Sistema de reconhecimento utiliza um conjunto de descritores de imagens chamado momentos invariantes associado a uma rede neural artificial de Kohonen. Este sistema tem por finalidade categorizar as imagens do ambiente, indicando, por exemplo, se há bombas ou não.

4.1. Momentos invariantes

Momentos invariantes são um conjunto de sete descritores que caracterizam uma imagem independente de variações de rotação, translação e escala. O momento

bidimensional de ordem $(p+q)$ é dado por:

$$m_{pq} = \iint x^p y^q f(x, y) dx dy \quad (4.1.1)$$

$p, q \in$

Como uma matriz de pixels estabelece uma função discreta, já que cada pixel retorna um valor entre 0 a 255, podemos reescrever a equação (4.1.1) como:

$$m_{pq} = \sum_{x,y} x^p y^q f(x, y) \quad (4.1.2)$$

$p, q \in \mathbb{N}$

A massa total da função $f(x, y)$ é determinado pelo momento m_{00} , conforme a equação:

$$m_{00} = \sum_{x,y} f(x, y) \quad (4.1.3)$$

Existe um ponto no qual a aplicação pontual da massa total gera o mesmo momento que a massa distribuída, este ponto é dito centróide de $f(x, y)$ e suas coordenadas x e y são dadas pela equação:

$$\begin{aligned} \bar{x} &= \frac{1}{m_{00}} \sum x f(x, y) = \frac{m_{10}}{m_{00}} \\ \bar{y} &= \frac{1}{m_{00}} \sum y f(x, y) = \frac{m_{01}}{m_{00}} \end{aligned} \quad (4.1.4)$$

O momento central é obtido se deslocando a origem para o centróide, da seguinte forma:

$$\mu_{pq} = \sum_{x,y} (x - \bar{x})^p (y - \bar{y})^q f(x, y) \quad (4.1.5)$$

Ainda é necessário normalizar o momento para que os valores resultantes não sejam extremos a ponto de serem ignorados pelo sistema de reconhecimento de padrões. O momento central de ordem $(p+q)$ normalizado é obtido dividindo o momento central de mesma ordem por um fator definido por μ_{00}^γ , conforme indicado:

$$\gamma = 1 + \frac{p+q}{2}$$

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^\gamma} \quad (4.1.6)$$

A partir dessas equações são estabelecidos sete momentos invariantes à translação, rotação e escala, chamados de momentos de Hu, ou conjunto de Hu. São eles:

$$\begin{aligned} \varphi_1 &= \eta_{20} + \eta_{02} \\ \varphi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\ \varphi_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\ \varphi_4 &= (\eta_{30} + \eta_{12})^2 + (3\eta_{21} + \eta_{03})^2 \\ \varphi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ &\quad + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\ \varphi_6 &= (\eta_{20} - \eta_{02})(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 \\ &\quad + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\ \varphi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ &\quad + (3\eta_{12} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \end{aligned} \quad (4.1.7)$$

4.2. Rede neurais de Kohonen

O modelo de rede neural idealizado por Teuvo Kohonen é uma espécie de mapa auto-organizável de treinamento não supervisionado. Nesta estrutura os neurônios estão distribuídos como uma grade de geometria livre, geralmente retangular, que irá classificar os

estímulos de entrada através de um processo de aprendizado competitivo. A Figura 4.2.1 ilustra o aspecto conceitual de uma rede de Kohonen.

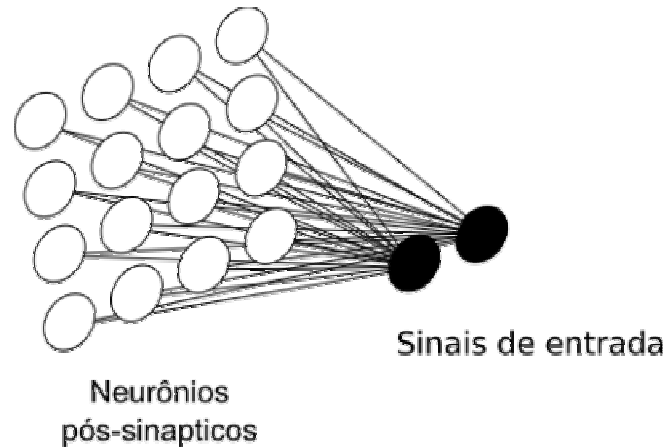


Figura 4.2.1: Estrutura de uma rede de Kohonen.

O algoritmo responsável pelo processo de organização da rede inicia os pesos sinápticos com valores pequenos, para que a rede não adquira inicialmente nenhuma organização. Em seguida ocorrem as etapas de competição, cooperação e adaptação, conforme descritos a seguir.

4.2.1. Processo competitivo

Uma entrada qualquer representada como $x = [x_1, x_2, \dots, x_n]^T$ é apresentada a rede, um neurônio deverá responder melhor a essa entrada e será dito o neurônio vencedor. O critério escolhido para indicar o neurônio vencedor é a distância euclidiana entre o vetor de entradas e o vetor de pesos do neurônio, como indicado abaixo:

$$d_i(t) = \sqrt{\sum_{j=1}^N (x_j(t) - w_{ij}(t))^2} \quad (4.2.1.1)$$

Onde:

- $d_i(t)$ é a distância euclidiana entre o vetor de pesos do neurônio i e o vetor de entradas na iteração t ;

- i é o índice do neurônio;
- j é o índice do nó de entrada;
- N é o número de entradas;
- $x_j(t)$ é o sinal de entrada nó j na iteração t ;
- $w_{ij}(t)$ é o valor do peso sináptico entre o nó de entrada j e o neurônio i na iteração t .

4.2.2. Processo cooperativo

Um neurônio vencedor estabelece o centro de uma vizinhança topológica de neurônios cooperativos. Estudos neurológicos indicam que a ativação de um neurônio biológico estimula seus vizinhos topológicos, de forma que quanto mais próximo um outro neurônio está do neurônio ativo, mais excitado pelo estímulo do neurônio ativo ele é. Em termos matemáticos, buscamos um parâmetro *vizinhança topológica* h_{ik} , que indica o grau de cooperação entre o neurônio i e seu vizinho k , que deve ser simétrico em relação ao neurônio k e deve decrescer constantemente com o aumento da distância l_{ik} , até que

$\lim_{l_{ik} \rightarrow \infty} h_{ik} = 0$. A função gaussiana (4.2.2.1) atende a esses dois requisitos:

$$h_{ik} = e^{-\left(\frac{l_{ik}^2}{2\sigma^2}\right)} \quad (4.2.2.1)$$

O parâmetro σ é denominado a largura efetiva da vizinhança e deve diminuir com o passar do tempo, indicando uma tendência de especialização. O valor σ é geralmente uma função do tipo:

$$\sigma(t) = \sigma_0 e^{-(t/\tau_l)} \quad (4.2.2.2)$$

Onde:

- σ_0 é o valor inicial de σ ;
- t é o número de iterações;

- τ_1 é uma constante de tempo.

4.2.3. Processo adaptativo

O processo de aprendizado, assim como das demais redes neurais, se dá pela atualização dos pesos sinápticos. Considerando um peso sináptico w_{ij} entre o nó de entrada j e o neurônio i , o ajuste Δw_{ij} que ele deve sofrer é:

$$\Delta w_{ij} = \eta(t) h_{ik}(t) (x_j - w_{ij}) \quad (4.2.3.1)$$

Onde o parâmetro $h_{ik}(t)$ é o índice vizinhança topológica na iteração t , referente ao neurônio melhor classificado (vencedor) k . O parâmetro taxa de aprendizagem $\eta(t)$ é definido pela expressão:

$$\begin{aligned} \eta(t) &= \eta_0 e^{-t/\tau_1} \\ \eta_0 &\in [0,1] \end{aligned} \quad (4.2.3.2)$$

Onde τ_1 é uma constante de tempo.

A taxa de aprendizagem descrita na equação decresce gradualmente, com a finalidade de evitar que novos dados comprometam seriamente o conhecimento já adquirido. Algo semelhante ocorre com o cérebro humano, quanto mais idoso menor é a capacidade de aprendizado.

4.2.4. Algoritmo geral de treinamento

Uma rede de Kohone pode ser treinando com o seguinte algoritmo:

Listagem 4.2.4.1

Definir os valores iniciais dos pesos sinápticos

Definir os valores de σ_0 , τ_1 , η_0 e o valor do erro

repeat

Inserir um exemplo de treinamento $x = [x_1, x_2, \dots, x_n]^T$

Encontrar a distância euclidiana para cada neurônio

Definir o neurônio vencedor

for neurônio da rede **do**

Calcular a largura efetiva $\sigma(t)$

Calcular a vizinhança topológica h

Calcular a taxa de aprendizagem $\eta(t)$

for arco do neurônio **do**

Calcular o valor Δw

Ajustar o arco

end

end

until distâncias euclidianas > erro

5. Resultados e discussões

Na impossibilidade de utilizar fotos do ambiente de navegação, já que esse ainda não foi confeccionado, as seguintes imagens foram utilizadas para treinar a rede e avaliar seu uso potencial:



martelo_01.jpg



martelo_02.jpg



martelo_03.jpg



martelo_04.jpg

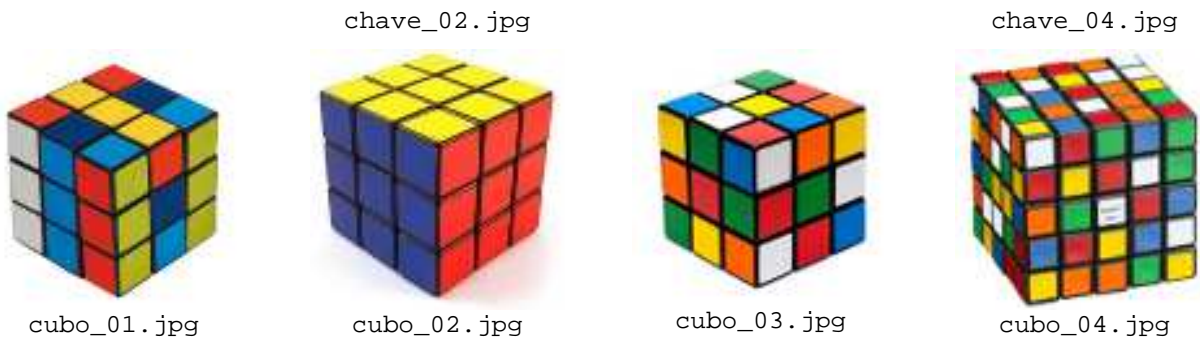


chave_01.jpg



chave_03.jpg





O conjunto de momentos invariantes obtidos para cada imagem está indicado na Tabela:

Tabela 5.1: Momentos invariantes para o conjunto de imagens de teste.

Imagem	φ_1	φ_2	φ_3	φ_4	φ_5	φ_6	φ_7
martelo_01.jpg	0.000697	0.000000	0.000001	0.000002	0.000000	0.000000	0.000000
martelo_02.jpg	0.000707	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
martelo_03.jpg	0.000740	0.000000	0.000000	0.000001	0.000000	0.000000	0.000000
martelo_04.jpg	0.000706	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
cave_01.jpg	0.000675	0.000000	0.000001	0.000000	0.000000	0.000000	0.000000
cave_02.jpg	0.000685	0.000000	0.000001	0.000004	0.000000	0.000000	0.000000
cave_03.jpg	0.000733	0.000000	0.000001	0.000040	0.000000	0.000000	0.000000
cave_04.jpg	0.000695	0.000000	0.000003	0.000043	0.000000	0.000000	0.000000
cubo_01.jpg	0.001207	0.000000	0.000009	0.000006	0.000000	0.000000	0.000000
cubo_02.jpg	0.001305	0.000000	0.000120	0.000063	0.000000	0.000000	0.000000
cubo_03.jpg	0.001207	0.000000	0.000001	0.000009	0.000000	0.000000	0.000000
cubo_04.jpg	0.001378	0.000000	0.000019	0.000062	0.000000	0.000000	0.000000

Observe que os momentos $\varphi_2, \varphi_5, \varphi_6$ e φ_7 são irrelevantes, pois sendo iguais para todas as imagens, não ajudam a caracterizar nenhuma delas, desta forma, o conjunto de treinamento $T = \{\varphi_1, \varphi_3, \varphi_4\}$ foi passado para uma rede de mapa retangular com 12x12 neurônios, $\sigma_0 = 0,4$, $\eta_0 = 0,2$ e $\tau_0 = 0,2$. Após 150 épocas de treinamento a rede distribuiu as imagens da seguinte maneira:

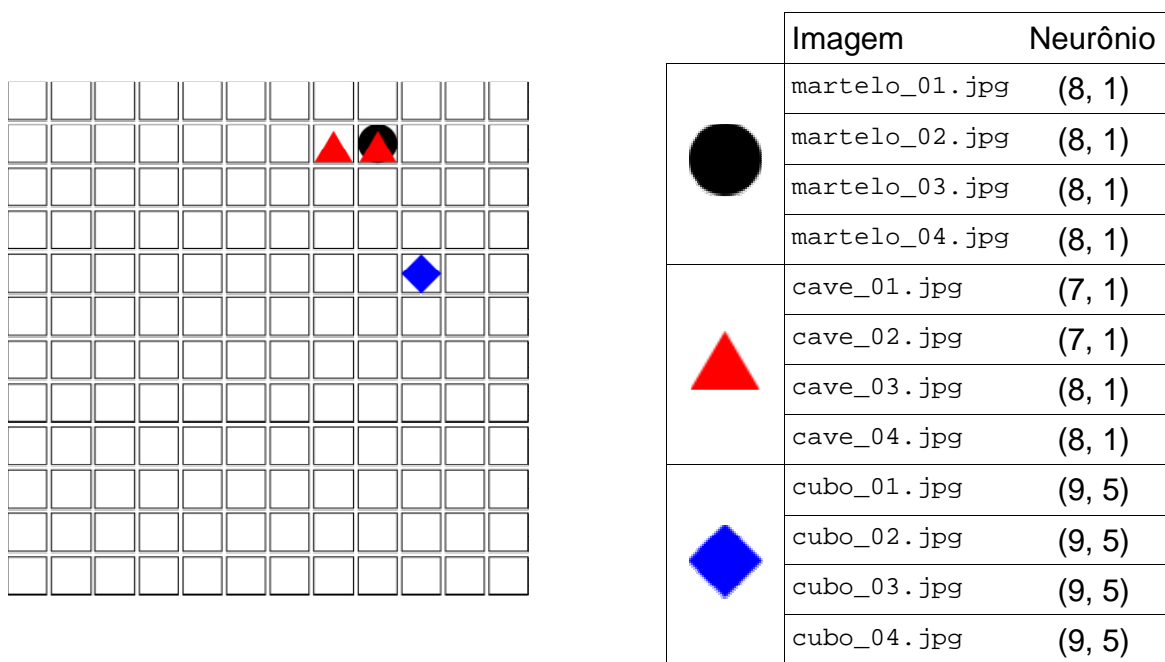


Imagem 5.1: Distribuição do conjunto de treinamento no mapa da rede.

De acordo com a Imagem 5.1, todas as imagens de cubos foram identificadas como pertencentes da mesma categoria, tendo sido reconhecidas pelo mesmo neurônio (neurônio (9, 5)), porém, as imagens de martelos e chave philips foram agrupadas numa única categoria, o que não surpreende, pois esses dois objetos tem silhuetas muito parecidas, ambos em forma de bastão.

A taxa de erro em cada época se comporta como indicado pelo gráfico:

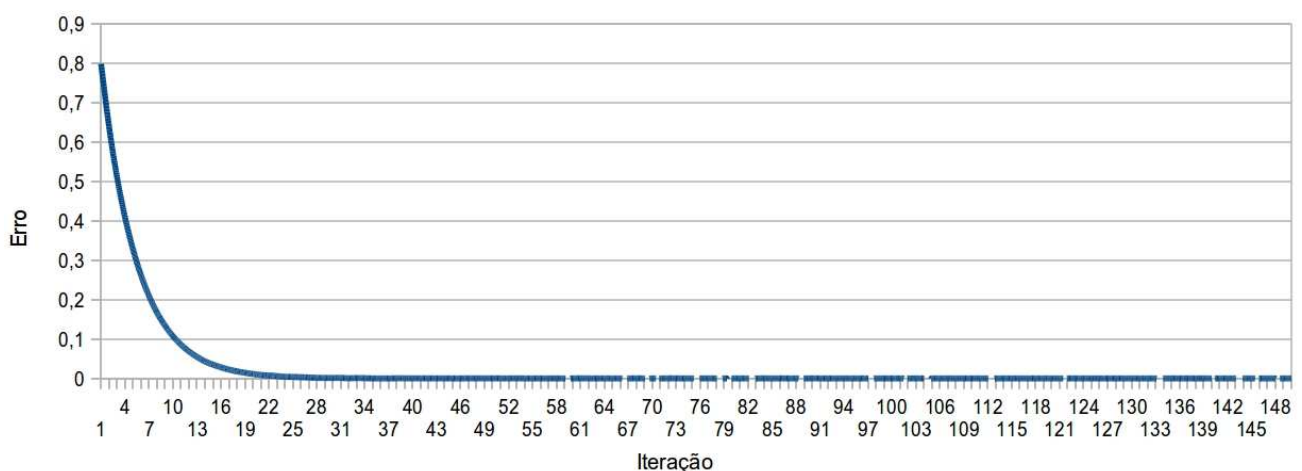


Imagem 5. 2: Variação do erro ao longo das épocas de treinamento.

6. Conclusões

A utilização da rede neural de Kohonen associada aos momentos invariantes mostrou-se uma ferramenta adequada para categorizar imagens complexas, o que permite supor ser a técnica apropriada para compor o mecanismo de identificação de padrões do robô PAMDA. Contudo, uma confirmação definitiva será possível quando o treinamento da rede ocorrer com imagens tiradas do próprio ambiente que o robô estará inserido.

7. Referências

- [1] M. K. Hu, *Visual Pattern Recognition by Moment Invariants*, IRE Trans. Info. Theory, vol. IT-8, pp.179–187, 1962.
- [2] LUDWIG JÚNIOR, O. ; MONTGOMERY, E. . *Redes Neurais: Fundamentos e Aplicações com Programas em C*. 1. ed. Rio de Janeiro: Editora Ciência Moderna Ltda, 2007. v. 1. 136 p.

8. Perspectivas de continuidade e desdobramento do trabalho

É necessário que a rede de Kohonen seja treinada com exemplos de acordo com o objetivo do reconhecimento e, tendo sido confirmado sua eficácia, é preciso incorporar o sistema de reconhecimento na plataforma do robô, para isso serão necessários uma câmera e um sistema de software que possa gerenciar um repositório de imagens para alimentar os algoritmos da rede. Também é necessário à criação de um sistema de meta-dado vinculada ao sistema de movimento do robô que relacione as imagens com a posição e a direção em que foi tirada, para evitar que um mesmo cenário seja fotografado mais de uma vez.

A detecção da ordem dos fios deve ser implementada para que o robô possa desarmar os fios das bombas na ordem correta.

Deve ser pensado também como irá ocorrer a comunicação entre o sistema de reconhecimento e o sistema físico, sempre na perspectiva de otimizar o custo computacional.

9. Participação em congressos e trabalhos publicados ou submetidos

9.1. Participação em eventos

[1] Congresso Fluminense de Iniciação Científica e Tecnológica. Quantização cromática utilizando o conjunto de cores mais contrastantes. 2010. (Congresso).

[2] Maré de agilidade. 2010. (Encontro).

9.2. Resumos expandidos publicados em anais de congressos

[1] SALLES, H. R. ; CARVALHO, R. M. ; NETO, F. A. F. ; TAMARIZ, A. R. . Quantização cromática utilizando o conjunto de cores mais contrastantes. In: Congresso Fluminense de Iniciação Científica e Tecnológica, 2010, Campos dos Goytacazes. Congresso Fluminense de Iniciação Científica e Tecnológica, 2010.

9.3. Apresentações de Trabalho

[1] SALLES, H. R. ; CARVALHO, R. M. ; NETO, F. A. F. ; TAMARIZ, A. R. . Quantização cromática utilizando o conjunto de cores mais contrastantes. 2010. (Apresentação de Trabalho/Congresso).

9.4. Outras atividades

[1] Monitor voluntário da disciplina de Engenharia de Software apoiada por computador: laboratório de computação.