
APLICAÇÃO DE REDES NEURAIS DE HOPFIELD PARA DETECÇÃO E ESTIMAÇÃO DE MOVIMENTO ENTRE DOIS QUADROS DE UM SINAL DE VÍDEO DIGITAL

Carlos Eduardo de Castro Santana^{*}, Cairo L. Nascimento Jr.^{**}, Fernando T. Sakane^{**}

^{*} EMBRAER – Divisão de Sistemas Mecânicos

Av. Brigadeiro Faria Lima, 2170
12227-901 – São José dos Campos – SP

^{**} Instituto Tecnológico de Aeronáutica
Divisão de Engenharia Eletrônica

12228-900 – São José dos Campos – SP

E-mails: csantana@embraer.com.br, cairo@ele.ita.br, sakane@ele.ita.br

Resumo: São apresentados nesse artigo o desenvolvimento, implementação e teste de uma técnica envolvendo uma rede neural de Hopfield modificada para detectar e estimar o vetor-movimento utilizado na codificação da diferença entre dois quadros consecutivos de um sinal de vídeo digitalizado, conforme proposto no padrão MPEG entre outros. Apresentam-se inicialmente a rede neural de Hopfield modificada e como os seus parâmetros são obtidos a partir da luminância dos pixels de dois quadros consecutivos. Em seguida é mostrado que a detecção e estimação do vetor-movimento podem ser obtidas quando a dinâmica da rede neural de Hopfield minimiza uma certa função de custo definida de forma apropriada para este problema. A técnica proposta nesse trabalho foi testada em duas seqüências de quadros compostos por pixels com 256 níveis de cinza, sendo a primeira seqüência sintética e a segunda uma seqüência real. Simulações da técnica proposta e da técnica da busca exaustiva mostram que para ambas as seqüências testadas os vetores-movimento foram detectados e estimados corretamente.

Palavras Chaves: detecção e estimação de movimento, codificação digital de sinais de vídeo, rede neural de Hopfield.

Abstract: This article concerns the development, implementation and tests of a modified Hopfield neural network for the detection and estimation of the motion-vector used to code the difference between two consecutive frames of a digitized video signal. Firstly the modified Hopfield neural network is presented and then it is shown: a) how its parameters can be derived from the luminance values of pixels in the two consecutive frames, and b) that an estimate of the motion-vector can be generated by letting the dynamics of the modified Hopfield neural network iteratively minimize a certain cost function properly defined for this problem. The proposed solution is then applied to two frame sequences with 256 gray-leved pixels, the first sequence is synthetic while the second is a natural sequence. Computer simulations of the

proposed solution and an exhaustive search technique showed that the correct motion-vector was obtained in both tested sequences.

Keywords: movement detection and estimation, digital video coding, Hopfield neural network.

1 INTRODUÇÃO

Embora amplamente utilizada nos dias de hoje, a digitalização de sinais de vídeo traz o inconveniente de gerar uma quantidade de dados bastante alta, o que requer uma grande largura de faixa de frequência para a transmissão de informação e uma grande capacidade de armazenamento. Como consequência, se torna cada vez mais importante o desenvolvimento e a utilização de técnicas eficientes de redução de dados e compressão de sinais de vídeo (Tekalp, 1995).

Codificadores de vídeo mais eficientes, como aqueles utilizados pelo padrão MPEG, baseiam-se no princípio da codificação diferencial preditiva com compensação de movimento para melhor explorar a redundância temporal entre os quadros do sinal de vídeo, o que requer o uso da detecção e estimação de movimento. Existem vários métodos para a detecção e a estimação de movimento como, por exemplo, os métodos baseados no princípio do casamento de blocos (Tekalp, 1995).

A melhor exploração das técnicas de solução do problema de detecção e estimação de movimento ainda é um campo aberto de pesquisa. Porém visto que este problema pode ser formulado como um problema de otimização e sendo as redes neurais uma ferramenta que tem se mostrado eficiente para resolver este tipo de problema (particularmente pela possibilidade de implementação em hardware com processamento paralelo), então redes neurais têm sido aplicadas para a detecção e estimação de movimento (Fuhrt *et alii*, 1997; Tekalp, 1995; Skrzypkowiak e Jain 1994; Shackelford, 1989).

A técnica desenvolvida e apresentada nesse artigo utiliza a rede neural de Hopfield com atualização das suas unidades em

instantes de tempo discretos (Hopfield e Tank, 1985; Tank e Hopfield, 1986, Nascimento Jr. e Yoneyama, 2000) e é composta por duas etapas:

1. os parâmetros da rede neural (pesos e bias) são determinados utilizando-se os valores das luminâncias dos pixels do quadro para um instante de tempo t (quadros atual) e do quadro para um instante de tempo $t - 1$ (quadro referência);
2. as saídas das unidades da rede neural são atualizadas iterativamente até que as mesmas se estabilizem, caracterizando assim a convergência da dinâmica da rede neural para um ponto de equilíbrio local.

Quando a rede neural convergir para um ponto de equilíbrio ficam então definidos a solução proposta pela rede neural e o vetor-movimento que desejamos detectar e estimar.

Cada ponto de equilíbrio local da dinâmica da rede neural pode corresponder a um valor mínimo local ou global de uma determinada função escalar pseudo-energia. Se for um mínimo global a solução do problema é ótima no sentido global. Não se sabe a priori se os parâmetros determinados para a rede neural resultarão em um ponto de mínimo global ou local (Hopfield e Tank, 1985; Tank e Hopfield, 1986).

2 REDE NEURAL DE HOPFIELD COM ATUALIZAÇÃO DISCRETA E SAÍDA DAS UNIDADES DISCRETA

2.1 Dinâmica da Rede Neural de Hopfield

A figura 1 mostra uma rede neural de Hopfield formada por uma única camada constituída por N unidades com saída discreta.

Cada unidade i possui dois conjuntos de estados de saída, $y_i = 0$ ou 1 para a rede cujas unidades são unipolares, e $y_i = -1$ ou 1 para a rede cujas unidades são bipolares.

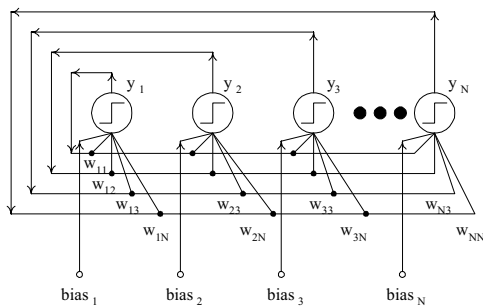


Figura 1: Rede de Hopfield com unidades discretas.

Denotando por w_{ij} o peso recebido pela unidade i vindo da unidade j , a entrada total da unidade i no instante de tempo discreto k , denotada por $net_i(k)$, é definida por:

$$net_i(k) = \sum_{j=1}^N w_{ij} y_j(k) + bias_i \quad (1)$$

A saída de cada uma das unidades no instante $k+1$ pode ser determinada pela função *threshold* (ou limiar) como a seguir:

$$y_i(k+1) = \begin{cases} 1, & \text{se } net_i(k) > 0 \text{ e unidade uni ou bipolar;} \\ 0, & \text{se } net_i(k) < 0 \text{ e unidade unipolar;} \\ -1, & \text{se } net_i(k) < 0 \text{ e unidade bipolar;} \\ y_i(k), & \text{se } net_i(k) = 0 \text{ e unidade uni ou bipolar.} \end{cases} \quad (2)$$

\mathbf{W} denota a matriz de pesos com dimensão $N \times N$, onde:

$$\mathbf{W} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1N} \\ w_{21} & w_{22} & \cdots & w_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ w_{N1} & w_{N2} & \cdots & w_{NN} \end{bmatrix} \quad (3)$$

Hopfield propôs a atualização assíncrona das unidades da rede neural, ou seja, que em um dado instante de tempo k apenas uma unidade da rede é escolhida para mudar de estado. O processo de atualização se repete até que a rede encontre um ponto de equilíbrio estável, o que é evidenciado quando a saída de cada unidade da rede se mantém constante após a atualização de todas as suas unidades.

O escalar $E(k)$, definido pela equação (4), pode ser interpretado como o valor da **função pseudo-energia da rede neural de Hopfield** no instante de tempo k (Hopfield e Tank, 1985; Tank e Hopfield, 1986):

$$E(k) = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N w_{ij} y_i(k) y_j(k) - \sum_{i=1}^N bias_i y_i(k) \quad (4)$$

Note que o escalar $E(k)$ apresenta limites superior e inferior pois todos os elementos no lado direito da equação (4) são limitados inferior e superiormente.

2.2 Estabilidade da Rede Neural

O desenvolvimento a seguir demonstra condições que são suficientes (mas não necessárias) para a estabilidade da rede neural de Hopfield com atualização discreta no tempo e saídas com valores discretos a partir de qualquer condição inicial.

Por rede neural de Hopfield estável entende-se que dada uma condição inicial qualquer das suas unidades a rede irá convergir para um ponto de equilíbrio e nele permanecerá. Nesse caso fica excluída a possibilidade de oscilações permanentes na saída das unidades da rede como p. ex. ciclos limites (Santana 2002).

Supondo atualização assíncrona (onde apenas uma unidade é atualizada de cada vez), que na iteração k será atualizada a unidade L , $1 \leq L \leq N$ e que a matriz \mathbf{W} é simétrica, então a partir da equação (4) pode-se mostrar que (Santana 2002):

$$\Delta E(k+1) = -\Delta y_L(k+1) net_L(k) - \frac{1}{2} w_{LL} [\Delta y_L(k+1)]^2 \quad (5)$$

onde as seguintes definições foram usadas:

$$\begin{aligned} \Delta E(k+1) &= E(k+1) - E(k), \\ \Delta y_L(k+1) &= y_L(k+1) - y_L(k). \end{aligned}$$

Portanto se os elementos da diagonal principal da matriz \mathbf{W} forem não negativos ($w_{LL} \geq 0$) e a atualização da saída das unidades for implementada pela função *threshold* de acordo com a equação (2), então cada vez que uma unidade alterar a sua saída o valor da função escalar pseudo-energia necessariamente tem que diminuir. Como esta função tem valor inicial finito e é limitada inferiormente, pode-se afirmar que o valor do escalar $E(k)$ é sempre finito e diminui

monotonicamente à medida que k aumenta. A demonstração detalhada para a estabilidade da rede neural de Hopfield pode ser encontrada em Santana (2002).

3 RESOLUÇÃO DE PROBLEMAS DE OTIMIZAÇÃO UTILIZANDO A REDE NEURAL DE HOPFIELD

As redes neurais de Hopfield são muito utilizadas para solucionar problemas de otimização com restrições (Hopfield e Tank, 1985; Tank e Hopfield, 1986; Mendes, 2000). Para tal, é necessário que seja definida a função escalar pseudo-energia apropriada para o problema que deseja-se resolver. A equação abaixo mostra a expressão geral da função pseudo-energia, denotada por E_p , para um problema de otimização qualquer:

$$E_p = E_c + \sum_j R_j E_j \quad (6)$$

O primeiro termo E_c corresponde à função original que se deseja minimizar e o segundo termo corresponde às j restrições do problema que devem ser satisfeitas. Cada restrição é ponderada pelo parâmetro constante R_j , onde $R_j > 0$ para $\forall j$ e as funções E_j são projetadas de forma tal que: a) quando a restrição j é satisfeita então $E_j = 0$, e b) quando a restrição j não está satisfeita então $E_j > 0$. Espera-se que ao ser minimizada a função E_p , então a função custo E_c foi minimizada e as j restrições do problema foram obedecidas.

4 A TÉCNICA “SIMULATED ANNEALING” (TÊMPERA SIMULADA)

Quando a rede neural de Hopfield com atualização assíncrona e função *threshold* é utilizada para resolução de problemas de otimização, os experimentos computacionais mostram que a rede pode convergir para mínimos locais indesejáveis que não representam boas soluções para o problema (Hopfield e Tank, 1985; Tank e Hopfield, 1986; Mendes, 2000).

A técnica denominada “Simulated Annealing” (Kirkpatrick *et alii*, 1983) pode então ser utilizada neste caso para alterar a dinâmica da rede neural e possivelmente aumentar a “zona de atração” do ponto de mínimo global da função pseudo-energia, aumentando assim a possibilidade de convergência para soluções ótimas do problema (Santana, 2002).

A aplicação da técnica “Simulated Annealing” na dinâmica da rede neural de Hopfield significa usar, no lugar da função *threshold*, a função sigmóide e aumentar monotonicamente a cada iteração o ganho desta função. Assim para valor de ganho suficientemente grande a função sigmóide tende para a função *threshold* (a função sigmóide pode ser interpretada como uma versão suavizada da função *threshold*).

5 O PROBLEMA DA DETECÇÃO E ESTIMAÇÃO DE MOVIMENTO UTILIZANDO A REDE NEURAL DE HOPFIELD

A figura 2 ilustra um bloco do quadro atual, chamado de bloco F, no qual se deseja detectar e estimar o vetor-movimento em relação ao quadro referência.

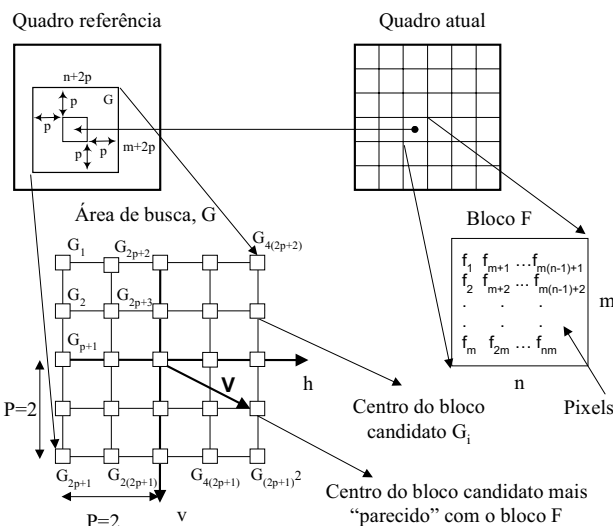


Figura 2: Ilustração do bloco F no qual se deseja detectar e estimar o vetor-movimento

A este bloco F é associada uma área de busca G no quadro referência que engloba $(2p+1)^2$ blocos candidatos, onde p é chamado de parâmetro de busca e define o número de blocos candidatos englobados pela área de busca G. Na figura 2, a título de ilustração, o parâmetro p é apresentado como sendo 2.

A coordenada espacial do bloco candidato localizado no centro da área de busca é a mesma do bloco F no quadro atual e é assumido como sendo esta a coordenada de referência para os demais blocos candidatos.

Deseja-se então aplicar uma rede neural de Hopfield para identificar o bloco candidato mais “parecido” com o bloco F, onde coordenada espacial deste bloco no interior da área de busca define o vetor-movimento associado ao bloco F. Tal vetor-movimento é composto por duas componentes, a componente horizontal h e a componente vertical v .

Primeiramente determina-se que o número de unidades da rede neural de Hopfield será igual ao número de blocos candidatos englobados pela área de busca G.

Para a determinação dos parâmetros da rede neural de Hopfield (matriz de pesos \mathbf{W} e vetor *bias*) é necessário conhecer: a) o valor da luminância dos pixels do bloco do quadro atual, e b) o valor da luminância dos pixels dos blocos candidatos englobados pela área de busca G. Então o vetor \mathbf{F} , a matriz \mathbf{G} e o vetor \mathbf{Y} são definidos a seguir.

\mathbf{F} é um vetor de dimensão $N \times 1$, cujos elementos são os valores da luminância dos pixels do bloco atual, os quais representados como bloco F na figura 2. As dimensões em número de pixels no sentido vertical e horizontal do bloco F são respectivamente m e n . $N = n \times m$ representa o número de pixels do bloco F e de cada bloco candidato. Assim temos:

$$\mathbf{F} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_m \\ f_{m+1} \\ \vdots \\ f_{2m} \\ \vdots \\ f_N \end{bmatrix}$$

\mathbf{G} é uma matriz de dimensão $M \times N$ cujos elementos são os valores da luminância dos pixel dos blocos candidatos englobados pela área de busca G . As dimensões em número de pixels no sentido vertical e horizontal dos blocos candidatos são as mesmas do bloco F . $M = (2p + 1)^2$ representa o número de blocos candidatos englobados pela área de busca G . Portanto temos:

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_1 \\ \mathbf{G}_2 \\ \vdots \\ \mathbf{G}_{2p+1} \\ \mathbf{G}_{2p+2} \\ \vdots \\ \mathbf{G}_{2(2p+1)} \\ \vdots \\ \mathbf{G}_M \end{bmatrix},$$

onde cada vetor \mathbf{G}_j , para $j = 1, 2, \dots, M$, representa o valor da luminância dos pixels de cada bloco candidato G_j . Assim temos:

$$\mathbf{G} = \begin{bmatrix} g_1^1 & g_2^1 & \dots & g_m^1 & g_{m+1}^1 & \dots & g_{2m}^1 & \dots & g_N^1 \\ g_1^2 & g_2^2 & \dots & g_m^2 & g_{m+1}^2 & \dots & g_{2m}^2 & \dots & g_N^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ g_1^M & g_2^M & \dots & g_m^M & g_{m+1}^M & \dots & g_{2m}^M & \dots & g_N^M \end{bmatrix},$$

onde g_i^j representa o i -ésimo pixel do j -ésimo bloco candidato, onde $1 \leq i \leq N$ e $1 \leq j \leq M$.

Define-se \mathbf{Y} como um vetor $M \times 1$ cujos elementos são as saídas das unidades da rede neural, ou seja:

$$\mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix}$$

Deseja-se encontrar os parâmetros de uma rede neural de Hopfield (matriz de pesos \mathbf{W} e vetor **bias**) que irá determinar qual é o bloco candidato G_i mais "parecido" com o bloco F do quadro atual, o que significa minimizar a expressão da função

custo definida como $\left\| \mathbf{F} - \sum_{j=1}^M y_j (\mathbf{G}_j)^T \right\|^2$, onde

$\|\mathbf{X}\|^2 = \mathbf{X}^T \mathbf{X} = \sum_{i=1}^N (x_i)^2$ e \mathbf{X} é vetor coluna com N componentes.

Então se existir o vetor \mathbf{G}_i tal que $\left\| \mathbf{F} - (\mathbf{G}_i)^T \right\|^2 = 0$, queremos encontrar y_j que minimize a expressão abaixo:

$$\min \frac{1}{2} \left\| \mathbf{F} - \sum_{j=1}^M y_j (\mathbf{G}_j)^T \right\|^2 = \min \frac{1}{2} \left\| \mathbf{F} - [\mathbf{G}_1^T \dots \mathbf{G}_M^T] \mathbf{Y} \right\|^2$$

onde os componentes do vetor coluna \mathbf{Y} são: $y_j = 0$ se $j \neq i$ e $y_j = 1$ se $j = i$.

De acordo com a equação (6), é necessário a definição dos termos da função escalar pseudo-energia de tal forma a incluir as restrições que devem ser obedecidas no problema original.

A componente E_c que corresponde ao problema original é:

$$E_c = \frac{1}{2} \left\| \mathbf{F} - [\mathbf{G}_1^T \dots \mathbf{G}_M^T] \mathbf{Y} \right\|^2$$

$$E_c = \frac{1}{2} \sum_{i=1}^N \left(f_i - \sum_{j=1}^M y_j g_i^j \right)^2$$

$$E_c = \frac{1}{2} \sum_{i=1}^N (f_i)^2 - \sum_{i=1}^N \sum_{j=1}^M f_i y_j g_i^j + \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^M \sum_{k=1}^M y_j y_k g_i^j g_i^k$$

Considerando que a rede neural será composta por unidades unipolares a expressão $-\sum_{j=1}^M h_j y_j (y_j - 1)$ é adicionada a E_c para

atuar como a restrição. Note que tal expressão será: a) nula, se a saída de cada unidade da rede for 0 ou 1, e b) positiva, em caso contrário, desde que todos os escalares h_j sejam positivos.

Pode-se então definir a seguinte expressão da função escalar pseudo-energia para o problema de detecção e estimação de movimento utilizando a rede neural de Hopfield:

$$E_p = E_c - \sum_{j=1}^M h_j y_j (y_j - 1) \quad (7)$$

Igualando-se a equação acima com a expressão da função pseudo-energia da rede de Hopfield, equação (4), teremos:

$$bias_c = \sum_{i=1}^N f_i g_i^c - h_c \quad (8)$$

$$w_{cd} = \begin{cases} -\sum_{i=1}^N g_i^c g_i^d, & \text{se } c \neq d \\ -\sum_{i=1}^N (g_i^c)^2 + 2h_c, & \text{se } c = d \end{cases}$$

onde $1 \leq c \leq M$, $1 \leq d \leq M$ e a condição $h_c \geq \frac{1}{2} \sum_{i=1}^N (g_i^c)^2$ deve ser

obedecida pois os elementos da diagonal principal da matriz de pesos \mathbf{W} devem ser nulos ou positivos (uma das condições suficientes para a estabilidade da rede neural de Hopfield, ver seção 2.2).

6 SIMULAÇÕES E RESULTADOS

Nessa seção são apresentadas simulações utilizando redes neurais de Hopfield aplicadas às seqüências sintéticas e reais de quadros de um sinal de vídeo digital.

As dimensões vertical e horizontal desses quadros em número de pixels são, respectivamente, 240 x 352, onde o valor da luminância dos pixels é representada por 256 tons de cinza.

As dimensões horizontal e vertical em número de pixels do bloco F são $m = n = 8$ e da área de busca é $(2p + m) = (2p + n) = 14$ pixels, onde o parâmetro de busca p em ambos os sentidos das direções horizontal e vertical é 3. Esse valor de p define $(2p + 1)^2 = 49$ blocos candidatos englobados pela área de busca.

Para as simulações a seqüência abaixo foi seguida:

1. Escolher o bloco do quadro atual em que se quer detectar e estimar o vetor-movimento (ver seção 6.1);
2. Estimar os parâmetros **bias** e **W** da rede neural utilizando a equação (8);
3. Inicializar as unidades da rede (ver seção 6.2);
4. Atualizar as unidades da rede neural de maneira assíncrona utilizando a técnica “Simulated Annealing” (ver seção 6.3) até ela atingir um ponto de equilíbrio estável (ver seção 6.4).

Nesse caso o estado das unidades da rede define o vetor-movimento do bloco do quadro atual escolhido em relação ao quadro referência.

6.1 Divisão dos Quadros Atual e Referência em Blocos

A figura abaixo mostra a divisão dos quadros atual e referência em blocos utilizada nas simulações.

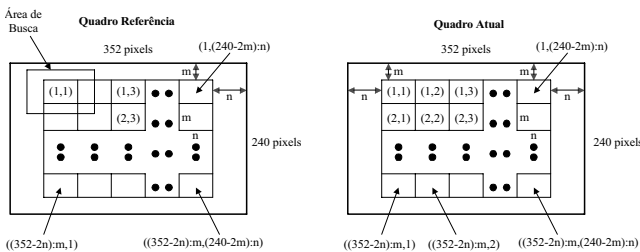


Figura 3: Divisão dos quadros atual e referência em blocos de dimensão vertical *m* pixels e dimensão horizontal *n* pixels.

6.2 Inicialização das Unidades da Rede Neural

Se todas as unidades da rede forem inicializadas com o mesmo valor de saída, a rede pode ficar “presa” neste ponto (situação de equilíbrio local instável). Assim o procedimento tradicionalmente usado é baseado na escolha de um valor escalar fixo ao qual um pequeno ruído aleatório com distribuição uniforme ou gaussiana é adicionado (Hopfield e Tank, 1985). Como nas simulações as unidades da rede neural são unipolares, então cada unidade foi inicializada por:

$$net(0) = 0,1(2 \text{ rand} - 1)$$

$$y(1) = \frac{1}{1 + \exp[-\lambda_0 net(0)]} \quad (9)$$

onde *rand* é número aleatório com distribuição uniforme entre 0 e 1. Assim *net(0)* pertence ao intervalo $[-0,1; 0,1]$, *y(1)* varia em torno de 0,5 e cada unidade é inicializada com um valor ligeiramente diferente.

6.3 Variação do Ganho da Função de Saída das Unidades da Rede Neural

Como foi visto na seção 4, para diminuir a possibilidade da rede convergir para mínimos locais que não representam boas soluções para o problema, o ganho λ da função sigmoíde (definida pela equação (9)) deve ser monotonicamente crescente. Nas simulações apresentadas o ganho λ foi aumentado linearmente a cada iteração da rede da seguinte forma:

$$\lambda(k) = \lambda_0 + \frac{(k-1)(\lambda_F - \lambda_0)}{Nit - 1} \quad (10)$$

onde λ_0 e λ_F são respectivamente o ganho inicial e final, *Nit* é o número máximo de iterações, e *k* é o número da iteração atual. Assim $\lambda(k) = \lambda_0$ para *k* = 0, e $\lambda(k) = \lambda_F$ para *k* = *Nit*.

6.4 Convergência da Rede Neural

Nas simulações apresentadas estabeleceu-se que a convergência da rede neural é obtida quando a diferença da saída de todas as unidades da rede entre duas iterações consecutivas é inferior a um certo limiar determinado pelo escalar ϵ , ou seja, quando:

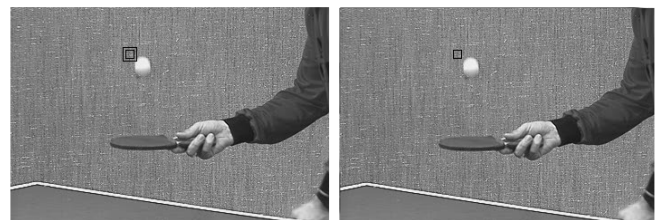
$$\|Y(k+1) - Y(k)\| < \epsilon \quad (11)$$

Nessas simulações as saídas das unidades da rede são atualizadas usando a função sigmoíde (equação (9)) até que a equação (11) seja satisfeita. Nas iterações seguintes as saídas das unidades são então atualizadas usando a função *threshold* até que a equação (11) seja novamente satisfeita. Assume-se então que a rede convergiu para uma solução.

6.5 Detecção e Estimação do Vetor-movimento para o Bloco F Localizado na Posição (6, 16)

Nesta simulação o quadro atual foi obtido deslocando-se todos os pixels do quadro referência 1 pixel para baixo (no sentido vertical) e 1 pixel para a direita (no sentido horizontal).

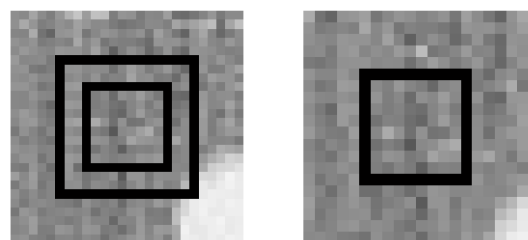
O bloco F escolhido para detecção e estimação do vetor-movimento em relação ao quadro referência está localizado na posição (6, 16) do quadro atual.



a) Área de busca do quadro referência. b) Bloco F do quadro atual.

Figura 4: Área de busca do quadro referência e o bloco F localizado na posição (6, 16) do quadro atual.

A figura 5 mostra o *zoom* em torno da área de busca do quadro referência e do bloco F localizado na posição (6, 16) do quadro atual, realçando os pixels do bloco F e da área de busca.



a) *Zoom* em torno da área de busca do quadro referência. b) *Zoom* em torno do bloco F do quadro atual.

Figura 5: *Zoom* em torno da área de busca do quadro referência e do bloco F localizado na posição (6, 16) do quadro atual.

A tabela 1 mostra os valores para os parâmetros (λ_0 , λ_F , Nit , ϵ e q) que definem a dinâmica da rede neural na simulação. Estes parâmetros foram ajustados por tentativa até que o vetor-movimento definido pela rede coincidissem com o deslocamento aplicado aos pixels do quadro referência.

Tabela 1: Parâmetros utilizados para detectar e estimar o vetor-movimento do bloco F localizado na posição (6, 16) do quadro atual em relação ao quadro referência.

Parâmetro	Valor
Nit (número máximo de iterações da rede neural)	10000 iterações
λ_0 (ganho da função sigmóide utilizado para definir o estado inicial das unidades da rede neural)	0,1
λ_F (ganho da função sigmóide quando a iteração da rede for Nit)	25
ϵ (limiar que define a convergência da rede)	0,00001
q (ver observação abaixo)	1

Obs.: Como a condição $h_c \geq \frac{1}{2} \sum_{i=1}^N (g_i^c)^2$ deve ser obedecida para

$1 \leq c \leq M$ (ver equação (8)), fazemos $h_c = \frac{q}{2} \sum_{i=1}^N (g_i^c)^2$ e arbitramos o parâmetro q obedecendo a restrição $q \geq 1$.

A tabela 2 mostra as componentes horizontal e vertical do vetor-movimento detectado e estimado para o bloco F utilizando-se a técnica neural. Santana (2002) mostra que este é o mesmo resultado obtido utilizando-se a técnica de busca exaustiva.

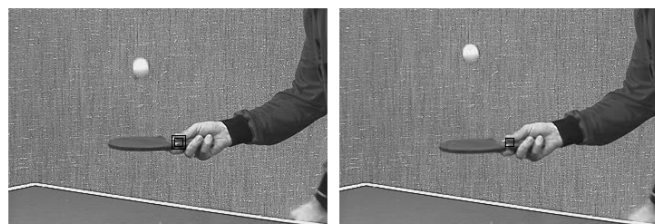
Tabela 2: Vetor-movimento detectado e estimado para o bloco F localizado na posição (6, 16) do quadro atual utilizando-se a técnica neural.

Número de iterações na simulação	266
Vetor-movimento (deslocamento horizontal)	1 pixel para a esquerda
Vetor-movimento (deslocamento vertical)	1 pixel para cima

6.6 Detecção e Estimação do Vetor-movimento para o Bloco F Localizado na Posição (18, 23)

A figura 6 mostra os quadros referência e atual utilizados nessa simulação, os quais pertencem a mesma seqüência de quadros em movimento de um certo sinal de vídeo digital real.

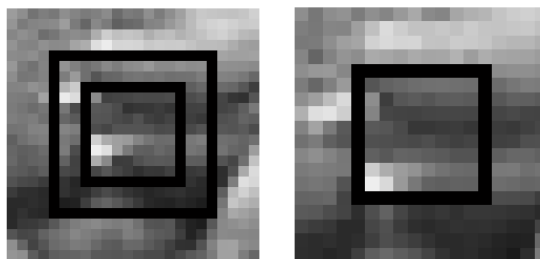
A figura 6 mostra também o bloco F localizado na posição (18, 23) do quadro atual e a área de busca associada ao bloco F no quadro referência.



a) Área de busca do quadro referência. b) Bloco F do quadro atual.

Figura 6: Área de busca do quadro referência e o bloco F localizado na posição (18, 23) do quadro atual.

A figura 7 mostra o zoom em torno da área de busca do quadro referência e do bloco F localizado na posição (18, 23) do quadro atual.



a) Zoom em torno da área de busca do quadro referência. b) Zoom em torno do bloco F do quadro atual.

Figura 7: Zoom em torno da área de busca do quadro referência e do bloco F localizado na posição (18, 23) do quadro atual.

A tabela 3 mostra os valores para os parâmetros (λ_0 , λ_F , Nit , ϵ e q) que definem a dinâmica da rede neural na simulação.

Tabela 3: Parâmetros utilizados para detectar e estimar o movimento do bloco F localizado na posição (18, 23) do quadro atual em relação ao quadro referência.

Parâmetro	Valor
Nit (número máximo de iterações da rede neural)	10000 iterações
λ_0 (ganho da função sigmóide utilizado para definir o estado inicial das unidades da rede neural)	0,1
λ_F (ganho da função sigmóide quando a iteração da rede for Nit)	25
ϵ (limiar que define a convergência da rede)	0,0001
q (mesma observação da tabela 1)	0,1

A tabela 4 mostra as componentes horizontal e vertical do vetor-movimento detectado e estimado para o bloco F utilizando-se a técnica neural. Santana (2002) mostra, também neste caso, que este é o mesmo resultado obtido utilizando-se a técnica de busca exaustiva.

Tabela 4: Vetor-movimento detectado e estimado para o bloco F localizado na posição (18, 23) do quadro atual utilizando-se a técnica neural

Número de iterações na simulação	195
Componente dh do vetor-movimento, (deslocamento horizontal)	0 pixel
Componente dv do vetor-movimento, (deslocamento vertical)	2 pixels para cima

7 CONCLUSÕES

Codificadores de vídeo mais eficientes, como aqueles utilizados pelo padrão MPEG, baseiam-se no princípio da codificação diferencial preditiva com compensação de movimento para melhor explorar a redundância temporal entre os quadros do sinal de vídeo, o que requer o uso da detecção e estimação de movimento.

Sendo o processamento da detecção e estimação de movimento um aspecto crítico do codificador, principalmente pelo tempo e número de operações requeridos, as redes neurais podem ser uma ferramenta muito eficiente para tal aplicação, uma vez que elas podem ser implementadas em hardware com processamento paralelo.

Este trabalho apresentou o desenvolvimento, a implementação e os teste de uma técnica que utiliza a rede neural de Hopfield para detectar e estimar o movimento de um quadro atual em relação a um quadro referência.

Foram apresentados resultados de simulações utilizando dois quadros consecutivos de seqüências temporal sintética e real, onde as luminâncias dos pixels eram representadas por 256 níveis de cinza, os blocos eram compostos por 64 pixels (8 x 8 pixels) e área de busca englobava 49 blocos candidatos.

A rede neural de Hopfield desenvolvida foi testada para alguns blocos do quadro atual e foi observado que os vetores-movimento foram corretamente determinados uma vez que coincidiram com os vetores-movimento obtidos utilizando-se a técnica da busca exaustiva onde a solução ótima é sempre obtida (Santana 2002).

As simulações também mostraram que o aumento do número de unidades da rede (igual ao número de blocos candidatos na área de busca) torna a dinâmica da rede mais sensível a pequenas variações nos valores dos parâmetros λ_0 , λ_F , q , ϵ e N_{it} .

Acreditamos que a dificuldade encontrada para se ajustar estes parâmetros está relacionada ao efeito dos mesmos sobre os auto-valores e auto-vetores da matriz de pesos da rede neural, conforme mostrado por Aiyer *et alii* (1990) no caso do TSP (*Traveling Salesman Problem*).

Durante as simulações foi também observado que, uma vez determinado um conjunto de parâmetros para uma rede neural estimar corretamente o vetor-movimento de um determinado bloco do quadro atual, tais parâmetros não necessariamente serão adequados para realizar a mesma tarefa para um outro bloco dentro de um mesmo quadro atual.

Algumas possíveis linhas para investigações futuras são:

- desenvolvimento de melhores métodos de ajuste dos parâmetros das redes neurais visando: 1) a sua menor dependência em relação aos diferentes blocos de um mesmo quadro, 2) uma maior velocidade de convergência para a solução correta, e 3) maior facilidade de ajuste no caso de blocos com um número maior de pixels e áreas de busca com maior número de blocos candidatos,
- comparação da eficiência da solução para este problema, em termos de número de operações ou tempo de computação, entre a técnica que usa a rede neural de Hopfield e outras técnicas como p. ex. a técnica de busca exaustiva.

REFERÊNCIAS BIBLIOGRÁFICAS

- Aiyer, S. V. B., M. Niranjan and F. Fallside (1990). A Theoretical Investigation into the Performance of the Hopfield Model. *IEEE Transactions on Neural Networks*, Vol. 1, nº 2, pp. 204-215.
- Furht, B., J. Greenberg and R. Westwater (1997). *Motion Estimation Algorithms for Video Compression*. Kluwer Academic Publishers, Boston.
- Hopfield, J. J. & D. W. Tank (1985). Neural Computation of Decision in Optimization Problems. *Biological Cybernetics*, Vol. 52, nº 3, pp. 141-152.
- Kirkpatrick, S., C. D. Gelatt and M. P. Vecchi (1983). Optimization by Simulated Annealing. *Science*, Vol. 220, nº 4598, pp 671-680.
- Mendes, L. A. G. (2000), Sistema Inteligente para Determinação de Seqüências de Montagem Otimizadas. Tese de Mestrado, Instituto Tecnológico de Aeronáutica, ITA-IEM, São José dos Campos – SP.
- Nascimento Jr., C. L. & T. Yoneyama (2000). *Inteligência Artificial em Controle e Automação*. Edgard Blücher, São Paulo.
- Santana, C. E. C. (2002), Aplicação de Redes Neurais de Hopfield para Detecção e Estimação de Movimento entre Dois Quadros de um Sinal de Vídeo Digital. Tese de Mestrado, Instituto Tecnológico de Aeronáutica, ITA-IEE, São José dos Campos – SP.
- Shackleford, J. B. (1989, Jun.). Neural Data Structures: Programming with Neurons. *Hewlett-Packard Journal* pp. 69-78.
- Skrzypkowiak, S. S. & V. K. Jain (1994). Video Motion Estimation using a Neural Network. *IEEE International Symposium on Circuits and Systems*, pp 217-220.
- Tank, D. W. & J. J. Hopfield (1986). Simple “Neural” Optimization Networks: an A/D Converter, Signal Decision Circuit, and a Linear Programming Circuit. *IEEE Transactions on Circuits and Systems*, Vol. 33, nº 5, pp 533-541.
- Tekalp, A. M (1995). *Digital Video Processing*. Prentice Hall, New Jersey.