Um Sistema Sistólico para Interpolação de Imagens

EDUARDO TOLEDO SANTOS

Escola Politécnica da Universidade de São Paulo
LSI - Laboratório de Sistemas Integráveis
Div. de Sist. Digitais - Grupo de Computação Gráfica
Av. Prof. Luciano Gualberto, travessa 3, n.158
05508-900 - São Paulo, SP, Brasil
toledo@lsi.usp.br

Abstract. This paper presents a systolic system for performing interpolation by Newton's method. A specific application is presented: image resolution enhancement. It's shown the proposed architecture presents interesting features relating to systolic systems.

1. Introdução

A altíssima capacidade de processamento requerida por certas aplicações aliada ao grande desenvolvimento da tecnologia de circuitos integrados tem propiciado o aparecimento dos chamados *sistemas de uso ou aplicação específica ("special-purpose systems")*.

Os sistemas sistólicos são um tipo de arquitetura que adapta-se extremamente bem à implementação de dispositivos de aplicação específica pois ensejam vários aspectos positivos como se verá a seguir.

Introduzidos pelo trabalho pioneiro de Kung e Leiserson [1], os sistemas sistólicos caracterizam-se pelo uso de processadores extremamente simples, chamados "células", que realizam operações elementares sobre um fluxo de dados que são "bombeados" através das células do sistema. O termo "sistólico" é uma analogia com a contração cardíaca (sístole) que impulsiona o sangue no sistema circulatório. Além da simplicidade dos elementos processadores, também caracterizam os sistemas sistólicos o pequeno número de tipos diferentes de células, a comunicação local entre processadores, a regularidade e modularidade dos sistemas e o fato de servirem à solução de problemas específicos que requerem curto tempo de resposta [2].

Existe uma série de aspectos chave relacionados ao projeto de sistemas de uso específico que mostram a aplicabilidade de arquiteturas sistólicas para este fim [3]. Estes aspectos, explanados a seguir, serão retomados mais a diante para caracterizar o sistema apresentado neste trabalho.

O custo de um sistema é um item de fundamental importância num projeto. Em projetos VLSI ("Very Large Scale of Integration") é especialmente verdade que uma grande produção (quantitativa) reduz tanto os custos fixos por "chip" quanto dilui o custo do projeto ("design"). No entanto, em se tratando de sistemas para aplicações específicas, a produção em baixas quantidades é a regra fazendo com que os custos fixos tornem-se pouco significativos em relação ao custo de desenvolvimento (projeto, "lay-out" do chip, depuração, etc.). Os sistemas sistólicos são uma resposta adequada a este problema na medida em que seu pressuposto básico é que sejam constituídos por **células simples**, de **poucos tipos diferentes**, passíveis de serem agrupadas em grande número, originando assim, **projetos simples e regulares**, e portanto baratos, porém extremamente eficientes. Seguem em vários aspectos a metodologia proposta por Mead e Conway [4].

A evolução tecnológica dos computadores tem demonstrado que a velocidade individual dos componentes num "chip" não vem crescendo na mesma proporção da capacidade de integração encontrada em sistemas VLSI [3]. Neste contexto, arquiteturas paralelas exibindo conceitos de multiprocessamento e "pipelining" proporcionam novas perspectivas de incremento no desempenho destas máquinas. Os sistemas sistólicos fazem amplo uso das características mencionadas: **integração em larga escala**, **multiprocessamento** e **"pipelining"**.

Ao se agruparem muitas unidades processadoras para que realizem um trabalho conjunto, dois problemas críticos logo se apresentam: comunicação e sincronização. Os sistemas sistólicos implicitamente realizam muita comunicação entre suas várias células porém evitam complicações mantendo esta **comunicação local**, **simples**, **regular** e **síncrona**.

Por mais rápido que seja o processamento de um sistema de aplicação específica, o desempenho do sistema estará limitado pela sua banda de E/S. Isto significa que o desempenho do processamento deverá estar **balanceado com a taxa de E/S** de modo que o computador hospedeiro, a memória ou qualquer outro dispositivo externo ao qual o sistema esteja conectado possa adequadamente fornecer entradas e ler resultados gerados pelo sistema.

Além destes aspectos, é importante considerar no projeto de um sistema sistólico implementado com tecnologia VLSI que o número de pinos num circuito integrado é bastante limitado. Assim, deve-se procurar limitar os sinais de comunicação com o ambiente externo ao "chip" de modo a permitir um encapsulamento padrão.

A área de processamento de imagens recebeu grande atenção daqueles que desenvolvem algoritmos sistólicos por requerer, em geral, baixo tempo de resposta e/ou alta capacidade de processamento, além de tipicamente se prestarem bem para implementação através de sistemas sistólicos.

O sistema apresentado aqui também insere-se na área de processamento de imagens e visa aumentar a resolução aparente de uma imagem quando se tem a necessidade de ampliá-la. O método utiliza interpolação polinomial e será descrito em detalhes mais adiante. De qualquer forma, pode-se dizer que sua implementação em máquinas seriais não costuma apresentar resultado satisfatório, dependendo do tamanho das imagens, se há a necessidade de resultados em tempo real.

Kung e Picard [5] apresentam uma solução alternativa àquela aqui proposta.

2. Interpolação de imagens

Uma imagem, no sentido considerado neste trabalho, é uma matriz bi-dimensional $Y=[y_{ij}]$, cujos elementos indicam a intensidade do ponto relativo a sua posição. Numa imagem monocromática esta intensidade indica a luminância naquele ponto. O caso colorido é totalmente análogo considerando-se, não uma, mas três matrizes $R = [r_{ij}]$, $G = [g_{ij}]$ e $B = [b_{ij}]$ de mesmas dimensões cujas intensidades corresponderão, respectivamente, às componentes vermelha (R), verde (G) e azul (B) que sobrepostas originam a cor de cada ponto representado. Tipicamente estas intensidades são representadas por valores de oito bits com o que obtem-se resultados satisfatórios (256 tons de cinza ou 16.7 milhões de cores no caso colorido).

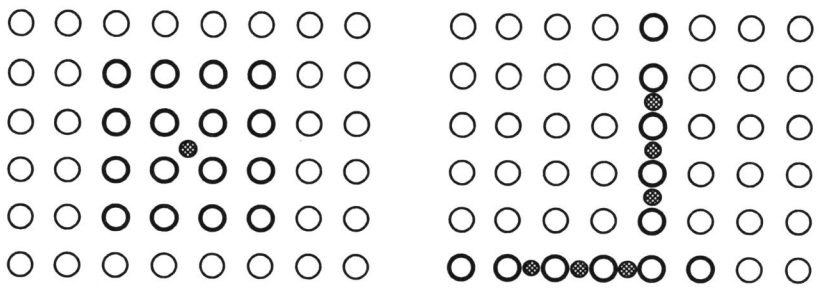
Os pontos da imagem, denominados "pixels", em seu conjunto formam a imagem representada. No entanto trata-se, em geral, apenas de uma representação discreta de uma realidade contínua. Ao ampliar-se uma imagem, os "pixels" deverão ficar maiores evidenciando a característica discreta da representação. A imagem apresentará um efeito indesejável chamado "mosaico" com grandes descontinuidades entre um "pixel" e outro. Na verdade, quando exibida num monitor, os "pixels" têm sempre a mesma dimensão, oque ocorre é que, ao ampliar-se a imagem, atribui-se a vários "pixels" do monitor a mesma cor, que representam assim, um único "pixel" da imagem original. A imagem ampliada não contém mais informação do que a imagem original e por isso os valores dos pixels são repetidos.

A técnica usada para artificialmente aumentar a resolução de uma imagem evitando o efeito "mosaico" quando da ampliação, consiste em, ao invés de atribuir ao conjunto de pixels da imagem ampliada que representam um único "pixel" da imagem original a intensidade deste, *interpolarem-se* os valores destes novos "pixels" em relação aos já existentes, suavizando-se descontinuidades. Desse modo, se utilizar-se uma função interpoladora adequada, evitar-se-á o surgimento de descontinuidades que denunciam a resolução da imagem original. Deve-se notar que trata-se de uma interpolação com função de duas variáveis já que a imagem é um ente bi-dimensional.

A escolha da função interpoladora representa uma questão de compromisso já que apesar de bons resultados serem conseguidos com, por exemplo, interpolação por superfícies Splines bi-cúbicas [6], o processamento requerido pode ser proibitivo. O uso de uma função linear pode ser extremamente simples mas os resultados serão insatisfatórios.

Neste trabalho escolheu-se a interpolação polinomial pelo método de Newton. Esta escolha deu-se em razão de sua adequação para implementação através de arquitetura sistólica e porque apresenta razoáveis resultados práticos como se pôde observar em implementação serial [7]. A interpolação se dará através da submatriz 4x4 em torno do ponto que se deseja interpolar (fig. 1a). Esta quantidade de pontos também representa uma questão de compromisso, porém o sistema sistólico proposto pode ser facilmente expandido para que se considerem mais pontos se desejado. Também disponível na arquitetura a ser apresentada, é a possibilidade de se utilizar o mesmo polinômio para se interpolarem vários pontos na mesma linha ou coluna (fig. 1b).

No caso de imagens coloridas, realiza-se o mesmo procedimento para cada uma das três componentes de cor independentemente, interpolando-se pontos nas mesmas coordenadas para cada componente de cor.



a) Pixel entre linhas e colunas b) Pontos alinhados

LEGENDA:
- Ponto da imagem original
- Ponto do polinômio interpolador
- Ponto (pixel) a interpolar

Fig.1 - Os pontos usados no cálculo do polinômio interpolador

3. Interpolação polinomial pelo método de Newton

3.1. Interpolação unidimensional

Dados n+1 pontos distintos (x_0,y_0), (x_1,y_1), ...(x_n,y_n), existe um único polinômio p de grau menor ou igual a n que passa por estes pontos, isto é, $p(x_i)= y_i$ [8]. Este polinômio, chamado *polinômio interpolador,* é único e pode ser calculado da seguinte forma, se fizermos a restrição

$$x_i = x_{i-1} + h,\ h = c^{te}. \quad (1)$$

(i.e., pontos equiespaçados):

$$p(x) = y_0 + \sum_{i=1}^{n} \frac{D_i(x_0)}{i!} \prod_{j=0}^{i-1} \frac{x - x_j}{h} \quad (2)$$

onde:

$$D_i(x_j) = \begin{cases} y_j, \text{ para } i = 0 \\ D_{i-1}(x_j+h) - D_{i-1}(x_j), \text{ para } 1 \le i \le n \end{cases}$$

são chamadas *diferenças simples.*

No caso particular que interessa a este trabalho, pode-se adotar h=1 sem perda de generalidade já que os "pixels" numa imagem são sempre equiespaçados. Por simplicidade, adotou-se espaçamento unitário. Mais ainda, pode-se supor $x_0=0$, $x_1=1$, ... $x_n=n$, pois os valores interpolados não sofrerão alterações devido a esta transformação.

Finalmente, como referido na secção anterior, far-se-á a interpolação sempre através de 4 pontos em cada dimensão de modo que podemos fixar n=3. Com estas simplificações e indicando $D_i(x_0)$ por D_i, o polinômio interpolador em sua forma expandida será:

$$p(x) = \frac{D_0}{0!} + x\frac{D_1}{1!} + x(x-1)\frac{D_2}{2!} + x(x-1)(x-2)\frac{D_3}{3!} \quad (3)$$

3.2. Interpolação bi-dimensional

Para interpolar-se um "pixel" contido em uma das linhas da imagem original basta calcular o polinômio interpolador que passa pelos dois pontos à sua esquerda e pelos dois à sua direita. Analogamente, a interpolação de um ponto contido numa coluna da imagem original é feita através do polinômio passando pelos dois pontos acima e abaixo dele.

Uma técnica para interpolar pontos não contidos em linhas ou colunas da imagem original é interpolarem-se pontos na mesma entrelinha do ponto considerado, porém em colunas correspondentes àquelas da imagem original. Calcula-se então o polinômio interpolador relativo aos novos pontos interpolados e, a partir deste, o ponto desejado. Evita-se, deste modo, o trato com funções de duas variáveis.

4. O sistema sistólico

4.1 O algoritmo sistólico

O algoritmo a ser implementado deve gerar as diferenças simples necessárias ao cálculo dos coeficientes do polinômio interpolador. Devido ao método de interpolação escolhido, o algoritmo sistólico resulta naturalmente já que o cálculo das diferenças simples de ordem i dependem apenas daquelas de

ordem i-1. Ordenando em forma de tabela, percebe-se isto claramente:

x_i	y_i	D_0	D_1	D_2	D_3
x_0	y_0	$\mathbf{D_{00}=y_0}$			
			$\mathbf{D_{10}=D_{01}-D_{00}}$		
x_1	y_1	$D_{01}=y_1$		$\mathbf{D_{20}=D_{11}-D_{10}}$	
			$D_{11}=D_{02}-D_{01}$		$\mathbf{D_{30}=D_{21}-D_{20}}$
x_2	y_2	$D_{02}=y_2$		$D_{21}=D_{12}-D_{11}$	
			$D_{12}=D_{03}-D_{02}$		
x_3	y_3	$D_{03}=y_3$			

Onde: $D_{ab} = D_a(x_b)$, i.e., diferença simples de ordem **a** no ponto $\mathbf{x_b}$.

Os valores $\mathbf{D_{i0}}$ (em negrito) são aqueles utilizados diretamente para o cálculo dos coeficientes do polinômio interpolador e que deverão portanto serem fornecidos pelo sistema sistólico ao próximo estágio do algoritmo.

Além do cálculo das diferenças simples é necessário também calcularem-se os fatores

$$P_i(x) = \frac{1}{i!}\prod_{j=0}^{i-1}(x-j) \qquad (4)$$

que multiplicam as diferenças D_i. Este segundo cálculo, juntamente com a multiplicação pelas diferenças simples D_i, serão efetuados por outra parte do sistema sistólico como se verá a seguir.

Utilizando os fatores $P_i(x)$, o polinômio interpolador na forma expandida pode ser representado por:

$$p(x) = D_0.P_0(x) + D_1.P_1(x) + D_2.P_2(x) + D_3.P_3(x) \qquad (5)$$

4.2 A arquitetura sistólica

Na fig. 2 tem-se uma representação da arquitetura do sistema sistólico, dividida em duas partes: uma para o cálculo dos D_i e outra para o cálculo dos fatores $P_i(x)$.

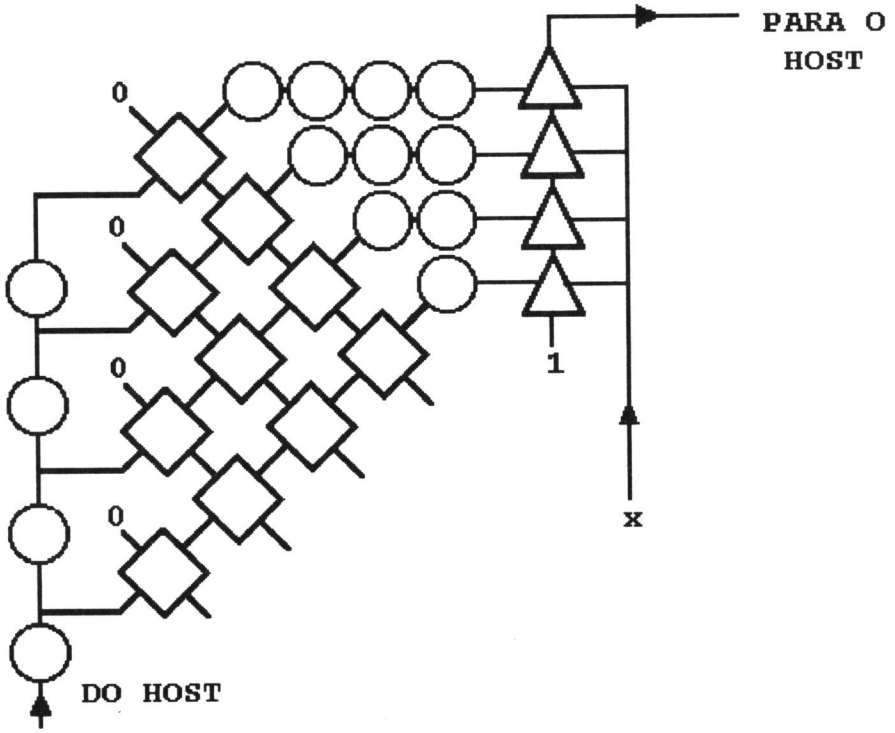


Fig. 2 - A Arquitetura Sistólica

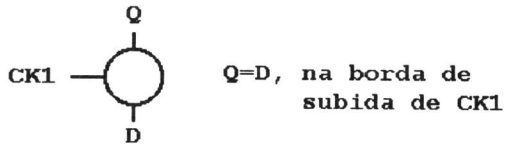
As células representadas por círculos na fig. 2 não realizam nenhum tipo de processamento mas apenas servem como registradores que armazenam os dados de entrada e saída.

Os losangos da figura representam processadores cuja única função é realizar a subtração dos valores em suas entradas, apresentando o resultado desta operação às suas saídas.

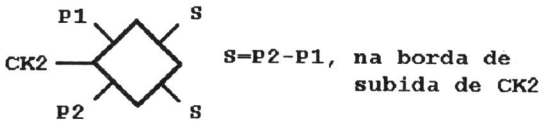
Células representadas por triângulos têm, neste sistema, a função mais complicada: devem realizar uma subtração, uma divisão e multiplicações sobre seus dados de entrada.

Todas as células são controladas por pulsos de relógio com fases bem estabelecidas (não estão representados na fig. 2). O funcionamento detalhado destas células será examinado a seguir.

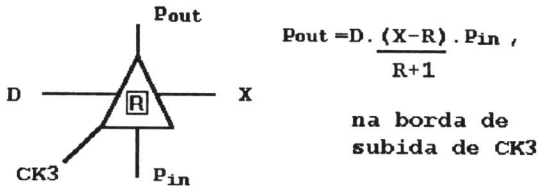
A **célula círculo** consiste de um registrador de 9 bits (8 bits mais sinal) que implementa-se facilmente como "flip-flops" tipo D. Veja-se a fig. 3a. A entrada D e a saída Q são ligadas em castata célula a célula de modo que os dados serão deslocados de uma célula a outra. Este aspecto permitirá que os valores entrem no "chip" um a cada vez e não os quatro em paralelo como era de se esperar, reduzindo o número de pinos no encapsulamento.



a) A célula círculo



b) A célula losango



c) A célula triângulo

Fig. 3 - As operações realizadas pelas células

A **célula losango** realiza a subtração indicada na fig. 3b. Seu projeto também é simples pois esta operação é feita sobre números inteiros.

Os processadores representados por **triângulos** (fig. 3c) devem manter armazenados internamente um valor que será subtraído de suas entradas X e que será usado para dividir o resultado da multiplicação pelos valores das entradas P_{in} e D que é então enviado à saida P_{out}. A célula triângulo poderia ser simplificada deixando a subtração a cargo de células losango ou completamente substituida pelo computador hospedeiro caso em que o sistema sistólico só calcularia as diferenças simples D_i.

5. Funcionamento do Sistema

Após o sistema entrar em regime, fornece resultados continuamente.

Se utilizarem-se sempre 4 pontos para a interpolação (dois à esquerda ou acima e dois à direita ou abaixo do ponto a interpolar) o cálculo de um polinômio utilizará 3 pontos já usado no cálculo do polinômio anterior e apenas um novo ponto. Assim, os valores dos pontos podem entrar um a um no sistema, interpolando-se um novo valor a cada ciclo.

Suponha-se os quatro pontos para o cálculo de um polinômio já posicionados nas quatro células círculo de entrada (fig. 2). No sub-ciclo seguinte, serão realizadas 10 subtrações em paralelo pelas células losango. As quatro subtrações correspondentes à primeira coluna de células losango é realizada apenas por questões de simetria já que o subtraendo é sempre zero.

No sub-ciclo a seguir, estes resultados são registrados pelas células círculo de saída e deslocados para a direita simultaneamente nas quatro linhas de células círculo. A última coluna destas células (mais à direita), envia seus dados às células triângulo que, lendo o valor x a interpolar, calculam parcelas dos produtos Pi(x).

O sub-ciclo seguinte é composto de 4 pulsos que deslocam os resultados das células triângulo para cima, saindo do sistema. Nesta operação, é realizada a multiplicação final dos fatores, originando termos que somados resultam no valor interpolado.

A fig. 3 indica as operações realizadas por cada célula. As fases dos "clocks" podem ser facilmente estabelecidas.

6. Conclusão

O sistema sistólico apresentado neste trabalho apresenta características que pretendem caracterizá-lo como um bom exemplo deste tipo de arquitetura:

- células simples e de poucos tipos;
- comunicação local e síncrona;

- apresenta regularidade, modularidade e expansibilidade;
- apresenta paralelismo e "pipelining";
- permite ajuste de banda de E/S;
- encapsulamento com poucos pinos;
- resolve um problema específico e de alta demanda de processamento.

Apesar do sucesso dos sistemas sistólicos na solução de determinados problemas, sistemas de uso específico ainda deparam-se com um mercado muito restrito, só encontrando financiamento geralmente junto a universidades e entidades militares. Este trabalho procurou explorar mais uma aplicação desta filosofia, enriquecendo a coleção de soluções sistólicas para aplicações em processamento de imagens.

Agradecimentos

Este trabalho foi desenvolvido no âmbito do curso de pós-graduação "MAC740 - Algoritmos Paralelos e Arquiteturas VLSI" ministrado na Universidade de São Paulo pelo prof. Dr. Siang Wun Song a quem o autor agradece pelo incentivo. O autor recebia uma bolsa de mestrado concedida pelo CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico, quando da realização deste trabalho.

Referências

[1] Kung, H.T. ; Leiserson, C.E., "Systolic Arrays for VLSI", in: Duff, I.S. ; Stewart, G.W. (ed.), **Sparse Matrix Proceedings 1978**, SIAM, p.256-82 ,1979.

[2] Song, S.W., "Algoritmos Paralelos e Arquitetura VLSI", **IV Escola de Computação**, IME-USP, 1984.

[3] Kung, H.T., "Why Systolic Architectures ?", **IEEE COMPUTER**, v.15, n.1, p.37-46, Nov. 1982.

[4] Mead, C. ; Conway, L., **"Introduction to VLSI Systems"**, Addison-Wesley Publishing Co., 1980.

[5] Kung, H.T. ; Picard, R.L., "Hardware Pipelines for Multi-Dimensional Convolution and Resampling", **Proceedings of the 1981 IEEE Computer Society Workshop on Computer Architecture for Pattern Analysis and Image Database Management**, IEEE Computer Society Press, Nov. 1981, p.273-8.

[6] Hou, H.S. ; Harry C. Andrews, "Cubic Splines for Image Interpolation and Digital Filtering", **IEEE Trans. on Acoustics, Speech and Signal Processing**, v. ASSP-26, n.6, December 1978, pp. 508-512.

[7] Araújo, L.J. ; Santos, E. T., **"Rotina de Aumento de Resolução de Imagens por Interpolação Polinomial"** - Relatório Interno - LSI/Grupo Gráfico, 1989.

[8] Humes, A.F. et al., **"Noções de Cálculo Numérico"**, McGraw-Hill, 1984.