UM PROCESSADOR DEDICADO PARA DETEÇÃO DE BORDAS

José Eduardo Cogo Castanho
Clésio Luis Tozzi

Departamento de Computação e Automação. Faculdade de Engenharia Elétrica. Universidade Estadual de Campinas. Caixa P. 6101. Cidade Universitária. CEP 13081. Campinas. SP. email: CLESIO@BRUC.ANSP.BR.

Resumo: A visão computacional exige grande volume de processamento. Este artigo aborda a minimização do esforço computacional pela uso de circuitos processadores especializados em aplicações específicas. Como exemplo, é apresentado um circuito processador para deteção de bordas de imagens em tempo real.

1 - INTRODUÇÃO

Uma característica própria da visão computacional é o grande volume de processamento exigido. Contribuem para isso a grande quantidade de dados presentes desde o início do processo e também o tipo das operações aplicadas sobre os mesmos [1]. Os processos costumam ser, dessa maneira, extremamente morosos quando são utilizados processadores convencionais, dificultando por exemplo aplicações que necessitem respostas rápidas ou operação em tempo real.

Para as aplicações e algoritmos hoje existentes, a solução mais comum para a aceleração do processamento consiste em empregar sistemas computacionais com grande capacidade e em geral baseados em arquiteturas paralelas. Entretanto, para aplicações específicas é interessante o emprego de circuitos processadores algoritmicamente dedicados [2],[3],[4].

Este trabalho apresenta uma proposta para implementação de um conjunto de processadores em pipeline para realização de algumas etapas do processamento de baixo nível em visão computacional. Um primeiro processador para extração de bordas de objetos em uma imagem multitonal é apresentado. O objetivo é acelerar as etapas de baixo nível do processamento utilizado para o reconhecimento de padrões em aplicações industriais (fig.1). A obtenção das

bordas deve ser realizada em tempo real, ou seja, no tempo de aquisição de uma imagem. As operações de alto nível podem então ser executadas em um microcomputador, após a redução do volume de informação pelo processador pipeline dedicado. Apesar da proposta apresentada estar voltada diretamente para a solução da etapa de deteção de bordas, os mesmos princípios podem ser extendidos para as outras etapas de baixo nível.

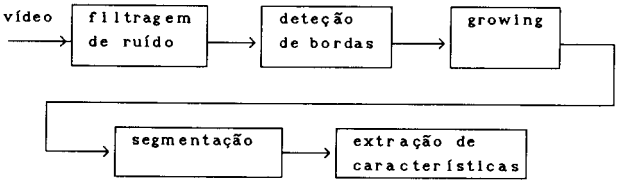

fig.1 - Etapas do processo de reconhecimento de padrões.

2 - A RELAÇÃO HARDWARE-ALGORITMO.

A deteção de bordas através dos operadores de Sobel envolve as etapas mostradas no diagrama da fig.2. Estas etapas podem ser implementadas diretamente em hardware dedicado com a mesma organização apresentada pelo diagrama, produzindo então uma estrutura pipeline. O fluxograma da fig.2 é dividido em quatro etapas que devem ser executadas sequencialmente. A convolução com núcleo de dimensão 3x3, é realizada em duas etapas paralelas, para deteção das bordas verticais e horizontais, assim como a extração do módulo dos resultados da convolução. Os resultados das duas convoluções são combinados através de uma soma e finalmente é feita a comparação dos resultados com um valor pré-determinado que define se aquela variação de luminosidade determina uma borda ou não. Após a obtenção das bordas a imagem torna-se binária, com os pixels que determinam bordas recebendo valor "1" e pixels que não representam borda recebendo valor "0". Assim, o volume de dados é sensivelmente reduzido pois cada pixel pode ser representado por um único bit.

Cada uma das etapas da deteção de bordas pode ser implementada através de circuitos dedicados, formando um circuito pipeline. As etapas de módulo, soma

e "thresholding" podem ser, cada uma, implementadas por exemplo utilizando "look-up tables". A operação de convolução, entretanto, é constituída de várias sub-operações que devem ser executadas em paralelo para permitir a manutenção do fluxo de dados para os outros estágios. Ainda, algumas das sub-operações da convolução podem ser executadas por um pipeline, alongando o número de estágios total.

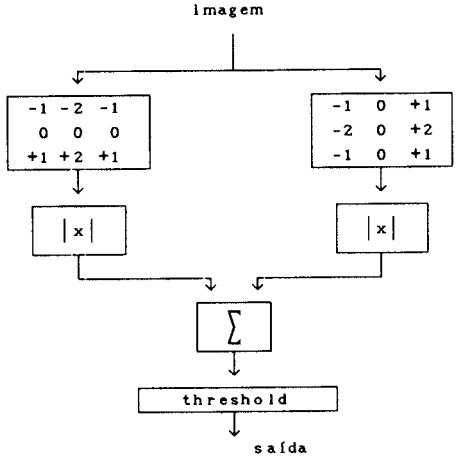

fig.2 -Fluxograma da aplicação do operador deSobelpara deteção de bordas de uma imagem.

A verificação do algoritmo para execução da convolução com o núcleo 3X3 mostra que as únicas operações funcionais envolvidas são soma e multiplicação. O restante das operações corresponde ao acesso aos dados de imagem de entrada, o armazenamento da imagem resultado e ao controle da estrutura do programa.

Para a execução do algoritmo de deteção de bordas em tempo de aquisição de imagem, isto é, em 30ms, um processador deve operar na própria taxa de vídeo, ou seja, o tempo de produção de um novo resultado (pixel), considerando uma imagem de 512x512 pontos, deverá ser de aproximadamente 100ns. Para a obtenção de um pixel em 100ns, o processador deve realizar 9 multiplicações e 9 somas nesse período, além de buscar os dados na memória. Supondo as operações de soma e multiplicação sendo executadas em pipeline, o ciclo de execução do processador deve ser de 100/9=11,1ns. Evidentemente, tanto o tempo de acesso à memória como o tempo de execução são muito reduzidos, tornando difícil a implementação de um processador como esse. A solução do problema só

é alcançada com a exploração de paralelismo de programa que pode ser feito pelo uso de replicação de processadores ou unidades funcionais.

3 - O PROCESSADOR PARA CONVOLUÇÃO

A representação da imagem como uma função espacial e seu armazenamento em uma memória conduz a diversas possibilidades de implementação de processadores para a execução da convolução.

O diagrama em blocos do módulo básico de um processador dedicado para a convolução é mostrado na fig.3. O fluxo de dados de entrada é equivalente ao fornecido por uma interface de aquisição de imagens de uma câmara de vídeo. Consequentemente os pixels devem ser inseridos na mesma ordem em que é feita a varredura de uma tela (sem entrelaçamento). Entretanto, cada linha é inserida três vezes no processador para permitir a obtenção dos resultados. A presença de um buffer permite, ainda, algumas alternativas na estrutura do sistema pois o fluxo de dados não está amarrado, podendo-se acessar qualquer ponto da imagem a qualquer momento.

Os registradores da entrada têm a função de alinhar os pixels na entrada dos multiplicadores. O estágio multiplicador realiza a multiplicação simultânea de cada um dos três pixels que entram neste estágio pelos respectivos coeficientes de uma linha do núcleo de convolução. Os coeficientes são trocados sempre que se está processando uma nova linha. Os somadores realizam a somatória presente na formulação da convolução. A somatória é feita entre os valores que entram no estágio e o conteúdo dos registradores da fifo. Cada endereço da fifo armazena um resultado parcial que corresponde a soma dos três primeiros produtos de pixel pelos coeficientes e após o processamento da segunda linha corresponde a soma de seis produtos. O resultado final obtido após o processamento da terceira linha não é armazenada na fifo e deve ser armazenado na memória principal. A fifo tem a função de armazenar os resultados intermediários obtidos a partir da computação parcial de cada linha de convolução. Possui comprimento igual ao da linha de imagem menos o número de estágios para a implementação do somador.

Este processador básico necessita operar a uma frequência de aproximadamente 30Mhz para garantir um pixel resultado a cada 100ns. É interessante, entretanto, operar a frequências ainda mais baixas a fim de permitir maior facilidade de implementação.

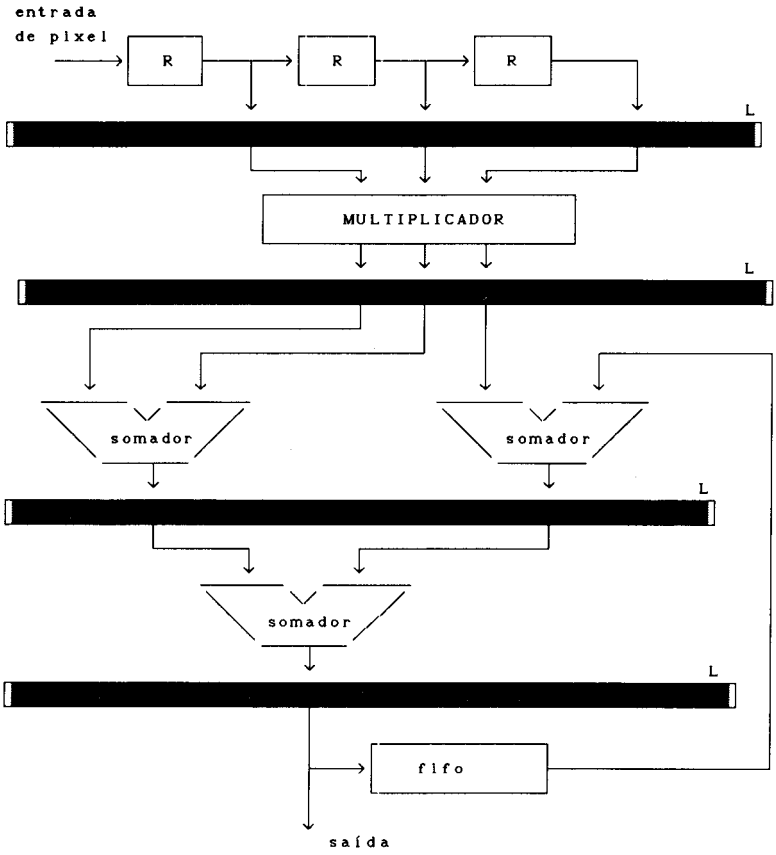

fig.3 - Estrutura básica de um processador pipeline para o operador 3X3.

Pode-se verificar facilmente que o emprego de três desses processadores (ou mais) em paralelo permite atingir o desempenho proposto operando a uma frequência de 10MHz. Para viabilizar essa proposta é necessário distribuir adequadamente dados de imagem entre os processadores. Essa solução só é possível no caso da imagem estar armazenada em um buffer.

A adoção de um buffer intermediário entre a interface de entrada e o processador possibilita o acesso a vários locais da tela simultaneamente. Evidentemente, o acesso à memória deve ser realizado através de um barramento que permita uma alta taxa de transferência, evitando conflitos no barramento

[5],[6]. Quanto maior for o número de processadores menor será a frequência de operação, entretanto, os problemas de conflito no barramento tendem a se agravar.

A divisão da imagem em três faixas horizontais, uma para cada processador permite atingir o desempenho desejado. Neste caso, a única alteração necessária deverá ser feita no fluxo de controle, com a diminuição das dimensões originais da imagem para cada processador. Assim, as dimensões da imagem para cada processador, originalmente nxm, passam a ser nx(m/k), onde k é número de processadores usados. Esta abordagem possui como vantagem o fato de se trabalhar com diversos processadores, não necessariamente sincronizados, apenas utilizando um árbitro para acesso ao barramento.

Pode-se adotar também, uma divisão por faixas verticais que, entretanto, forçará um alteração nas dimensões da fifo, bem como no fluxo de controle. Ou pode-se dividir a imagem em janelas para cada um dos processadores, com as mesmas consequências.

Uma outra alternativa leva em conta uma atuação sincronizada de três processadores, cada qual atuando sobre uma etapa diferente do processo, mas sobre os mesmos dados. Cada processador irá produzir resultados de linhas diferentes e intercaladas. Em outras palavras, num determinado instante, com a mesma linha da imagem como entrada, um processador estará multiplicando a primeira linha do núcleo, outro a segunda linha do núcleo e o último a terceira linha do núcleo. Consequentemente, o processador que está utilizando a terceira linha do núcleo está também produzindo um resultado. Além disso, a cada nova linha de entrada, os papéis de cada processador são trocados entre si, isto é, o processador que processava a primeira linha do núcleo passa a processar a segunda, o processador que processava a segunda linha do núcleo passa a processar a terceira e o processador que processava a terceira linha do núcleo passa a processar a primeira, num processo sequencial e cíclico.

Num processo evolutivo é possível derivar um novo processador a partir deste arranjo com três processadores sincronizados, simplificando de uma maneira geral o circuito e o controle envolvidos.

A idéia principal embutida no processo de obtenção deste novo processador consiste na especialização de cada uma das partes. Assim, da estrutura anterior nota-se que cada processador executa intercaladamente as mesmas funções que os outros dois. No caso, a função referida é a multiplicação de três pixels de imagem por três coeficientes do núcleo. A cada novo ciclo de relógio, cada um dos processadores tem como coeficientes para multiplicação uma nova linha do núcleo. É possível simplificar o circuito se cada

processador realizar sempre multiplicações por um mesmo coeficiente.

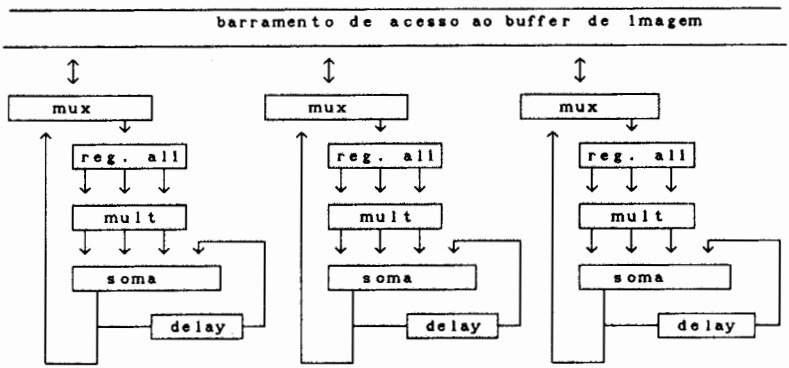

fig.4 - Três processadores intercalados.

Entretanto, torna-se agora necessário, que cada processador, após multiplicar um grupo de coeficientes de um linha de imagem por um linha do núcleo, comunique o resultado para o processador seguinte. Este por sua vez, efetuará a mesma operação com os pixels da linha seguinte, somará este resultado com o valor passado pelo processador anterior e então transfere a soma para o processador seguinte. O terceiro processador realiza então as mesmas operações com a terceira linha, obtendo o resultado final após somar o resultado obtido nele com o resultado proveniente do segundo processador. Como a operação sobre cada pixel de uma mesma coluna mas linhas diferentes ocorre após um atraso de 512 ciclos de relógio, a comunicação entre dois processadores adjacentes deve possuir uma linha de atraso correspondente para que as operações entre os processadores sejam sincronizadas corretamente. Esse atraso é efetuado pela mesma fifo do processador básico proposto, onde a realimentação para um estágio anterior é substituída pela conexão com o processador seguinte. Na fig.5 é apresentado o diagrama completo deste novo processador.

Verifica-se facilmente que esta configuração para o processador permite as seguintes simplificações: o número de fifos foi reduzido de 3 para 2; a largura da fifo é reduzida pois a largura dos operandos é menor nas etapas iniciais do processamento; o número de registradores de alinhamento é reduzido de 3 para 1, tanto na entrada como na saída; os multiplicadores podem ser

realizados com ponderadores; o controle torna-se mais simples pois não é necessário efetuar nenhuma troca de função e o pipeline é totalmente linear.

Esta configuração permite ainda que o processamento seja realizado em tempo de aquisição.

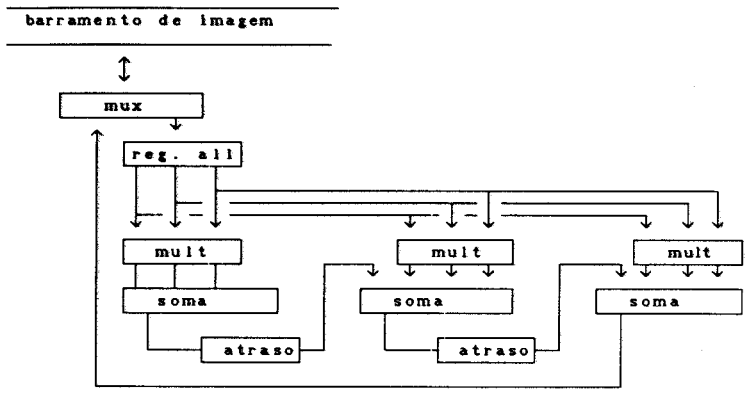

fig.5 - Processador 3x3.

Uma outra alternativa para a elaboração do processador pode ser obtida através da representação da imagem convoluída como função do tempo, como representado abaixo:

$$S(t)= K_8.I(t) + K_7.I(t-1) + K_6.I(t-2) + K_5.I(t-n) + K_4.I(t-n-1) + K_3.I(t-n-2) + K_2.I(t-2n) + K_1.I(t-2n-1) + K_0.I(t-2n-2)$$

onde S(t) é a função saída da convolução, I(t) é a função imagem, K_n são os coeficientes do núcleo, t representa o tempo e n é o comprimento da linha de imagem e representa uma atraso no sinal de entrada.

Esta forma de representar a convolução permite a visualização de um circuito processador (fig.6) com um atraso na entrada dos dados ao invés de acumular os resultados de somas parciais como sugere a representação espacial. Dessa forma é realizado um alinhamento de três linhas consecutivas da imagem de forma que os nove pixels a serem multiplicados pelo núcleo estejam presentes simultaneamente. A soma dos nove produtos pode então ser realizada

em um único passo. Esta abordagem também permite a operação de convolução em tempo de aquisição de imagem, além da redução da largura das fifos.

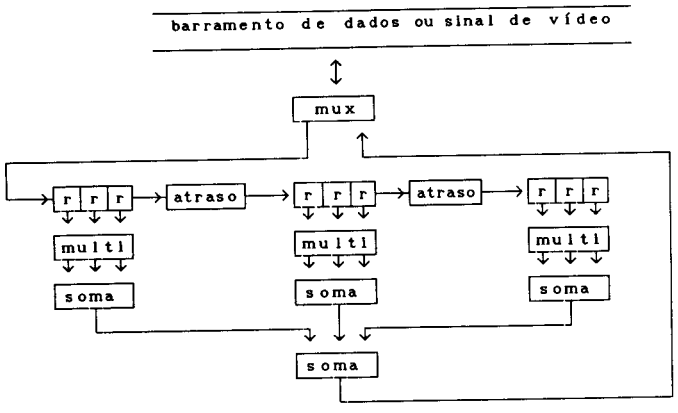

fig.6 - Processador 3x3 com atraso dos dados de entrada.

Este circuito, se considerado a repetibilidade de funções e simplificação de cada uma delas através de especialização é adequado a implementação em circuitos VLSI. Esta implementação está sendo desenvolvida por Costa [7]

4 - Conclusões

A utilização de um circuito dedicado para desempenho de funções específicas permitiu a obtenção de alto desempenho a um custo baixo, inclusive com o emprego de paralelismo de operações. Além da deteção de bordas, pode-se empregar circuitos dedicados para execução das outras etapas da visão computacional otimizando ainda mais o processo.

O processador para execução do núcleo de convolução 3x3 pode ser usado para qualquer operação envolvendo tal tipo de operador. A associação de diversos destes processadores em pipeline, cada qual executando operações distintas, permite a acelerar de forma substancial a velocidade de processamento de um sistema voltado para o reconhecimento de padrões.

Sem muitas dificuldades, apenas pela extensão dos conceitos apresentados, é possível obter novos processadores para núcleos de convolução de maior

dimensão, aumentando ainda mais as áreas de aplicação. Uma outra possibilidade bastante interessante é a modificação do circuito básico de maneira que seja possível a associação de vários circuitos básicos para o processamento de um núcleo maior.

BIBLIOGRAFIA:

[1] - Rosenfeld. A. "Computer Vision: Basic Principles". Proceedings of the IEEE, vol 76, no. 8, August 1988.

[2] - Pratt, William K. "Algorithmic-Based Machine-Vision Computing". Digital Design. October 25, 1986.

[3] - Ruetz, Peter A. and Robert W. Brodersen. "An Image-Recognition Sistem Using Algorithmically Dedicated Integrated Circuits". Machine Vision and Applications. January, 1988.

[4] - Bursky, Dave. "CMOS four-chip set processes images at 20MHz date rates". Eletronic Design. May 28, 1987.

[5] - Castanho, J. Eduardo C. "Uma Estação de Trabalho para Visão Computacional". Tese de mestrado. Faculdade de Engenharia Elétrica. Unicamp. Campinas.1990.

[6] - Castanho, J. Eduardo C. "Um sistema para processamento de imagem".I SIBGRAPI. Petrópolis.1988

[7] - Costa, Henrique Sérgio G. da. "Processador Pipeline de Imagem - CI.01". Relatório Técnico nº 036/89 Faculdade de Engenharia Elétrica de Campinas - UNICAMP. Campinas 1989.

Este trabalho recebeu auxílio da SID INFORMÁTICA através do projeto ESTRA