

Reconhecedor de Imagens Usando Técnicas de Alto Desempenho*

Eduardo Moschetta,[†] Arlon Zimmer da Cunha,[‡]
Fernando Santos Osório, Gerson Geraldo H. Cavalheiro

Programa de Pós-Graduação em Computação Aplicada – PIPCA
Centro de Ciências Exatas e Tecnológicas
Universidade do Vale do Rio dos Sinos
São Leopoldo – RS – Brasil
{eduardom, arlon, osorio, gersonc}@exatas.unisinos.br

Resumo

Trabalhar com imagens requer uma grande quantidade de processamento. No caso do reconhecimento de imagens, é feita uma consulta a um banco de imagens, onde tenta-se casar um padrão com cada imagem desse banco. Assim, comparando-se um padrão com todos os blocos possíveis de cada imagem, gera-se uma carga ainda maior de processamento. Esse artigo apresenta uma solução concorrente para esse problema usufruindo-se de arquiteturas paralelas e distribuídas.

Palavras chaves: busca de imagens, programação concorrente, alto desempenho.

Introdução

Nas atuais tecnologias de computação, sistemas multimídia, em especial tratando imagem e vídeo, estão cada vez mais presentes em diversas aplicações. Neste contexto, um problema real é, a partir de uma coleção de imagens, localizar uma imagem específica [DAS 97, SMI 95]. Este trabalho aborda o reconhecimento de imagens, especificamente o casamento de uma imagem com uma região de outra imagem, que pode (ou não) conter a primeira. O objetivo é oferecer um sistema de consultas a bancos de dados de imagens a partir de imagens-exemplo (*Query by Image*), ou ainda, um sistema de recuperação de imagens a partir desse banco. O princípio básico é percorrer todas as imagens do banco de imagens e, a cada imagem, procurar pelo padrão proposto.

Existe uma grande demanda por sistemas que buscam informações num grande conjunto de dados. Como exemplo, temos os sites de busca da Web que procuram algum conjunto de palavras, fornecido pelo usuário, em uma coleção de textos da Internet. Assim, como esse conjunto de palavras pode ser considerado como um padrão textual nesse sistema, podemos querer encontrar um padrão de imagem em um conjunto de imagens. É exatamente esse sistema de busca de imagens que o reconhecedor citado implementa.

Neste trabalho é apresentada uma solução concorrente a este problema, onde a carga total de processamento, busca do padrão, é compartilhada entre diferentes nodos de um agregado. O restante deste trabalho discute o algoritmo utilizado, as técnicas de balanceamento de carga aplicadas e a implementação realizada.

* Apoio: CNPq, FAPERGS, UNISINOS

[†] PIBIC - CNPq

[‡] ITI - CNPq

Método de Procura de uma Subimagem

Para o algoritmo iniciar a procura de uma imagem parcial, são necessárias duas informações: a própria imagem parcial (o *padrão* a ser procurado) e um banco de imagens. Ao terminar a procura, o algoritmo retorna a imagem do banco com a qual o padrão teve o melhor *matching* (casamento ou grau de similaridade). O cálculo do *matching* pode ser realizado de diferentes formas. De uma maneira simplificada, assume-se que o padrão e as imagens do banco de imagens estejam na mesma escala. Assim, o *matching* se resume numa comparação pixel a pixel entre essas.

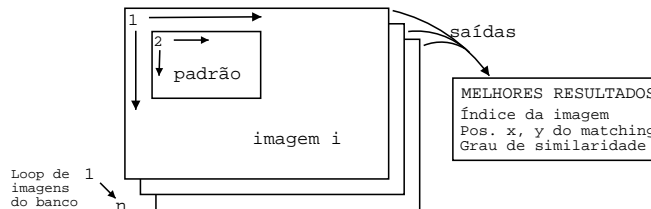


Figura 1: Algoritmo de *Matching*

O algoritmo de procura propriamente dito, é esquematizado na Figura 1. Ele consiste de um loop externo que percorre todas as n imagens do banco de imagens. A cada iteração, (1) é deslocado o *padrão* (subimagem), pixel a pixel, buscando-se o casamento entre esse e uma região da imagem i (movimento externo). Ainda, dentro de cada movimento externo, (2) precisa-se comparar todos os pixels da subimagem com os pixels do bloco sobreposto na imagem i (movimento interno). O resultado do movimento interno serve para determinar a similaridade entre o padrão e a região pesquisada na imagem i .

Métodos de Escalonamento Aplicados

A carga de trabalho é representada pela busca da subimagem em uma imagem do banco de imagens. Assim sendo, a distribuição do banco de imagens entre os nodos tem papel fundamental na distribuição da carga total.

Escalonamento Entre-nós

Considerando que cada nó possui um subconjunto das imagens como banco de imagens, cada nó pode executar suas tarefas de forma independente. Assim, no início da pesquisa, o algoritmo de procura cria um processo independente para cada nó, sendo o banco de dados dividido igualmente entre os nodos.

À medida que nodos são capazes de processar mais rapidamente que outros, a carga de trabalho é balanceada dinamicamente (migração de imagens do banco de um nó para outro). Isso é feito quando um nó termina de percorrer seu subconjunto de imagens, buscando imagens no banco do nodo que mais possui imagens não processadas na procura pelo padrão.

Escalonamento Intra-nó

Para cada imagem são criadas n cabeças de leitura, as quais percorrem a imagem de forma concorrente buscando o *matching* entre o padrão e uma região da imagem. Nesse caso, a cabeça de leitura é um procedimento, representado por uma *thread*, que se assemelha ao movimento externo (Figura 1). Porém, ela percorre apenas uma linha da imagem, a qual serve de base superior para a comparação, retornando o melhor resultado encontrado nessa linha. Todo esse procedimento é aplicado para cada imagem contida na parcela do banco de imagens a qual o nó é responsável.

O objetivo do uso de *threads* a este nível é duplo [CAV 01]: por um lado, busca o aumento da disponibilidade dos processadores ao cálculo (em especial se a arquitetura do nó for SMP); de outro, visa sobrepor com cálculo efetivo parte das perdas ocorridas durante a comunicação entre nodos, ou seja, quando há a necessidade de sincronização entre os processos desses nodos para realizar o balanceamento de carga.

Implementação

Na implementação realizada, o banco de imagens encontra-se em um diretório acessível a todos os nodos. No início da procura, esse é dividido em listas de imagens de mesmo tamanho, que vão estar disponíveis na memória local de cada nó. Conseqüentemente, cria-se um processo para cada nó que é responsável pela procura de um padrão em seu subconjunto de imagens gerado. Dentro de cada um desses processos, são criadas várias cabeças de leitura (*threads*) para cada imagem do seu subconjunto de imagens.

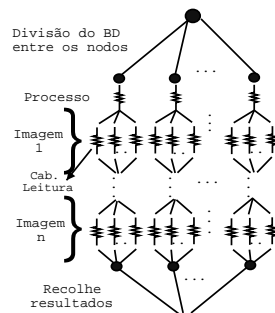


Figura 2: Modelo de Implementação

A sincronização entre as cabeças de leitura ocorre na passagem de uma imagem para outra do banco de imagens, a fim de que se colete os melhores resultados achados na mesma. Do mesmo modo, é necessário haver sincronização entre os processos criados em cada nó, para comparar os melhores resultados entre os diferentes subconjuntos de imagens. Esse modelo de implementação pode ser melhor visualizado na Figura 2.

Como ferramentas de implementação, foi utilizado MPI (*Message Passing Interface*), para comunicação inter-processos, e a biblioteca Pthreads, para o *multithreading*.

O cálculo da similaridade compara o padrão com uma região de uma das imagens do banco, podendo empregar diferentes algoritmos (comparação de histogramas, cor e textura, entre outros). Um algoritmo simplificado para obter-se a similaridade entre as imagens pode considerar apenas a soma dos valores absolutos das diferenças entre as cores (cada cor possuindo um valor correspondente) dos pixels comparados [JAI 95], através da fórmula: $soma = soma + |COR_{pixel\ do\ padrão} - COR_{pixel\ da\ imagem\ do\ BD}|$.

Quanto ao tipo de imagem, convencionou-se o uso de imagens com formato .PNG, pois nesse formato há pouca perda de qualidade nos diferentes níveis de compactação de imagens. Como esse formato utiliza o padrão de cores RGB, esse foi usado como referência para determinar o valor da cor de cada pixel [GOM 94].

Conclusão

O uso de uma arquitetura paralela e distribuída no reconhecimento de imagens demonstrou que este problema pode ser incluído na classe dos "problemas trivialmente paralelizáveis" [LEM 02]. Isto é fato, uma vez que no processamento as imagens independem uma das outras.

O uso de uma solução concorrente aplicada ao problema de busca de informações (*Query by Image*) permitiu viabilizar o processamento de grandes bases de dados de imagens. Com isto, esse sistema torna-se uma ferramenta poderosa em aplicações de multimídia e mesmo de visão artificial em problemas de robótica móvel. Os próximos passos a serem trabalhados deverão envolver técnicas mais elaboradas tanto para o casamento de imagens quanto para a distribuição de carga.

Referências

- [CAV 01] CAVALHEIRO, G. G. H. Introdução à programação paralela e distribuída. In: *I Escola Regional de Alto Desempenho*, Gramado. **Anais...**, 2001.
- [DAS 97] DAS, M.; RISEMAN, E.; DRAPER, B. Focus: searching for multi-colored objects in a diverse image database. In: *IEEE Conf. on Comp. Vis. and Pattern Recognition*, Porto Rico. **Anais...**, 1997.
- [GOM 94] GOMES, J.; VELHO, L. **Computação gráfica: imagem**. Rio de Janeiro, Brasil: IMPA/SBM, 1994.
- [JAI 95] JAIN, R.; KASTURI, R.; SCHUNCK, B. G. **Machine vision**. Singapore: McGraw-Hill International Editions, 1995.
- [LEM 02] LEMKE, N. Problemas trivialmente paralelizáveis. In: *II Escola Regional de Alto Desempenho*, São Leopoldo. **Anais...**, 2002.
- [SMI 95] SMITH, J. R.; CHANG, S.-F. **Tr: automated image retrieval using color and texture**. Columbia University - Technical Report, 1995. (414-95-20).