

Um Algoritmo para Reconhecimento de Objetos Baseado em Regiões

Paulo Sérgio Rodrigues

Arnaldo de Albuquerque Araújo

*Departamento de Ciência da Computação
DCC - ICEx - UFMG. Caixa Postal 702.
30161-970. Belo Horizonte, MG - BRAZIL
Phone: +55 31 499-5860 Fax: +55 31 499-5858*

Resumo

Este artigo apresenta um novo algoritmo, batizado de *Graph Region Arrow Shot*, para reconhecimento de objetos baseado em características globais da imagem. Essas características são diversas e, portanto, podem ser variadas; são tomadas durante a varredura do algoritmo e representam a estrutura morfológica global de objetos em imagens, portanto, capturam elementos que não são capturados por algoritmos tradicionais de reconhecimento. O algoritmo pode funcionar sob condições restritas como um banco de dados pequeno ou um segmentador simples, que não consiga gerar imagens com regiões corretamente conectadas. O algoritmo foi testado para um pequeno conjunto de imagens de objetos, mas com uma grande variação de imagens que representam diversas perspectivas de cada um, visões do objeto.

1 Introdução

A palavra “reconhecimento”, por si só, é motivo de polêmica. O que pode ser reconhecido como sendo um determinado objeto por uma pessoa pode ser reconhecido como algo totalmente diferente por outra [21]; e o que é pior, dependendo do contexto, uma mesma pessoa pode dar inúmeras interpretações para a mesma imagem [2, 10, 15]. No entanto, são diversos os trabalhos que tratam da tarefa de reconhecimento automático de objetos e cenas [1, 6, 9, 12, 22]. Há mais de duas décadas, essa palavra tem sido usada largamente na literatura para significar entendimento, interpretação, classificação, cognição e outras tarefas inerentemente humanas. Com o desenvolvimento de áreas de visão computacional, psicologia, neurociências e inteligência artificial, sabe-se hoje em dia que a tarefa de reconhecimento automático de objetos, como uma “imitação” do sistema visual humano,

ainda está longe de ser alcançada, e muito trabalho ainda deve ser demandado [4].

O que se tem atualmente são sistemas especialistas com objetivos específicos e que, na maioria das vezes, falham em reconhecer objetos de uma maneira geral, mesmo considerando o seu próprio universo de atuação [5, 7]. Para essa tarefa, diversas técnicas surgiram ao longo das últimas três décadas [20, 19, 14, 16, 17, 11, 8, 13, 18]. Tem-se, entretanto, a grande maioria dos trabalhos como sendo uma combinação de algumas delas [21]. Entre as diversas técnicas utilizadas, sem dúvida as mais exploradas são aquelas que usam informações de forma para o reconhecimento. Este fato se deve também à crença entre os psicologistas de que, embora o sistema visual biológico possa se valer de diferentes parâmetros para atingir seus objetivos (p.ex: cor, forma, movimento, contexto, som, textura, etc.), o mais utilizado é, sem dúvida, a forma. A forma de um objeto deve, no entanto, ser mais especificada. Pode-se definir forma simplesmente como as bordas externas, internas, disposição de umas em relação às outras, ou mesmo o conjunto de regiões que, no geral, semanticamente compõem a cena.

Muitos algoritmos funcionam em cima de uma única região, considerando apenas a borda externa e, no máximo, as bordas internas. Isso significa que, objetos compostos por mais de uma região não são considerados. Por outro lado, a esmagadora maioria dos métodos de reconhecimento necessitam de um pré-processamento da imagem antes do passo final de reconhecimento em si. E muitos desses pré-processamentos incluem uma fase de segmentação; e é sabido que segmentar uma imagem também é um processo cognitivo e, por esta razão, os algoritmos mais populares atualmente, ainda produzem uma saída

deficiente em muitos casos. Um exemplo típico acontece quando se obtém como resposta, não uma única região representando o objeto, mas um conjunto de regiões que, semanticamente, “deveriam” estar conectadas, Fig. 3. Entretanto, essas regiões mantêm uma estrutura morfológica global que pode ser usada para “reconhecimento”.

O presente trabalho introduz um novo algoritmo para reconhecimento de objetos baseado em estrutura morfológica. Ele captura a complexidade das estruturas globais que não são capturadas pelos algoritmos tradicionais. Esse algoritmo, que foi chamado de “Graph Region Arrow Shot” (GRAS), possui complexidade linear e obtém as características a partir de um conjunto de regiões geradas pelo segmentador. O nome foi escolhido porque as regiões vão sendo alcançadas de maneira análoga a uma estrutura de reservatórios ligados por tubulações que está sendo preenchida com água; além do que, cada reservatório vai sendo alcançado a partir de seus vizinhos mais próximos.

Para mostrar a validade do algoritmo, apresenta-se o conceito de visões de um objeto e é feita a comparação com algoritmos para reconhecimento baseados em Análise de Fourier, Análise dos Componentes Principais e Momentos Invariantes.

2 Conceitos Preliminares

Serão apresentadas nesta seção algumas definições e conceitos que irão ser usados ao longo deste artigo. Alguns já são usados na literatura.

Após o pré-processamento e segmentação da imagem, cada pixel com valor 0 é considerado “background”, e todo pixel com valor diferente de 0 é considerado como fazendo parte de algum objeto. Então, um objeto é um conjunto de pixels, conectados ou não, com o mesmo valor diferente de 0. Quando alguns pixels de um mesmo objeto estão conectados, diz-se que eles formam uma região. Logo, um objeto pode ser composto de uma ou mais regiões.

Imediatamente ao conceito de objeto, pode-se definir o conceito de cena. Uma cena é um conjunto de dois ou mais objetos com valores de pixels diferentes.

Neste trabalho, o conceito de visão é um pouco menos formal do que o conceito de objeto ou cena. Este conceito está diretamente ligado ao conceito de perspectiva. Um objeto pode ser gerado a partir da perspectiva de uma imagem. Cada perspectiva diferente chama-se “visão do

objeto”. Intuitivamente, o conjunto de visões representa o mesmo objeto. A idéia de visão, entretanto, não necessita, neste trabalho, de um formalismo maior, uma vez que só será utilizada para provar a efetividade do algoritmo proposto. A Fig. 1 ilustra este conceito. Nela, dezesseis imagens estão distribuídas em quatro por quatro linhas. Cada linha representa quatro visões (de acordo com o conceito de visão adotado aqui) diferentes de um tipo de objeto. Da primeira para a quarta linha, esses objetos são: *pen*, *eraser*, *sciss* e *vosg*. Neste trabalho, serão adotados 8 objetos distintos, cada um com um número entre 400 e 800 visões diferentes.

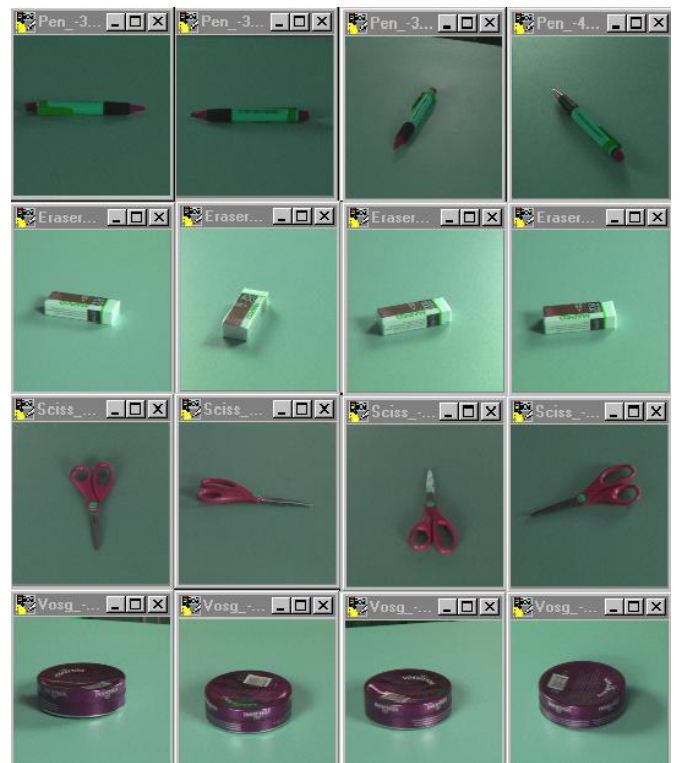


Figura 1: Um exemplo de 4 objetos diferentes, cada um com quatro visões distintas, mostradas na mesma linha. A linha um apresenta quatro visões de *pen*, a linha dois, *eraser*, a linha três, *sciss* e a linha quatro, *vosg*.

Finalmente, o conceito que será definido é o de “reconhecimento”. Embora essa palavra já seja usada intuitivamente há muito tempo na literatura, é necessário estabelecer o que significa realmente “reconhecer” neste trabalho. Quando um conjunto de regiões que representa um objeto é obtido, ele serve como entrada para o algoritmo GRAS, que então produz um vetor de características numéricas associado à entrada dada. Então, para objetos que representam as mesmas imagens, reco-

nhecer significa produzir vetores de características numericamente idênticos. Analogamente, reconhecer também significa obter vetores numericamente próximos se os objetos forem gerados a partir de imagens semanticamente semelhantes, e numericamente distantes, se forem gerados a partir de imagens semanticamente diferentes.

3 Características usadas

Nesta seção serão apresentadas as características usadas no algoritmo proposto para o reconhecimento. Foram escolhidas características que capturam o comportamento morfológico global do objeto. No entanto, como será discutido mais tarde, essas características não são únicas, e pode-se definir e extrair diversas outras. Apesar disso, para a finalidade do trabalho, as características usadas são suficientes.

- **Número de Regiões (NR):** A maioria dos mais populares algoritmos de segmentação de imagens costuma gerar como saída, não exatamente uma única região que delimita o objeto em questão separado de seu *background*, mas um conjunto de regiões que foram separadas umas das outras, Fig 3, mesmo pertencendo semanticamente a um mesmo objeto. Pode-se então tomar o número de regiões como sendo uma característica do objeto.
- **Maior/Menor Caminho ((Ma/Me)C):** Segundo [3], o maior caminho entre dois nodos p e q em um grafo é o maior número de vértices percorridos, no máximo uma única vez, entre esses nodos. Analogamente, o menor caminho entre esses nodos é o número mínimo de vértices.

Para o trabalho proposto aqui, o maior/menor caminho é uma característica tomada das regiões do objeto. Os nodos p e q são tomados como sendo os dois mais extremos, com a maior e menor ordenada, respectivamente.

- **Número de Forks (NF):** é o número de nodos em um grafo direcionado que possuem grau de saída pelo menos 2.
- **Número de Splashes (NS):** é o número de nodos em um grafo direcionado que possuem grau de entrada pelo menos 2.

- **Número de Ciclos (NC):** é o número de ciclos em um grafo de acordo com a definição de [3].
- **Eixo Médio Global (EMG):** Cada região i do objeto possui um maior eixo, Em_i , e um menor, En_i , transversal a Em_i . Eixo médio global (EMG) é a razão entre a soma de todos os Em_i pela soma de todos os En_i .
- **Tempo de Preenchimento (TP):** é o número de arestas de um grafo. Chama-se assim para significar o tempo para preencher com água todo o grafo. Assumindo que o algoritmo leva o mesmo tempo para percorrer o caminho entre dois nodos vizinhos.

4 Pré-Processamento

Antes da introdução do algoritmo, será mostrado como a imagem é preparada para dar início à execução. Nesta etapa, também é calculado o limiar d . Este valor é estipulado empiricamente e, neste trabalho, foi tomado como sendo a média das distâncias entre os centróides das regiões da imagem. Antes de serem calculados esses valores, é necessário primeiro segmentar a imagem afim de extrair essas regiões. Embora o foco principal deste artigo seja a apresentação do algoritmo, sugeri-se, como etapa de pré-processamento, a aplicação de filtros lineares como o Sobel seguido de uma cadeia morfológica simples. A simplicidade do pré-processamento tem como principal propósito mostrar mais tarde que, mesmo com a utilização de pré-processamento simples para a extração das regiões, o algoritmo fornece resultados satisfatórios. Assim, a seqüência usada nas imagens com um único objeto e com *background* homogêneo foi a seguinte: aplicação do Sobel seguido de uma etapa de fechamento de “pontes”; operação morfológica “fechamento” com elemento estruturante em cruz; e, finalmente, operação morfológica “dilatação” com o mesmo elemento estruturante. Essas operações visam eliminar pequenas regiões que porventura possam ter sido originadas por ruído, e também segmentar a imagem em regiões importantes, conectadas ou não, de modo que preservem a estrutura morfológica global, requisito fundamental para o algoritmo. A Fig. 2 apresenta uma imagem original do objeto *sciss* e a Fig. 3 mostra o resultado após as operações acima terem sido aplicadas. A partir da imagem resultante, os centróides de cada região são calculados e suas distâncias estipula-

das. A matriz de distâncias, que representa as distâncias de todos os centróides entre si, é a estrutura de dados sob a qual, o algoritmo é executado.



Figura 2: Imagem original de um objeto (*sciss*) usada.



Figura 3: Imagem de regiões da Fig. 2 após o pré-processamento sugerido.

5 O Algoritmo GRAS

Antes de executar o algoritmo, para cada região do objeto é calculado o centróide $c(x, y)$ e rotulado com um valor inteiro. O centróide de maior ordenada (ponto mais alto) é rotulado com o valor 1; a segunda maior ordenada com o valor 2, e assim sucessivamente até ser atingido o cen-

tróide de menor ordenada, que é rotulado com o maior valor inteiro usado, Fig. 1a.

As distâncias entre os centróides são então calculadas e servem como entrada para o algoritmo. Como saída, é gerado um vetor de características numéricas, definidas na Seção 3.

A idéia geral do algoritmo é ligar os pontos entre si criando um grafo direcionado, preenchendo de cima para baixo, a partir do ponto mais alto, sempre conectando os pontos mais próximos até alcançar o mais baixo. No final, o grafo construído possui uma série de características que podem ser extraídas durante a varredura. São características globais da estrutura morfológica do objeto, e que não são capturadas por nenhum algoritmo que se baseia apenas na identificação de bordas internas ou externas.

Seja i, j ($i < j$), dois valores de rótulos de centróides quaisquer do conjunto de regiões que representam o objeto. Seja também d , um valor numérico qualquer para limiar de distância (ver Seção 4). O algoritmo funciona tendo como entrada os valores das distâncias entre os centróides e baseando-se no conjunto de regras enumeradas a seguir:

1. se $i = 1$ e existe j a uma distância menor ou igual a d , aplica-se a regra 2, caso contrário, aplica-se a regra 3.
2. se um centróide $i \neq j$ está a uma distância menor ou igual a d de um centróide j , então cria-se uma aresta direcionada de i para j . Diz-se que i flecha j por limiar;
3. se um centróide i não possui nenhum outro centróide a uma distância menor ou igual a d , e j é o centróide mais próximo dele, então cria-se uma aresta direcionada de i para j . Diz-se que i flecha j por proximidade.
4. se um centróide $i \neq 1$ ainda não foi flechado, ele é flechado pelo centróide, do conjunto de centróides já visitados, mais próximo dele; em seguida aplica-se a regra 2 ou 3 para i .
5. um ponto j não pode flechar um ponto i se i já tiver flechado j .
6. se i for o rótulo de maior valor, não flecha ninguém.

Para ilustrar o algoritmo, considere o conjunto de regiões da Fig. 4a. Cada região foi rotulada com um valor inteiro de 1 a 15, da esquerda para a direita e de cima para baixo. Considere o conjunto de pontos da Fig. 4b. Cada um desses pontos representa o centróide de cada uma das regiões da Fig. 4a.

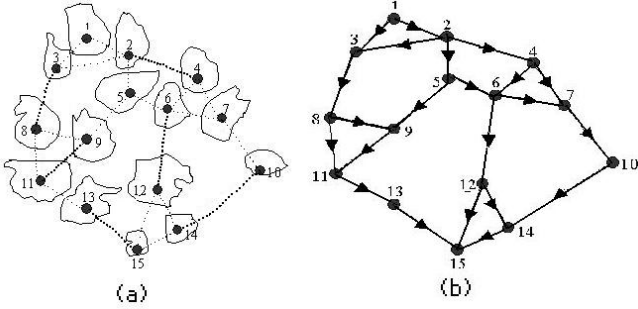


Figura 4: (a) um conjunto de regiões de um objeto; (b) grafo conectado após o algoritmo GRAS

O algoritmo inicia a partir do centróide de coordenada mais alta. Nesse caso, 1. Para simular que um centróide j está a uma distância menor ou igual a d de um ponto i , na Fig. 4a ligamo-los por pontilhados (. . .). Analogamente, se i e j estão separados por uma distância maior do que d , e j é o ponto maior que i mais próximo dele, ligamo-los por tracejados pontos mais fortes (....).

Assim, no exemplo dado e de acordo com o conjunto de regras enumeradas acima, 1 flecha 2 e 3 por limiar (regra 1, e depois 2). Em seguida, o algoritmo continua varrendo o conjunto de pontos a partir de 2. Os pontos que estão a uma distância menor ou igual a d de 2 são 1, 3 e 5. Pela regra 5, 2 não pode flechar 1, mas pela regra 2 flecha 3 e 5 por limiar. Pelas mesmas regras, 3 não pode flechar 1 e 2, mas flecha 8 por proximidade. Continuando com o exemplo dado: 4, como ainda não foi flechado, é flechado por 2 (regra 4) e flecha 6 e 7 por limiar (regra 2); 5 flecha 6 e 9 por limiar (regra 2); 6 flecha 7 por limiar (regra 2); 7 flecha 10 por proximidade (regra 3); 8 flecha 9 e 11 por limiar (regra 2); 9 flecha 11 por proximidade (regra 3); 10 flecha 14 por proximidade (regra 3); 11 flecha 13 por limiar (regra 2); 12, como no caso de 4, é flechado por proximidade por 6 (regra 4) e flecha 14 e 15 por limiar (regra 2); 13 flecha 15 por proximidade (regra 3); 14 flecha 15 por limiar (regra 2); e, finalmente, 15 não flecha ninguém (regra 6).

Durante a varredura do algoritmo, várias características consideradas globais do objeto podem ser ex-

traídas. No entanto, para os propósitos deste artigo, é suficiente o uso apenas das características definidas na Seção 3. No caso do exemplo mostrado aqui, os valores dessas características são: $NR = 15$; $MaC = 8$ (1,2,4,6,7,10,14,15); $MeC = 5$ (1,2,5,6,12,15); $NF = 7$ (1,2,4,5,6,8,12); $NS = 8$ (3,6,7,9,11,14,15,15); $NC = 8$; $TP = 22$; $EMG = (E_{m1} + E_{m2} + \dots + E_{m15}) / (E_{n1} + E_{n2} + \dots + E_{n15})$.

6 Experimentos

Um sistema de visão que utilize o algoritmo proposto, ao receber como entrada um objeto i , é capaz de responder a dois tipos de questões específicas: (a) que objeto representa i ? e (b) quais os objetos que mais se assemelham a i ? Essas duas questões são fundamentais e podem ser aplicadas a sistemas com objetivos diversos. A tipo (a) pode ser necessária quando deseja-se saber precisamente a natureza de um objeto, como por exemplo, que objeto aparece em determinada cena. A tipo (b) está mais ligada à área de recuperação de informação visual, como na *Web*. Pode-se, por exemplo, querer saber, não exatamente a natureza do objeto de entrada, como no caso específico da questão (a), mas quais (ou quantas) imagens representam objetos que mais se assemelham à imagem do objeto dada como entrada.

Para avaliar o algoritmo sob esses dois pontos de vista, foram feitos dois tipos de experimentos, um para cada tipo de questão. O primeiro deles mede o grau de certeza que o algoritmo tem quando se varia o número de consultas; e o segundo mede a precisão com que foram recuperadas as imagens a medida que se varia a quantidade de imagens relevantes recuperadas.

Em todos os experimentos foi usado um único conjunto de imagens. Esse conjunto é composto de 8 subconjuntos diferentes, cada um contendo entre 300 e 800 visões dos seguintes objetos: *eraser*, *pen*, *sciss*, *sharp*, *staple*, *tape*, *triangle* e *vosg*, num total de aproximadamente 4000 imagens distintas. Neste artigo, o conjunto dessas 4000 imagens é chamado de S e cada um dos 8 subconjuntos são denominados da seguinte forma: $E = eraser$, $P = pen$, $Sc = sciss$, $Sh = sharp$, $St = staple$, $Tp = tape$, $Tr = triangle$ e $V = vosg$.

Para acelerar o processo, as consultas são feitas não diretamente sobre S , mas sobre um conjunto B , construído a partir de S calculando-se o vetor de características para

um determinado subconjunto dele. Então, para se construir B , fixou-se o número máximo de visões para cada objeto. No caso dos experimentos apresentados neste artigo, foram usadas 10 visões de 8 objetos diferentes, num total de 80 imagens. Como todas as consultas em B foram feitas com imagens aleatórias de S , isso significa, obviamente, comparar 100% das imagens com somente 2% delas. O objetivo de comparar apenas 80 imagens de B com 4000 imagens aleatórias de S é para mostrar que, mesmo sob condições restritas, o algoritmo obtém resultados satisfatórios. Assim, dada uma imagem i de um objeto, tomada aleatoriamente de algum subconjunto de S , uma consulta em B do tipo (a) é feita classificando-se i como sendo a imagem de um objeto pertencente a um dos 8 subconjuntos de S , citados acima; e uma consulta, em B , do tipo (b) lista um subconjunto de imagens de B que mais se assemelham à imagem i . Um exemplo possível de busca do tipo (b) pode ser observado na Fig. 5. Nela, a imagem mais a esquerda superior é a *query*, e as demais, por ordem de similaridade, são listadas da esquerda para a direita e de cima para baixo. Nota-se que, o maior número de visões possíveis do mesmo tipo de objeto que serve de *query* possui similaridade mais alta. Sempre que se compara duas imagens i e j , a medida de similaridade tomada é a distância euclidiana entre os vetores de características de i e de j .

Para avaliar o algoritmo à luz da questão (a), construiu-se o gráfico *sucesso X número de buscas*, mostrado na Fig. 6. Esse gráfico específico foi montado repetindo-se a consulta para grupos de 1,2,3,...,94 imagens aleatórias de E (o subconjunto de S e B que representa *eraser*), e para cada grupo, tomou-se a porcentagem do total de consultas onde a imagem mais similar era do tipo *eraser*. Como exemplo de interpretação do gráfico, tomemos o ponto de ordenada 28; isso significa que, ao repetir-se a consulta com 28 imagens diferentes de E , em 24 (cerca de 86%), o objeto com maior similaridade foi classificado corretamente como sendo um objeto do tipo E . Analogamente, o pior desempenho, para o objeto do tipo E , ocorreu quando a consulta foi repetida para 94 imagens de E .

Para avaliar o algoritmo à luz da questão (b), foram construídos gráficos do tipo *precisão X revocação*, mostrado na Fig. 7. Esse tipo de gráfico representa, no eixo *revocação*, a variação da razão entre o número de imagens de objetos que o algoritmo deveria recuperar e o número de imagens de objetos realmente recuperados; e no eixo

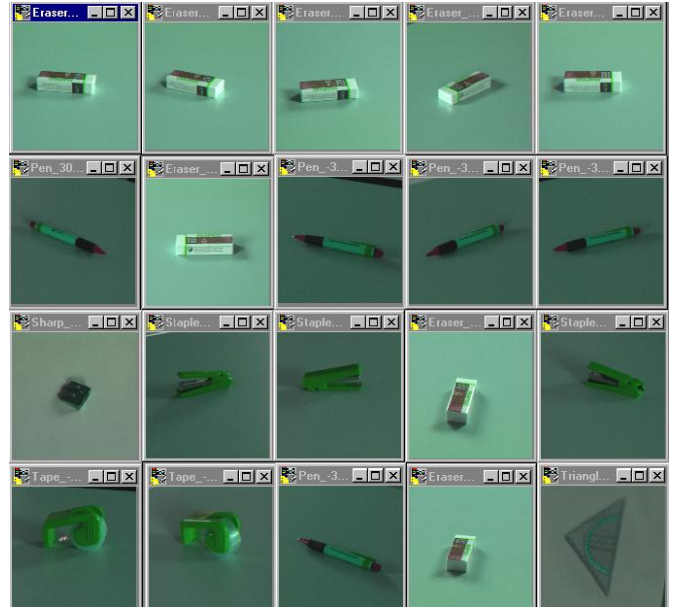


Figura 5: Exemplo de consulta para o objeto do tipo *Eraser*. A imagem mais a esquerda, parter superior, representa a *query*. Nesse caso, outros quatro tipos de objetos *pen*, *staple*, *tape* e *triangle* também foram, embora com similaridade baixa, recuperados.

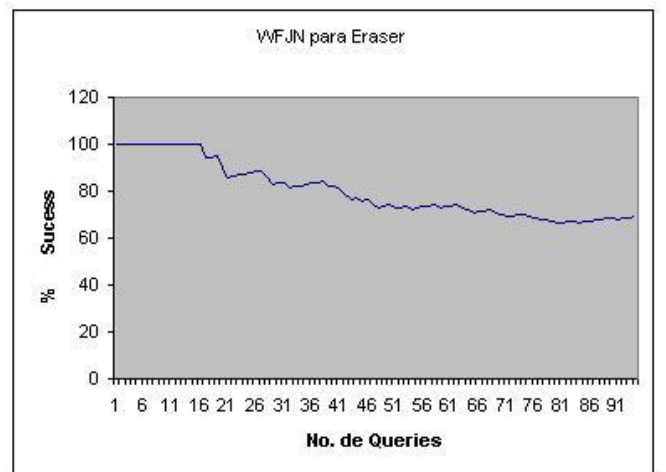


Figura 6: Desempenho do GRAS quando o objeto a ser buscado é do tipo *Eraser*

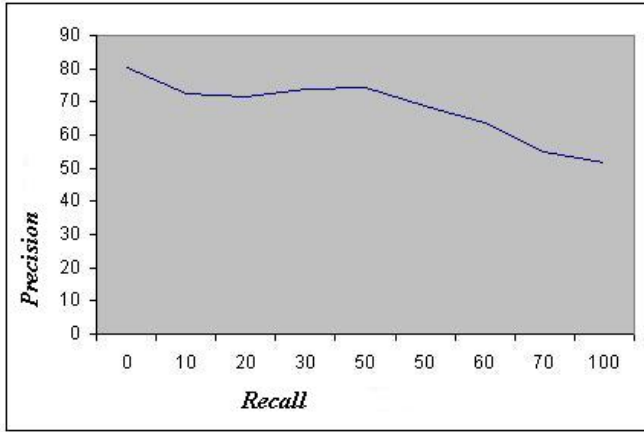


Figura 7: Desempenho do GRAS quando o objeto a ser buscado é do tipo *Eraser*

precisão, a variação do grau de similaridade encontrada; quanto maior o número de imagens relevantes (neste caso, de *eraser*) com alta similaridade encontradas, maior será a precisão. Como exemplo de interpretação desse gráfico, tomemos o ponto de ordenada 50%; isso significa que, quando o algoritmo consegue recuperar 50% das imagens de objetos do tipo *eraser* (imagens relevantes), elas aparecem com grau de similaridade (*precisão*) de 75% em relação à *query* tomada na consulta. Para montar o gráfico da Fig. 7, o experimento foi repetido 94 vezes e a média aritmética simples foi tomada entre os resultados.

O mesmo experimento apresentado na Fig. 6 foi repetido para cada um dos 8 conjuntos de imagens de objetos citados acima. Para efeito de comparação, os resultados foram todos plotados em um só gráfico, e são mostrados na Fig. 8. O mesmo foi feito para o experimento apresentado na Fig. 7, e os resultados são mostrados na Fig. 9.

7 Conclusões

Foi proposto um novo algoritmo o qual pode ser usado na recuperação de imagens de objetos em uma grande base de dados, como Sistemas de Recuperação de Imagens com Base no Conteúdo (RIBIC), bem como, para reconhecimento de objetos específicos como em Sistemas de Inteligência Artificial. Este algoritmo, o qual foi chamado de “GRAS” (*Graph Region Arrow Shot*), obteve bons resultados mesmo usando um pré-processamento simples das imagens de entrada. Devido a este fato, o GRAS possui um baixo custo computacional. Esta é uma carac-

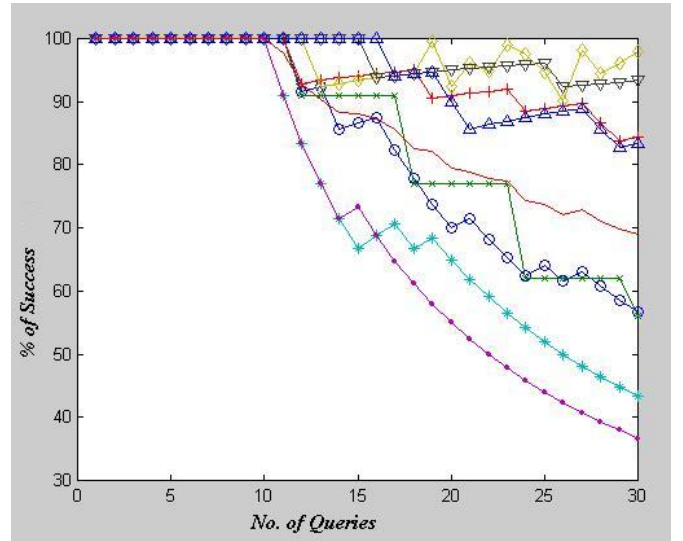


Figura 8: Desempenho do GRAS para cada objeto: (Δ) = eraser; (\circ) = pen; (\times) = sciss; (+) = sharp; (*) = staple; (\bullet) = tape; (\diamond) = triangle; (∇) = vosg. A linha sólida representa a média entre as curvas, portanto, dá uma idéia do comportamento global do algoritmo.

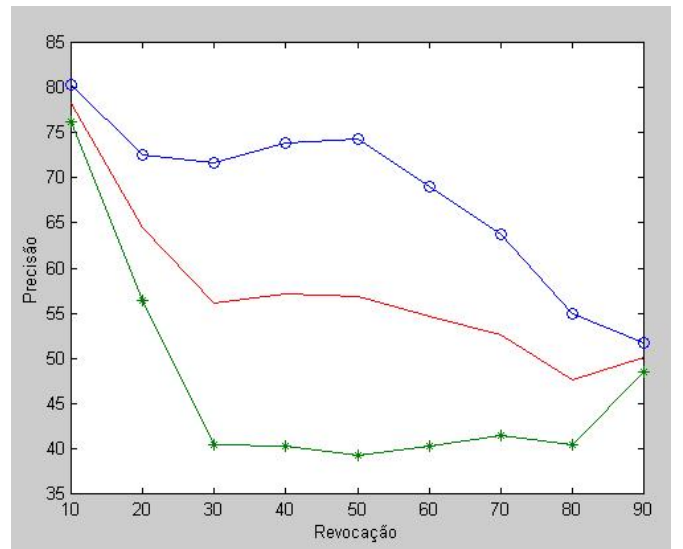


Figura 9: Desempenho do GRAS para cada objeto: (\circ) = eraser; (\star) = pen. A linha sólida representa a média entre as curvas, portanto, dá uma idéia do comportamento global do algoritmo.

terística importante do algoritmo, o qual é passado sobre as regiões da imagem do objeto alvo extraíndo as características globais relevantes para reconhecimento. Esta estratégia extrai informações de forma que, geralmente, não são extraídas por algoritmos que usam processamento local apenas. Nos gráficos “precisão x revocação” é possível avaliar a performance do GRAS para a recuperação de todas as imagens relevantes na base de dados que são similares à imagem de consulta; e nos gráficos “números de consultas x sucesso” é possível avaliar o GRAS de acordo com a capacidade de recuperar a imagem mais similar à imagem de consulta. Para ambos os tipos de experimentos, o GRAS foi executado comparando um pequeno número de imagens, como consultas, com a base de dados total, como imagens-alvos. Embora este seja um trabalho em andamento, os resultados foram encorajadores.

Referências

- [1] Vlatko Becanovic. Image object classification using saccadic search, spatio-temporal patterns encoding and self-organization. *Pattern Recognition Letters*, 21:453–263, 2000.
- [2] M. M. Chun. Contextual cueing of visual attention. *Trends in Cognitive Sciences*, 4(5):170–177, 2000.
- [3] T. H. Cormem, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. McGraw-Hill, 1997.
- [4] J. P. Eakins. Towards intelligent image retrieval. *Pattern Recognition*, 35:3–14, 2002.
- [5] J. P. Eakins and M. E. Graham. Content-based image retrieval. Available online at <http://www.unn.ac.uk/iidr/CBIR/report.html>, October 1999. JISC Technology Applications Programme Report 39.
- [6] J. M. Fuertes, M. Lucena, N. P. de la Banca, and J. Chamorro-Martnez. A scheme of colour image retrieval from databases. *Pattern Recognition Letters*, 22:323–337, 2001.
- [7] V. N. Gudivada and V. V. Raghavan. Content-based image retrieval. *IEEE Computer*, 28(9):18–22, 1995.
- [8] V. N. Gudivada and V. V. Raghavan. Design and evaluation of algorithms for image retrieval by spatial similarity. *ACM Transaction on Information Systems*, 13(2):115–144, 1995.
- [9] A. Guri-Dugu and A. Oliva. Classification of scene photographs from local orientation features. *Pattern Recognition Letters*, 21:1135–1140, 2000.
- [10] J. M. Henderson and A. Hollingworth. High-level scene perception. *Annual Review of Psychology*, 50:243–271, 1999.
- [11] A. K. Jain and A. Vailaya. Image retrieval using color and shape. *Pattern Recognition*, 29(8):1233–1244, 1996.
- [12] D. Lee and H. Kim. A fast content-based indexing and retrieval technique by the shape information in large image database. *The Journal of Systems and Software*, 56:165–182, 2001.
- [13] K. C. Liang and C. C. J. Kuo. Implementation and performance evaluation of a progressive image retrieval system. In I. K. Sethi and R. C. Jain, editors, *Storage and Retrieval for Image and Video Databases*, volume VI, pages 37–48. SPIE 3312, 1998.
- [14] F. Liu and R. W. Picard. Periodicity directionality and randomness: World features for images modelling and retrieval. *IEEE Transaction on Pattern Analyses and Machine Intelligence*, 18(7):722–733, 1996.
- [15] N. H. Mackworth and A. J. Morandi. The gaze selects informative details within pictures. *Perceptual Psychophys*, 2:547–552, 1967.
- [16] B. S. Manjunath and W. Y. Ma. Texture features for browsing and retrieval of large image data. *IEEE Transactions on Pattern Analyses and Machine Intelligence*, 18:837–842, 1996.
- [17] R. Mehrotra and J. E. Gary. Similar shape retrieval in shape data management. *IEEE Computer*, 28(9):57–62, september 1995.
- [18] S. Ravela and R. Manmatha. Retrieving images by appearance. In *Proceedings of IEEE International Conference on Computer Vision (ICCV98)*, pages 608–613, Bombay, India, 1998.

- [19] J. R. Smith and S. F. Chang. *Querying by color regions using the VisualSEEk content-based visual query system*, pages 23–41. AAAI Press, Menlo Park, CA, 1997. Intelligent Multimedia Information Retrieval.
- [20] M. J. Swain and D. H. Ballard. Color indexing. *International Journal on Computer Vision*, 7(1):11–32, 1991.
- [21] S. Ullman. *High-level vision*. Massachusetts Institute of Technology, 1996.
- [22] X. S. Zhou. Edge-based structural features for content-based image retrieval. *Pattern Recognition Letters*, 22:457–468, 2001.