

Processamento de Imagens usando Grafos

Prof. Alexandre Xavier Falcão

Aula 01

1 Introdução

Teoria dos grafos é uma área bem estabelecida, rica em algoritmos eficientes e com várias aplicações nas ciências e engenharias. Neste sentido, soluções eficientes e eficazes para problemas de processamento de imagem podem ser obtidas pela simples redução do problema em um problema de teoria dos grafos com algoritmo conhecido. Esta estratégia também possibilita o desenvolvimento de variantes e novos algoritmos, os quais são necessários em várias aplicações.

A metodologia proposta envolve a interpretação de uma imagem como um grafo. Existem várias formas de modelar um grafo a partir de uma imagem, definindo os nós do grafo e uma relação de adjacência entre eles. Dependendo do problema, pixels, vértices de pixels, arestas de pixels ou regiões da imagem podem ser os nós. A relação de adjacência leva em conta a posição relativa entre os nós e, opcionalmente, propriedades da imagem. Podemos ainda estender a idéia para grafos onde os nós são objetos de uma imagem ou objetos (ou imagens) de um banco de dados.

Como resultado, o problema de processamento de imagem é reduzido à aplicação de um algoritmo em grafo seguida de uma operação local sobre atributos do grafo resultante. O curso apresenta esta metodologia no projeto de operadores de filtragem de imagem, segmentação de imagem, representação e descrição de objetos de imagem, e classificação de padrões em imagem. Aspectos de implementação e os conceitos básicos relacionados com imagem digital e algoritmos em grafo serão apresentados e explicados durante as aulas, à medida que forem sendo necessários.

2 Imagem Digital

Dispositivos de aquisição geram uma **imagem digital multi-dimensional e multi-paramétrica** como um par $\hat{I} = (D_I, \vec{I})$, onde $D_I \subset Z^n$ e $\vec{I}(p) = (I_1(p), I_2(p), \dots, I_k(p)) \in Z^k$ é um mapeamento vetorial que associa k valores inteiros para cada *spel* (*space element* — pixel em nD). Em outras palavras, esses dispositivos obtêm amostras de um conjunto de k propriedades físicas com determinados espaçamentos no \mathfrak{R}^n , cujos valores são quantizados em certos intervalos de inteiros.

Por exemplo, uma foto colorida gera uma imagem com $n = 2$ e $k = 3$, onde $(I_1(p), I_2(p), I_3(p))$ são os valores de luz refletida em $p \in D_I$ nos comprimentos de onda do vermelho, verde, e azul, respectivamente. Imagens de satélite com o sensor *Thematic Mapper*, por exemplo, têm $n = 2$ e $k = 7$, representando imagens da luz refletida na superfície da terra nos comprimentos de onda do azul, verde, vermelho, infra-vermelho próximo, infra-vermelho médio, infra-vermelho, e a quantidade de calor refletida na banda termal. Um vídeo digital, no entanto, mistura espaço e tempo para formar 3 dimensões ($n = 3$) e cada quadro possui 3 bandas de cor ($k = 3$). No caso de imagem cinza, $k = 1$, então a notação será simplificada para (D_I, I) . Se $I(p) \in \{0, 1\}$, $\forall p \in D_I$, \hat{I} é dita **binária**.

Muito embora os valores originais dos spels sejam inteiros, as transformações matemáticas podem gerar valores reais e até complexos. Uma transformada de Fourier $\mathcal{F}(\hat{I})$ de uma imagem cinza $\hat{I} = (D_I, I)$, por exemplo, gera uma imagem complexa $\hat{F} = (D_I, \vec{F})$, onde $\vec{F} = (F_1, F_2)$ contém a parte real $F_1(p)$ e a imaginária $F_2(p)$ para cada pixel $p \in D_I$.

Observe então que um spel p tem uma coordenada espacial D_I e uma coordenada paramétrica $(I_1(p), I_2(p), \dots, I_k(p)) \subset Z^k$. Estas coordenadas definem um espaço de parâmetros (características ou atributos) da imagem \hat{I} (Figura 1).

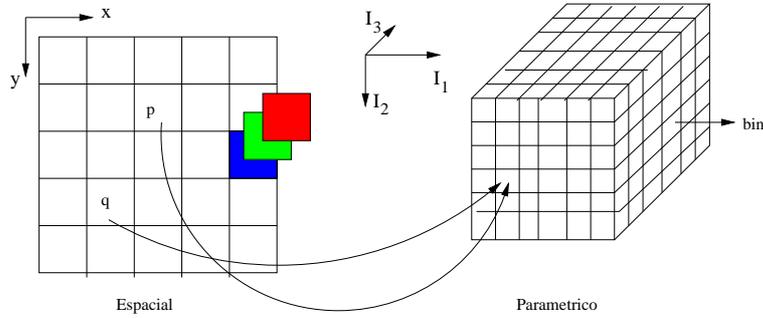


Figura 1: Mapeamento entre as coordenadas espaciais e paramétricas para uma imagem colorida. Pixels de mesma cor caem no mesmo *bin* (similar ao spel) no espaço paramétrico.

3 Grafos a partir de imagens

Por simplicidade, vamos adotar na maioria das definições o caso onde os nós do grafo são pixels $p = (x_p, y_p)$ e os arcos são formados por pixels adjacentes. Portanto, uma **relação de adjacência** $A \subset D_I \times D_I$ é uma relação binária entre pixels, a qual depende de suas posições relativas, e opcionalmente de outras propriedades locais da imagem. Se $(p, q) \in A$ (ou $q \in A(p)$), dizemos que q é **adjacente** a p . Exemplos para o caso de uma imagem 2D:

1. Circular: $(p, q) \in A$ se $d(p, q) \leq \rho$, onde d é distância Euclideana e ρ é um escalar.
2. Retangular: $(p, q) \in A$ se $|x_q - x_p| \leq \frac{a}{2}$ e $|y_q - y_p| \leq \frac{b}{2}$, onde a e b são os comprimentos dos lados do retângulo com centro em (x_p, y_p) .
3. Baseada em conjunto: $(p, q) \in A$ se $q - p \in \{(-1, -1), (1, -1)\}$.

4. Baseada em posição relativa e propriedades locais: $(p, q) \in A$ se $d(p, q) \leq \rho_i$ e $d(\vec{I}(p), \vec{I}(q)) \leq \rho_a$, onde $d()$ é o operador de distância Euclidiana e ρ_i e ρ_a são os raios nos espaços imagem e de atributos (paramétrico), respectivamente.
5. Baseada em posição relativa e propriedades locais: $(p, q) \in A$ se $d(p, q) \leq \rho_i$ e q estiver entre os k -vizinhos mais próximos de p no espaço de atributos, onde k é menor que o número de pixels no raio ρ_i .

Uma relação de adjacência leva, portanto, à definição de um **grafo** $G = (D_I, A)$ para a imagem $\hat{I} = (D_I, \vec{I})$. Conforme mencionado na introdução, os grafos podem ser definidos de várias outras formas, dependendo da aplicação. Portanto, estaremos adotando estas variações à medida que forem necessárias.

No caso de relações de adjacência que só dependem da posição relativa dos pixels q e p (casos de 1 a 3), dizemos que são invariantes à translação. Essas adjacências podem ser implementadas como vetores de deslocamentos $q - p \in \{(dx[i], dy[i])\}$, $i = 0, 1, \dots, |A(p)| - 1$.

4 Filtragem de imagem baseada em operações locais

Vários filtros lineares e não-lineares são baseados em transformações locais que envolvem o brilho do pixel e de seus adjacentes. A ordem em que ocorrem estas transformações não importa, mas normalmente são implementadas varrendo a imagem da esquerda para direita e de cima para baixo.

Um exemplo é a filtragem linear sobre uma imagem cinza $\hat{I} = (D_I, I)$ para gerar uma imagem $\hat{J} = (D_I, J)$, onde

$$J(p) = \sum_{\forall q \in A(p)} w(q - p)I(q) \quad (1)$$

e $w(q - p) \in \Re$ é o coeficiente do filtro na posição $q - p$. Usando os vetores de deslocamentos acima e um peso $w[i]$ para cada posição, a implementação fica:

$$J[x_p, y_p] = \sum_{i=0}^N w[i]I[x_p + dx[i], y_p + dy[i]], \quad (2)$$

onde $(x_p + dx[i], y_p + dy[i]) = (x_q, y_q)$, e $J[x_p, y_p]$ e $I[x_p + dx[i], y_p + dy[i]]$ são as matrizes que armazenam os valores dos pixels nas respectivas imagens.

De forma similar, podemos ter uma filtragem não-linear, como a dilatação morfológica por exemplo:

$$J(p) = \max_{\forall q \in A(p)} I(q). \quad (3)$$

Note que A faz o papel de um elemento estruturante planar.

5 Vetor gradiente

Outro exemplo útil é o cálculo do vetor gradiente de uma imagem $\hat{I} = (D_I, I)$ com base nas diferenças de brilho entre todos os adjacentes e o pixel central. Neste caso, podemos usar a adjacência A circular e definir o vetor gradiente $\vec{G}(p) = (G_x(p), G_y(p))$ por:

$$\vec{G}(p) = \sum_{\forall q \in A(p)} (I(q) - I(p)) \vec{p}q, \quad (4)$$

onde $\vec{p}q = (\frac{x_q - x_p}{d(p,q)}, \frac{y_q - y_p}{d(p,q)})$ é o vetor unitário na direção de p para q , e $d()$ é a distância Euclidiana.

6 Tarefa de casa

Um problema clássico de filtros lineares para suavização de imagens é o fato que eles borram as bordas dos objetos. Filtros de difusão buscam resolver o problema com uma detecção simples de borda para reduzir o tamanho do kernel em pixels próximos das bordas. Uma abordagem similar seria adotarmos a relação de adjacência do caso 5, onde o valor de k seria suficientemente pequeno para evitar adjacentes em regiões distintas e suficientemente grande para gerar uma suavização. Vamos avaliar esta idéia implementando a relação de adjacência do caso 5, uma função para a equação 2 de convolução, uma função para retornar os coeficientes $w(q - p) = \exp^{-d^2(p,q)/2\sigma^2}$, onde $\sigma = 0.5\rho_i$, e um programa que recebe uma imagem cinza e os parâmetros ρ_i e k , aplica esta operação de filtragem, e retorna uma imagem suavizada. Avalie seu programa para várias imagens e valores de ρ_i e k .