



PUCPR- Pontifícia Universidade Católica Do Paraná
PPGIA- Programa de Pós-Graduação Em Informática Aplicada

PROF. DR. JACQUES FACON

LIMIARIZAÇÃO BIMODAL DE OTSU

Resumo: Este artigo descreve a teoria do Algoritmo de Limiarização Bimodal de Otsu assim que a sua implementação.

Palavras Chave: Limiarização, Segmentação Global, algoritmo de Otsu

1. Introdução

O algoritmo a ser implementado tem como objetivo limiarizar uma imagem particionando os pixels de uma imagem de l níveis de cinza em duas classes, C_0 e C_1 , que podem representar o objeto e o fundo.

2. Algoritmo de Limiarização Bimodal de Otsu

O método de limiarização bimodal de Otsu é baseado na análise de discriminante. A operação de limiarização é considerada como sendo o particionamento dos pixels de uma imagem de l níveis de cinza em duas classes, C_0 e C_1 , que podem representar o objeto e o fundo, ou vice-versa, sendo que esta partição se dará no nível de cinza t , desta forma teremos $C_0 = \{0, 1, \dots, t\}$ e $C_1 = \{t + 1, t + 2, \dots, l\}$. Seja σ_w^2 a variância dentro da classe, σ_B^2 a variância entre as classes e σ_T^2 a variância total. Um limiar ótimo pode ser obtido pela minimização de uma das funções critérios seguintes:

$$\lambda = \frac{\sigma_B^2}{\sigma_w^2}$$

$$\eta = \frac{\sigma_B^2}{\sigma_T^2}$$

$$\kappa = \frac{\sigma_T^2}{\sigma_w^2}$$

Das três funções critérios apresentadas acima, η é a mais simples, logo, o limiar ótimo t^* é definido por:

$$t^* = \text{Arg max } \eta$$

sendo que

$$\sigma_T^2 = \sum_{i=0}^{l-1} (i - \mu T)^2 P_i$$

$$\mu T = \sum_{i=0}^{l-1} i P_i$$

$$\sigma_B^2 = \omega_0 \omega_1 (\mu_1 \mu_0)^2$$

$$\omega_0 = \sum_{i=0}^t P_i$$

$$\omega_1 = 1 - \omega_0$$

$$\mu_1 = \frac{\mu T - \mu_t}{1 - \mu_0}$$

$$\mu_0 = \frac{\mu_t}{\omega_0}$$

$$\mu_t = \sum_{i=0}^t i P_i$$

$$P_i = \frac{n_i}{n}$$

Onde n_i é o número de pixels com nível de cinza i e n é o número total de pixels de uma dada imagem definida como:

$$n = \sum_{i=0}^{l-1} n_i$$

Além disso, P_i é a probabilidade de ocorrência do nível de cinza i .

O método de Otsu como proposto disponibiliza meios para se analisar outros aspectos além da seleção de um limiar ótimo para uma dada imagem. Para a seleção do limiar t^* de uma imagem, a classe de probabilidades ω_0 e ω_1 indicam as porções das áreas ocupadas pelas classes C_0 e C_1 . As médias de classes μ_0 e μ_1 servem como estimativa dos níveis médios das classes na imagem original em níveis de cinza. Além disso, o valor máximo de η , denotado por η^* , pode ser usado como medida para avaliar a separabilidade das classes C_0 e C_1 na imagem original ou a bimodalidade do histograma. Esta é uma medida bastante significativa pois é invariante para transformações afins da escala de níveis de cinza; é unicamente determinada dentro do intervalo

$$0 \leq \eta \leq 1.$$

O limite inferior (zero) é obtido quando e somente quando uma dada imagem tenha um único e constante nível de cinza, e o limite superior (um) é obtido quando e somente quando imagens de dois valores são dadas.

Em imagens digitais, a uniformidade dos objetos tem papel importante na separação destes objetos do fundo. De fato, a abordagem de Otsu para limiarização de imagens em nível de cinza é eficiente com base na uniformidade medida entre as duas classes C_0 e C_1 a serem segmentadas.

O método proposto se caracteriza por sua natureza não paramétrica e não supervisionada de seleção de limiar e tem as seguintes desejáveis vantagens:

- O processo como um todo é muito simples; são utilizados somente os momentos cumulativos zero e de primeira ordem do histograma de níveis de cinza;
- Uma extensão direta para problemas que exijam vários limiares é viabilizada face ao critério no qual o método está baseado, qual seja, a análise de discriminantes;
- Um limiar ótimo (ou conjunto de limiares) é selecionado de forma automática e estável, não baseado na diferenciação (uma propriedade local como um vale) mas sim na integração (propriedade global) do histograma;
- Viabiliza a análise de outros aspectos importantes, tais como estimativa dos níveis médios das classes, avaliação da separabilidade das classes, etc;
- O método é extremamente genérico; cobre um largo escopo de processos de decisão não supervisionados; o conjunto de suas aplicações não se limita a binarização de imagens em níveis de cinza tal como descrito anteriormente, mas também pode ser utilizado em outros casos de classificação não supervisionada no qual um histograma de alguma característica discriminativa que classifique objetos esteja disponibilizado.

Cabe ressaltar no entanto, que o limiar calculado não é apropriado para imagens nas quais as variâncias dos níveis de cinza de objetos e fundo ou populações de pixels correspondentes sejam extremamente diferentes.

3. Referências

Otsu N., "A Threshold Selection Method from Gray-level Histograms", IEEE Transactions on Systems, Man and Cybernetics, v. SMC 9, no 1, pp.62-66, 1979.

IMPLEMENTAÇÃO:

```
/*-----*/
/*      Artigo: A THRESHOLD SELECTION METHOD FROM GRAYLEVEL HISTOGRAMS      */
/*OTSU, IEEE TRANS. SYSTEMS, MAN & CYBERNETICS, VOL 9 No1 JAN.1979, 62-66*/
void Limiarotsu (bufferout,bufferin,grade,width,height)
unsigned char      *bufferout, *bufferin;
int               grade ;
int               width,height;
{
    int           col,lin,val,klimiar,col1,lin1;
    int           histo[256];
    float  media[256],proba[256],ww[256],mu[256],mutotal[256],var[256],ratio[256],vartotal,maxratio;
    unsigned char data;

    for (lin1 = 0; lin1 < height; lin1 += grade)
    {
        for (col1 = 0; col1 < width ; col1 += grade)
        {

            histogram(bufferin,col1,lin1,col1+grade,lin1+grade,histo,width,height);

            /* calculo das probabilidades a priori */

            for (val = 0; val < 256; val++)
                proba[val] = (float) (histo[val]/(float)(grade*grade) );

            for (val = 0; val < 256; val++)                media[val] = 0.0;

            media[1] = (float) proba[1] ;
            ww[1] = (float) proba[1] ;
            mu[1] = (ww[1] != 0) ? media[1]/ww[1] : 0.0 ;
            for (val = 2; val < 256; val++)
            {
                ww[val] = ww[val-1] + proba[val] ;
                media[val] = media[val-1] + (float) proba[val]*val ;
                mu[val] = (ww[val] != 0) ? media[val]/ww[val] : 0 ;
            }

            for (val = 1; val < 256; val++)
            {
                mutotal[val] = ((1-ww[val]) != 0) ? (media[255] - media[val])/(float)(1-ww[val]) : 0 ;
                /* sigmaB^2= wov1(mu1-m2)(mu1-m2) */
                var[val] = ww[val]*(1- ww[val])* (mu[val] - mutotal[val])*(mu[val] - mutotal[val]) ;
            }

            vartotal = 0 ;
            for (val = 1; val < 256; val++)
                vartotal += (val-mu[255]) * (val-mu[255]) * proba[val] ; /* sigmaT^2= Soma (i-muT)^2 Pi */

            for (val = 1; val < 256; val++)
                ratio[val] = var[val]/vartotal ;
```

```

/* calculo do limiar klimiar */

val = 1;
maxratio = (float) ratio[val];
klimiar = val ;
for (val = 2; val < 256 ; val++)
{
    if( ratio[val] > maxratio )
    {
        maxratio = (float) ratio[val];
        klimiar = val ;
    }
}

printf("\n          Valor do limiar %d", klimiar);

for (lin = lin1; lin < lin1 + grade; lin++)
for (col = col1; col < col1 + grade ; col++)
{
    data = *(bufferin + lin*width + col);
    *(bufferout + lin*width + col) = (data < klimiar) ? 0 : 255;
}
}
}
}

/*-----*/
XtCallbackProc LimiarOtsu (w,client_data,call_data)
Widget w;
caddr_t client_data;
caddr_t call_data;
{
    int grade ;

    printf ("\n Inicio da limiarizacao Otsu --> Maximizacao da varianca entre classes\n");
    printf ("          Tamanho da grade ?");
    scanf("%d",&grade);

    Limiarotsu(DataRes.Buffer,Data.Buffer,grade,Data.Width,Data.Height);

    printf ("\n Termina da limiarizacao Otsu \n");
    XBell (XtDisplay(MainForm),0);
    DrawImage();
}

```