Processamento Digital de Imagens

Prof. Bogdan Tomoyuki Nassu

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Hoje

- Como tratar imagens com variações locais de iluminação?
  - Isso é um desafio para a nossa abordagem usando binarização.

Hoje

- O limiar sempre fica muito baixo...

Hoje

- ... ou muito alto.

Variações locais de iluminação

•Como modificar a binarização com limiarização para que ela trate variações locais de iluminação?

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Variações locais de iluminação

- Como modificar a binarização com limiarização para que ela trate variações locais de iluminação?
- R: Em vez de comparar cada pixel com um limiar fixo, podemos comparar cada pixel com a sua vizinhança.

Variações locais de iluminação

Variações locais de iluminação

•Como modificar a binarização com limiarização para que ela trate variações locais de iluminação?

•R: Em vez de comparar cada pixel com um limiar fixo, podemos comparar cada pixel com a sua vizinhança.

- = Não importa tanto se o pixel é claro, e sim se ele é claro em comparação com o que está em volta dele!
- Como fazer esta comparação?

Limiarização adaptativa

- Uma medida simples: a média das intensidades dos pixels ao redor de um pixel.
  - Normalmente, usamos uma janela quadrada ou retangular.
    - Motivo: eficiência (veremos isso em breve).

```
for (cada pixel f(x,y))
    define uma janela de largura w
        esquerda em x-w/2
        direita em x+w/2
        topo em y-w/2
        baixo em y+w/2
    toma a média μ(x,y) dos pixels na janela
    se f(x,y)-μ(x,y) > limiar
        g(x,y) = 1
    senão
        g(x,y) = 0
```

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Calculando as médias locais

- Nosso desafio é então calcular a média para cada janela.
- Isso pode ser feito através do *filtro da média*.
  - *Blur, box blur, box filter, mean filter.*
  - Entrada: uma imagem.
  - Parâmetros: a largura e a altura da janela.
    - Os dois valores precisam ser ímpares.
  - Saída: outra imagem, com cada pixel $g(x,y)$ tendo o valor médio dos pixels ao redor do pixel $f(x,y)$ na imagem original.
    - Para uma janela quadrada de largura $2w+1$:

$$g(x,y)=\frac{1}{(2w+1)^2}\sum_{i=-w}^{i=+w}\sum_{j=-w}^{j=+w} f(x+i,y+j)$$

- Qual é a aparência da imagem de saída?

Filtro da média

- Qual é a aparência da imagem de saída?
  - R: uma versão borrada da imagem de entrada.
  - Quanto maior o tamanho da janela, mais borrada a imagem.

Sem blur | 3x3 | 7x7 | 15x15 | 25x25

- O filtro da média tem outras utilidades, além de produzir as médias para a limiarização adaptativa.
  - Mais sobre isso em breve...

Aplicando o filtro da média

- O filtro da média pode ser aplicado com uma “janela deslizante”.
  - Vejamos um exemplo com uma janela 3x3.
  - Por enquanto, vamos ignorar pixels cujas janelas ficariam fora da imagem (as margens).

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Filtro da média: exemplo 3x3

$$\frac{0+0.4+0.5+0.2+0.7+0.9+0+0+0}{9}$$

Entrada

0	0.4	0.5	1.0	0.3	0.5
0.2	0.7	0.9	0.2	0.1	0
0	0	0	0	0	0
0	0	0	1.0	0	0
0	0	0	1.0	0	0
0	0	0	1.0	0	0

Saída

	0.3				

Filtro da média: exemplo 3x3

Entrada

0	0.4	0.5	1.0	0.3	0.5
0.2	0.7	0.9	0.2	0.1	0
0	0	0	0	0	0
0	0	0	1.0	0	0
0	0	0	1.0	0	0
0	0	0	1.0	0	0

Saída

	0.3	0.4			

Filtro da média: exemplo 3x3

Entrada

0	0.4	0.5	1.0	0.3	0.5
0.2	0.7	0.9	0.2	0.1	0
0	0	0	0	0	0
0	0	0	1.0	0	0
0	0	0	1.0	0	0
0	0	0	1.0	0	0

Saída

	0.3	0.4	0.3		

Filtro da média: exemplo 3x3

Entrada

0	0.4	0.5	1.0	0.3	0.5
0.2	0.7	0.9	0.2	0.1	0
0	0	0	0	0	0
0	0	0	1.0	0	0
0	0	0	1.0	0	0
0	0	0	1.0	0	0

Saída

	0.3	0.4	0.3	0.2	

Filtro da média: exemplo 3x3

Entrada

0	0.4	0.5	1.0	0.3	0.5
0.2	0.7	0.9	0.2	0.1	0
0	0	0	0	0	0
0	0	0	1.0	0	0
0	0	0	1.0	0	0
0	0	0	1.0	0	0

Saída

	0.3	0.4	0.3	0.2	
	0.2				

Filtro da média: exemplo 3x3

Entrada

0	0.4	0.5	1.0	0.3	0.5
0.2	0.7	0.9	0.2	0.1	0
0	0	0	0	0	0
0	0	0	1.0	0	0
0	0	0	1.0	0	0
0	0	0	1.0	0	0

Saída

	0.3	0.4	0.3	0.2	
	0.2	0.3	0.2	0.1	
	0	0.2	0.2	0.2	
	0	0.3	0.3	0.3	

Implementando o filtro da média

- Como implementar o filtro da média?
  - Primeiro: como seria um algoritmo “ingênuo”?
  - Por enquanto, vamos simplesmente ignorar as margens.
  - Vamos supor uma janela de altura *h* e largura *w*.

Algoritmo ingênuo

```
for (cada linha y)
    for (cada coluna x)
    {
        soma = 0;
        for (cada linha y' no intervalo [y-h/2,y+h/2])
            for (cada coluna x' no intervalo [x-w/2,x+w/2])
                soma += f(x',y')
        g(x,y) = soma / (h*w)
    }
```

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Algoritmo ingênuo

- Por que o algoritmo ingênuo é... “ingênuo”?

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Algoritmo ingênuo

- Por que o algoritmo ingênuo é... “ingênuo”?
  - R: Ele realiza muitas operações redundantes.
- Como poderíamos reaproveitar computações anteriores?

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Algoritmo menos ingênuo

- Por que o algoritmo ingênuo é... “ingênuo”?
  - R: Ele realiza muitas operações redundantes.
- Como poderíamos reaproveitar computações anteriores?
  - R: Podemos guardar a soma da coluna anterior. Quando a janela “desliza”, precisamos apenas remover os valores que “saem” e adicionar os valores que “entram” na janela.

Algoritmo menos ingênuo

Soma = 2.7
2.7 / 9 = 0.3

Entrada

0	0.4	0.5	1.0	0.3	0.5
0.2	0.7	0.9	0.2	0.1	0
0	0	0	0	0	0
0	0	0	1.0	0	0
0	0	0	1.0	0	0
0	0	0	1.0	0	0

Saída

	0.3				

Algoritmo menos ingênuo

Soma = 2.7 – (0+0.2+0) + (1.0+0.2+0) = 3.7
3.7 / 9 = 0.4111

Entrada

0	0.4	0.5	1.0	0.3	0.5
0.2	0.7	0.9	0.2	0.1	0
0	0	0	0	0	0
0	0	0	1.0	0	0
0	0	0	1.0	0	0
0	0	0	1.0	0	0

Saída

	0.3	0.4			

Algoritmo menos ingênuo

Soma = 3.7 – (0.4+0.7+0) + (0.3+0.1+0) = 3
3 / 9 = 0.3333

Entrada

0	0.4	0.5	1.0	0.3	0.5
0.2	0.7	0.9	0.2	0.1	0
0	0	0	0	0	0
0	0	0	1.0	0	0
0	0	0	1.0	0	0
0	0	0	1.0	0	0

Saída

	0.3	0.4	0.3		

Comparando os algoritmos

- O quanto o algoritmo que mantém as somas é melhor?
  - Vejamos quantos pixels cada alternativa acessa, aproximadamente.
  - Considere que temos *N* linhas x *M* colunas e uma janela *h* x *w.*
  - O tempo de execução nós vamos comparar em breve.
- Algoritmo ingênuo:
- Algoritmo que mantém as somas:

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Melhorando mais

- É possível melhorar o algoritmo se observarmos que a soma de uma região retangular é *separável*.
  - = Podemos calcular a média na horizontal, e a “média das médias horizontais” na vertical.
  - É mais fácil de entender vendo um exemplo de execução.

a	b	c	d	e	f
g	h	i	j	k	l
m	n	o	p	q	r
s	t	u	v	w	x

Na horizontal...

Na vertical...

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Comparando os algoritmos

- Vamos contar os pixels acessados considerando que o filtro da média é separável.
  - Conta aproximada.
  - *N* linhas x *M* colunas e uma janela *h* x *w.*
- Algoritmo ingênuo:
- Algoritmo que mantém as somas:
- Filtro separável:
- Filtro separável + mantém as somas:

Sobre alocação de memória

- Os dois primeiros algoritmos precisam de uma imagem de entrada e outra de saída.
  - Não podemos realizar a filtragem *in-place* porque precisamos dos valores originais.
- O filtro separável precisa ainda de uma terceira imagem.

- A alocação de memória consome tempo!
  - Este aspecto é frequentemente negligenciado.
  - Considerar o filtro da média como um filtro separável pode ser vantajoso apenas quando a janela é grande.
  - Outro “truque” é criar um buffer e reaproveitá-lo para várias funções / chamadas que precisam dele.

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Imagens integrais

- Podemos criar um algoritmo ainda mais rápido para o filtro da média usando *imagens integrais*.
  - A imagem integral é obtida somando cada pixel com todos os pixels que estão acima e à esquerda.

$$g(x,y)=\sum_{\substack{x'\leq x\\ y'\leq y}} f(x',y')$$

```
for (cada linha y)
{
    g(0,y) = f(0,y)
    for (cada coluna x fora a primeira)
        g(x,y) = f(x,y) + g(x-1,y)
}

for (cada linha y fora a primeira)
    for (cada coluna x)
        g(x,y) = g(x,y) + g(x,y-1)
```

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Imagens integrais

- Como aproveitar uma imagem integral para implementar o filtro da média?
  - Vejamos um exemplo...

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

a	b	c	d
e	f	g	h
i	j	k	l
m	n	o	p

a	a + b	a + b + c	a + b + c + d
a + e	a + b + e + f	a + b + c + e + f + g	a + b + c + d + e + f + g + h
a + e + i	a + b + e + f + i + j	a + b + c + e + f + g + i + j + k	a + b + c + d + e + f + g + h + i + j + k + l
a + e + i + m	a + b + e + f + i + j + m + n	a + b + c + e + f + g + i + j + k + m + n + o	a + b + c + d + e + f + g + h + i + j + k + l + m + n + o + p

Imagens integrais

- Como aproveitar uma imagem integral para implementar o filtro da média?
  - R: A imagem integral nos permite calcular a soma de *qualquer* região da imagem apenas com a soma e subtração de 4 valores.

$$g(x,y)=\sum_{\substack{x'\leq x\\ y'\leq y}} f(x',y')$$

$$\sum_{\substack{l<x\leq r\\ t<y\leq r}} f(x,y)=g(r,b)-g(r,t)-g(l,b)+g(l,t)$$

- Para implementar o filtro da média, basta obter a soma para cada janela e dividir pelo tamanho da janela.

Comparando os algoritmos

- Vamos contar os pixels acessados considerando que o filtro da média é separável.
  - Conta aproximada.
  - *N* linhas x *M* colunas e uma janela *h* x *w.*
- Algoritmo ingênuo:
- Algoritmo que mantém as somas:
- Filtro separável:
- Filtro separável + mantém as somas:
- Algoritmo com imagens integrais:
  - Este algoritmo também precisa de um buffer auxiliar.
- Vamos comparar os tempos obtidos por implementações dos 4 algoritmos!

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Margens

- O que fazer com os pixels cuja janela ficaria fora da imagem?

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Ignorar completamente (pular)?

Preencher com cor constante?

Usar valores da imagem original?

Margem de cor constante?

Margem de cor constante?

Margem de cor constante?

Margem de cor constante?

Pixels nas margens se repetem?

Pixels nas margens se repetem?

Pixels nas margens se repetem?

Pixels nas margens se repetem?

Imagem se repete infinitamente?

Imagem se repete infinitamente?

Imagem se repete infinitamente?

Imagem se repete infinitamente?

Imagem se repete espelhada?

Imagem se repete espelhada?

Imagem se repete espelhada?

Imagem se repete espelhada?

- A alternativa padrão no OpenCV é uma variante desta, mas os pixels da primeira/última linha/coluna não são replicados.

Margens

- O que fazer com os pixels cuja janela ficaria fora da imagem?
- Várias alternativas:
  - Ignorar completamente (simplesmente pular!).
  - Preencher com uma cor constante.
  - Preencher com os valores da imagem de entrada.
  - Supor uma margem de cor constante.
  - Valores nas extremidades se repetem infinitamente.
  - Imagem se repete infinitamente.
  - Imagem se repete espelhada.
  - Aplicar o filtro para janelas menores nas margens (= faz a média com menos valores).
- Qual a melhor alternativa para o filtro da média?

Margens

- O que fazer com os pixels cuja janela ficaria fora da imagem?
- Várias alternativas:
  - Ignorar completamente (simplesmente pular!).
  - Preencher com uma cor constante.
  - Preencher com os valores da imagem de entrada.
  - Supor uma margem de cor constante.
  - Valores nas extremidades se repetem infinitamente.
  - Imagem se repete infinitamente.
  - Imagem se repete espelhada.
  - Aplicar o filtro para janelas menores nas margens (= faz a média com menos valores).
- Qual a melhor alternativa para o filtro da média?
  - R: indeterminado!!! Depende do resultado desejado e da aplicação!

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Voltando...

- Podemos agora retornar à limiarização adaptativa.
- Atualizando o algoritmo:

```
boxBlur (f, μ, h, w)

for (cada pixel f(x,y))
    se f(x,y)−μ(x,y) > limiar
        g(x,y) = 1
    senão
        g(x,y) = 0
```

- Vejamos o resultado da subtração e da limiarização para alguns exemplos.

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Filtro da média: outros usos

- O filtro da média tem outras utilidades!
  - Exemplo: implementação rápida de efeito *depth of field*.
  - Exemplo: simular efeitos de luz e sombra.
  - Uso mais comum: borrar a imagem no início do processamento.
    - Por que alguém faria isso?

Efeito obtido com um filtro da média modificado, com formato hexagonal.

Luz e sombra

Ruído

- Toda medida feita por um sensor tem ruído.
  - Normalmente, resulta em pequenas variações na intensidade de pixels individuais.
  - = regiões homogêneas no mundo real surgem com variações em imagens capturadas.
  - Duas imagens capturadas em sequência, sem movimentação de câmera, podem ter variações no valor do pixel na mesma posição.

Patch com 25x25 pixels, capturado por uma webcam de baixa qualidade.

Superfícies heterogêneas

- Além do ruído, existem as variações naturais dos materiais.
  - Olhe para qualquer superfície ao seu redor – você não vai encontrar nada totalmente liso e homogêneo!
    - (ou mesmo totalmente limpo!)
- Muitas vezes, não queremos que o nosso programa “enxergue” tantos detalhes.
- O que isso tem a ver com o filtro da média?

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Superfícies heterogêneas

- O que isso tem a ver com o filtro da média?
  - O filtro da média “mistura” dados de pixels próximos, removendo detalhes e tornando as superfícies mais homogêneas.

Box blur no pré-processamento

- Suponha que apenas a região da página nos interessa...

Box blur no pré-processamento

- Limiarização global com T=0.3.

Box blur no pré-processamento

- Limiarização global com T=0.5.

Box blur no pré-processamento

- Aplicando o filtro da média 15x15 na imagem de entrada.

Box blur no pré-processamento

- Limiarização global com T=0.3.