

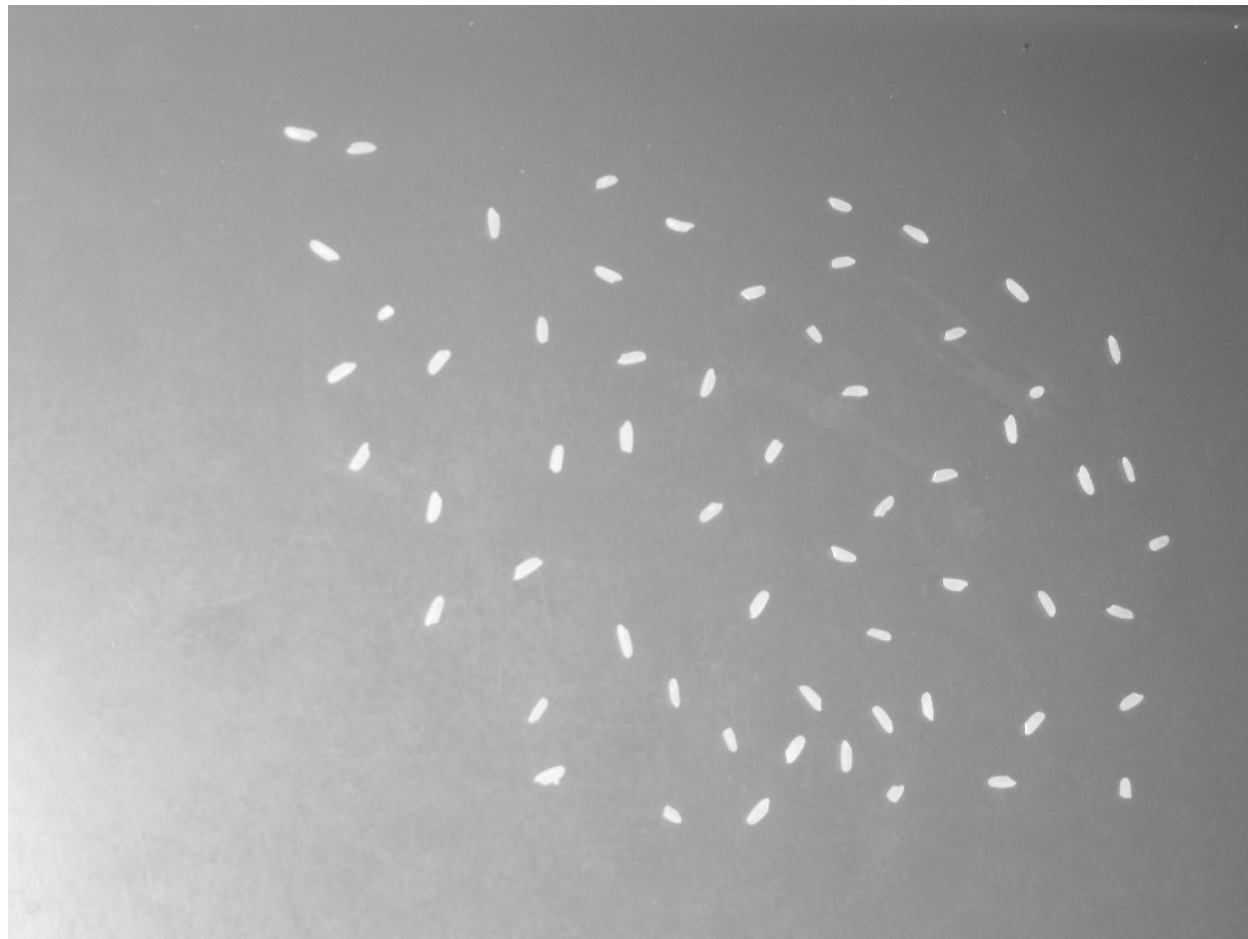
Processamento Digital de Imagens

Prof. Bogdan Tomoyuki Nassu



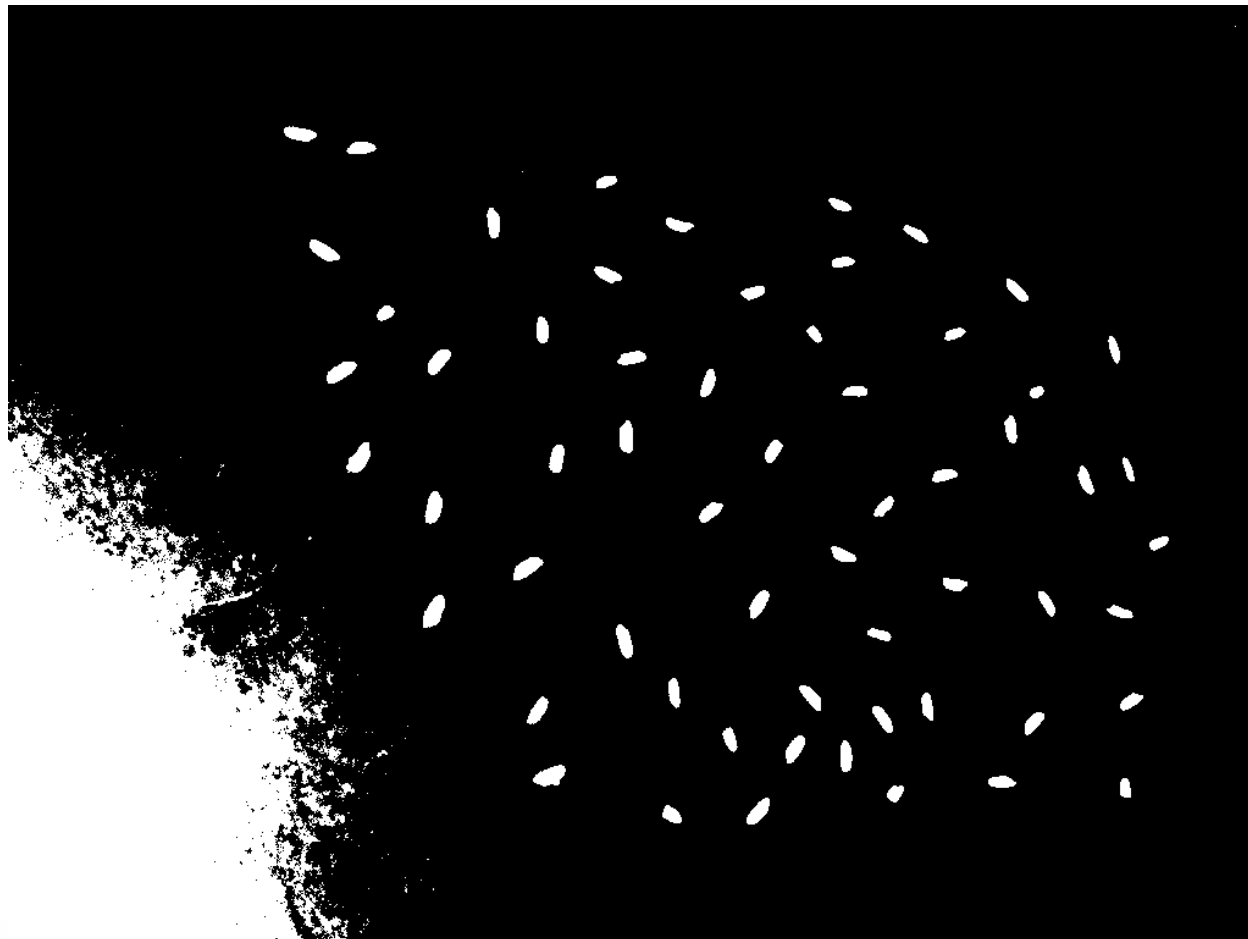
Hoje

- 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...



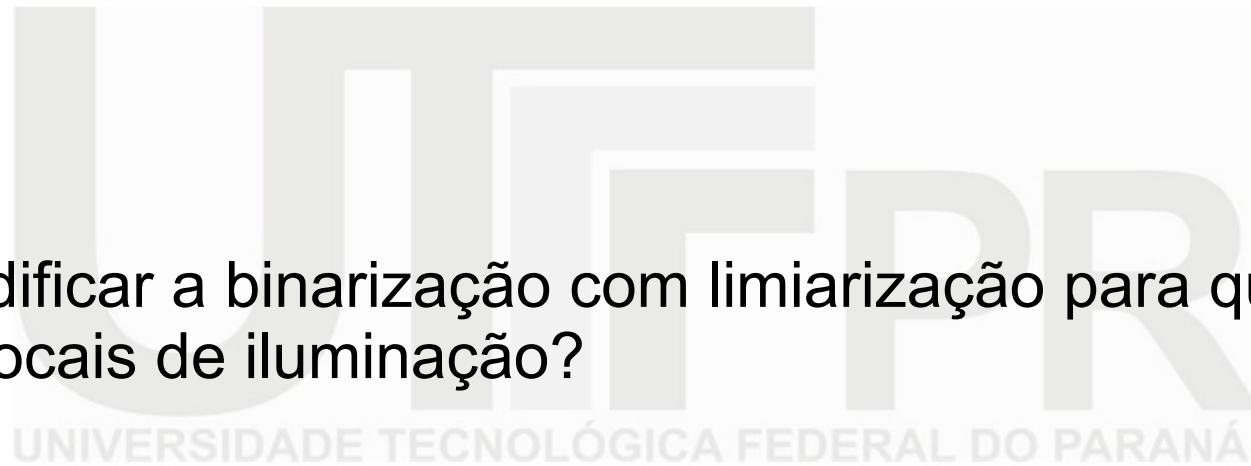
Hoje

- ... ou muito alto.



Variações locais de iluminação

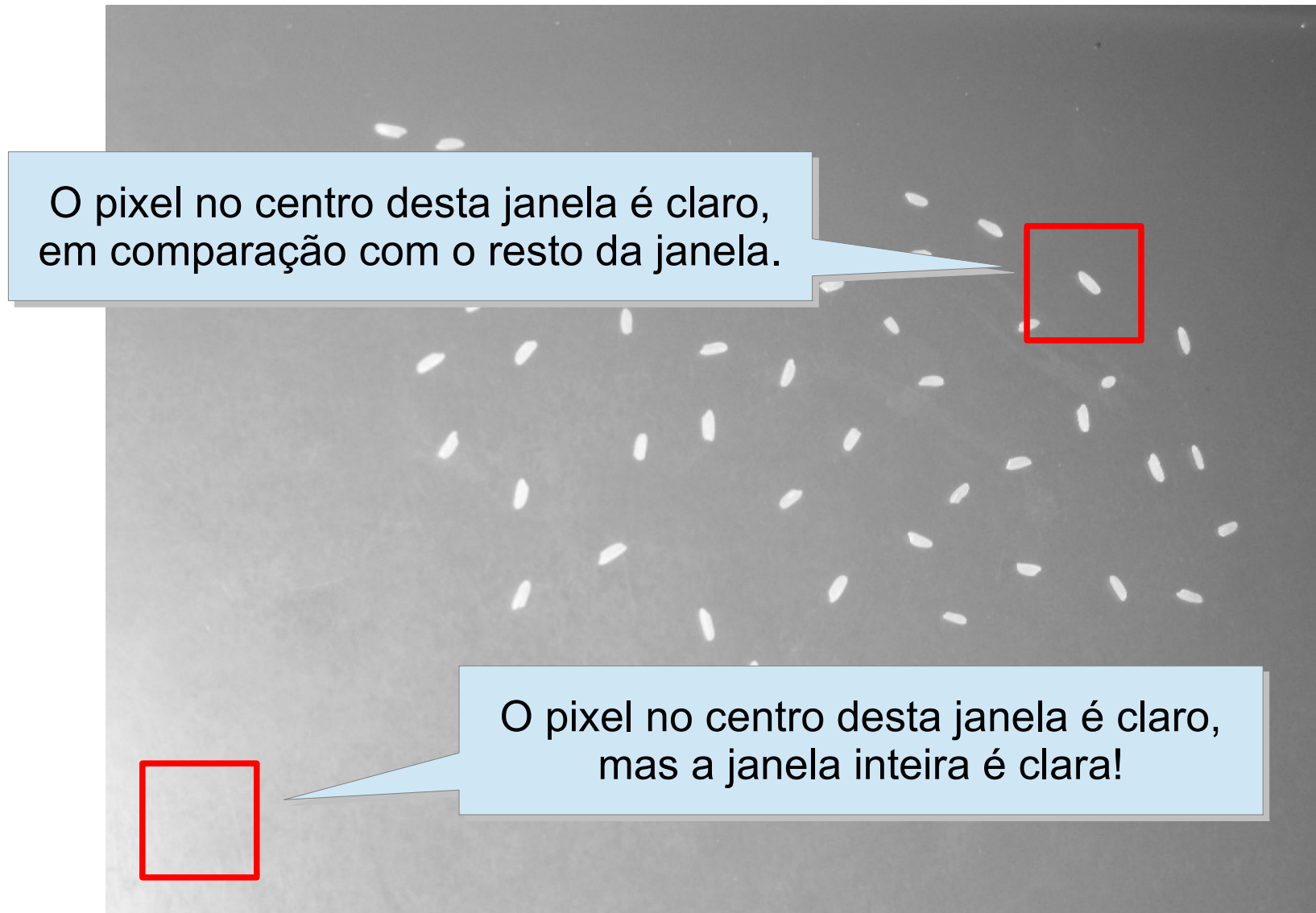
- Como modificar a binarização com limiarização para que ela trate variações locais de iluminação?



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  $\mu(x,y)$  dos pixels na janela  
  se  $f(x,y) - \mu(x,y) > \text{limiar}$   
     $g(x,y) = 1$   
  senão  
     $g(x,y) = 0$ 
```

Calculando as médias locais

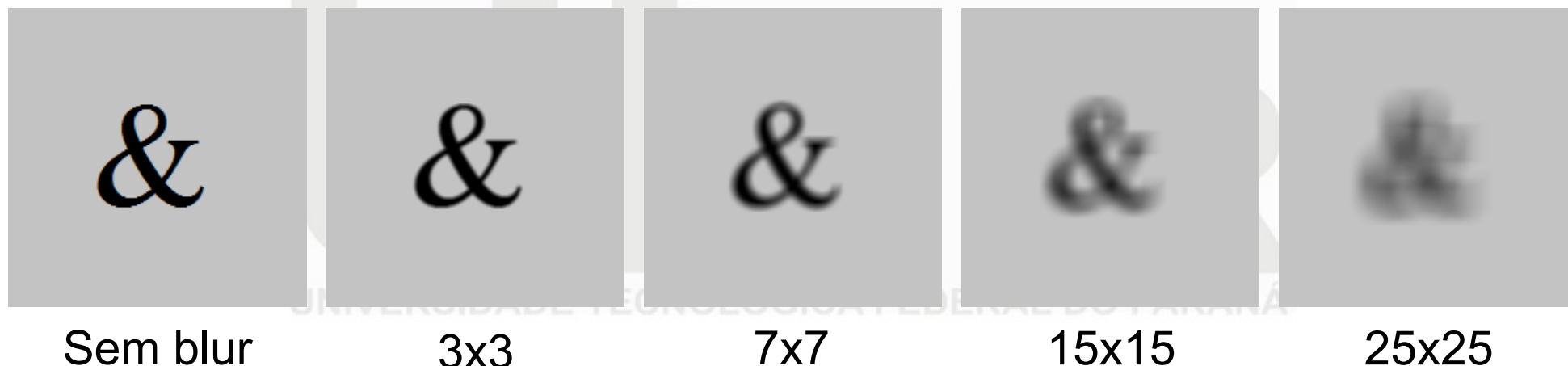
- Nosso desafio é então calcular a média para cada janela.
- Isso pode ser feito através do *filtro da média*.
 - *Blur, box blur, box filter, mean filter.*
 - Entrada: uma imagem.
 - Parâmetros: a largura e a altura da janela.
 - Os dois valores precisam ser ímpares.
 - Saída: outra imagem, com cada 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.



- 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).

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

Filtro da média: exemplo 3x3

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

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

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 +=  $f(x', y')$ 
     $g(x, y) = \text{soma} / (h*w)$ 
  }
```

Algoritmo ingênuo

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



Algoritmo ingênuo

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

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

$$\text{Soma} = 2.7 - (0+0.2+0) + (1.0+0.2+0) = 3.7$$
$$3.7 / 9 = 0.4111$$

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

$$\text{Soma} = 3.7 - (0.4+0.7+0) + (0.3+0.1+0) = 3$$
$$3 / 9 = 0.3333$$

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		

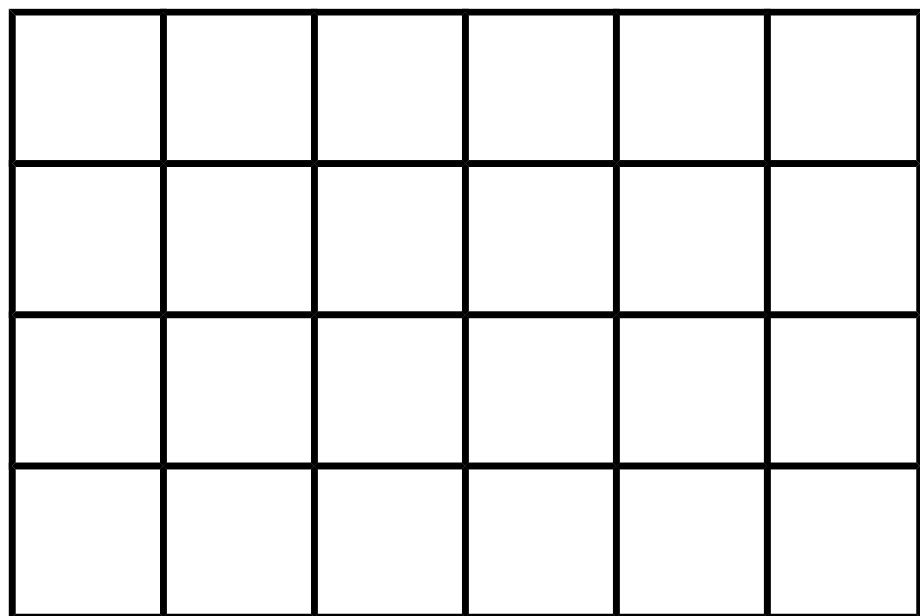
Comparando os algoritmos

- 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:

Melhorando mais

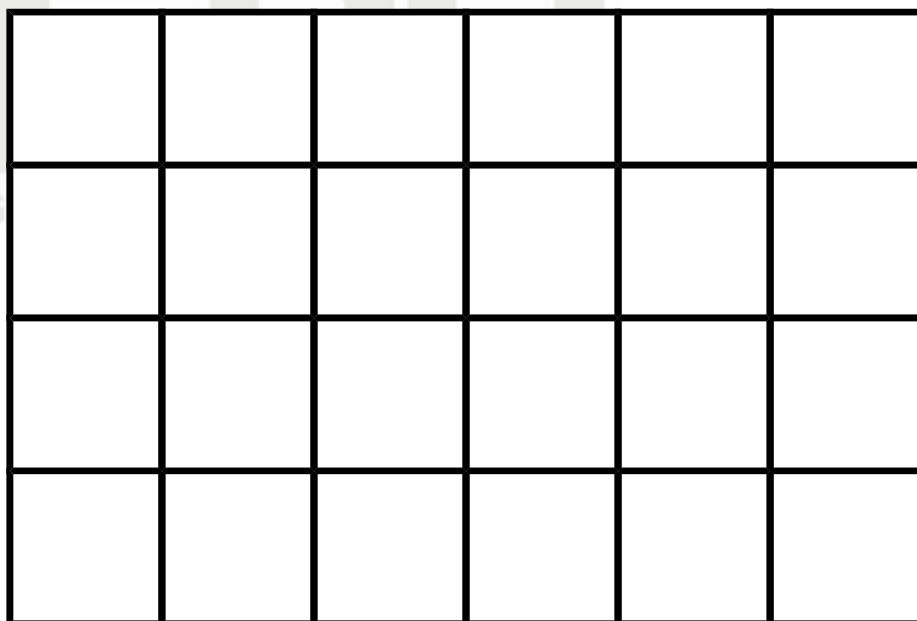
- É possível melhorar o algoritmo se observarmos que a soma de uma região retangular é *separável*.
 - = Podemos calcular a média na horizontal, e a “média das médias horizontais” na vertical.
 - É mais fácil de entender vendo um exemplo de execução.

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...

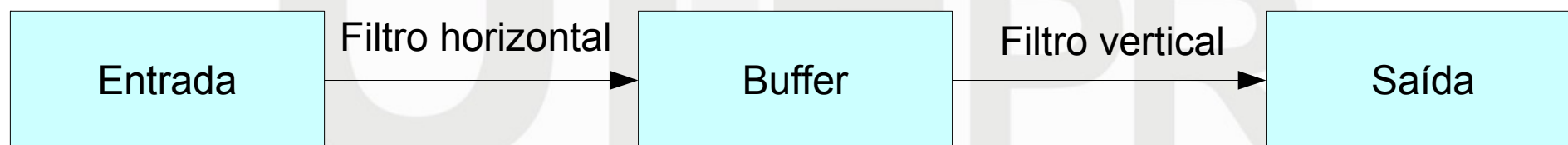


Comparando os algoritmos

- 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.

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)
```

Imagens integrais

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



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.

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

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

- 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!

Margens

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



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.
- 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.
- Qual a melhor alternativa para o filtro da média?
 - R: isso não existe, mas normalmente a última alternativa produz resultados mais “agradáveis”.
 - A implementação é simples para o algoritmo com imagens integrais!

Voltando...

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

```
boxBlur ( $f$ ,  $\mu$ ,  $h$ ,  $w$ )
```

```
for (cada pixel  $f(x,y)$ )  
  se  $f(x,y) - \mu(x,y) > \text{limiar}$   
     $g(x,y) = 1$   
  senão  
     $g(x,y) = 0$ 
```

- Vejam os resultados da subtração e da limiarização para alguns exemplos.

Outras alternativas

- A média de uma região quadrada é apenas uma das medidas que podemos usar na limiarização adaptativa.
 - Ela é normalmente usada por ter baixo custo computacional.
- Outras medidas podem ser usadas, como valores mínimos ou a mediana.
 - Funcionamento similar ao filtro da média: janela deslizante com regiões quadradas.
 - Vejamos alguns exemplos usando a mediana...

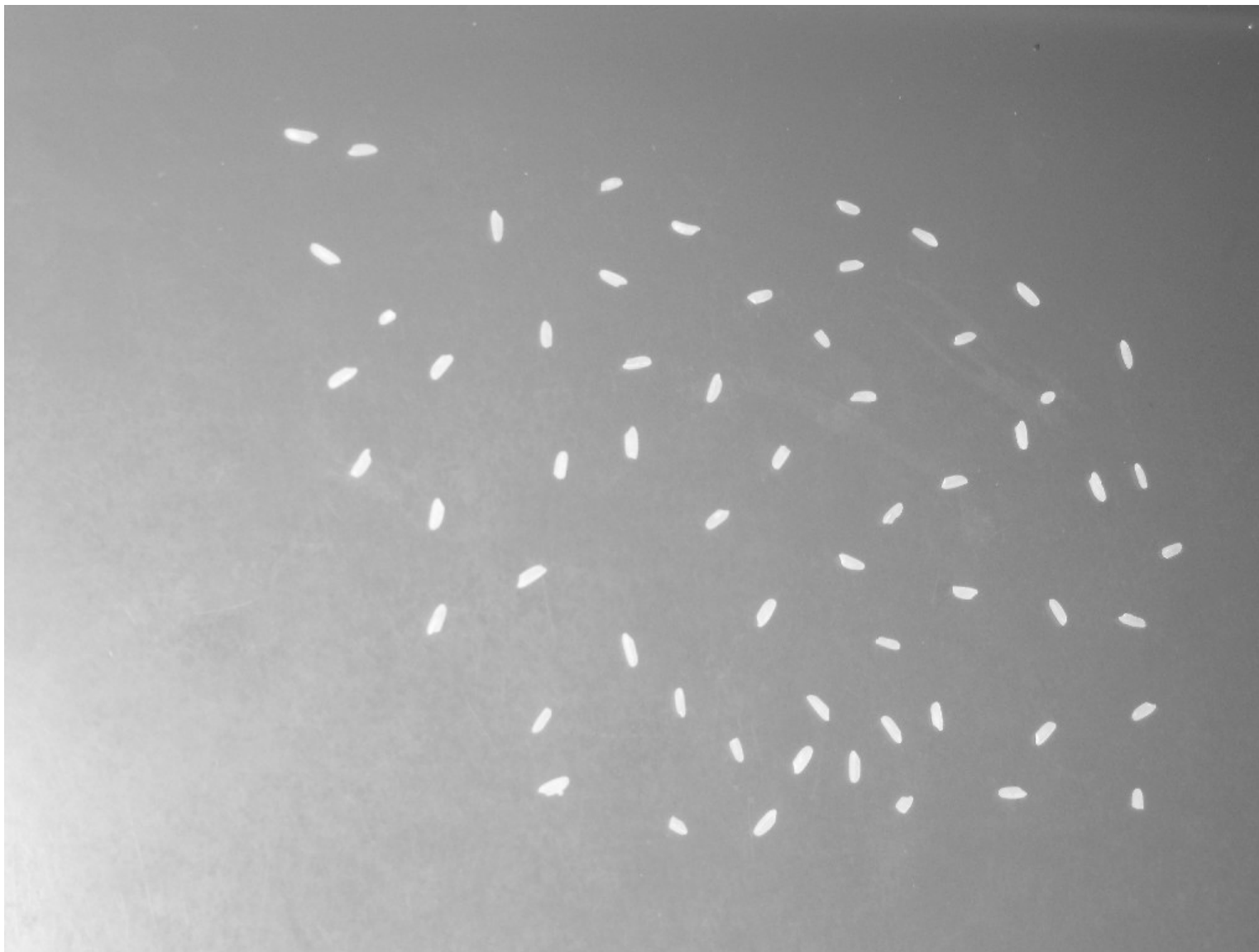
Outras alternativas

- A média de uma região quadrada é apenas uma das medidas que podemos usar na limiarização adaptativa.
 - Ela é normalmente usada por ter baixo custo computacional.
- Outras medidas podem ser usadas, como valores mínimos ou a mediana.
 - Funcionamento similar ao filtro da média: janela deslizando com regiões quadradas.
 - Vejamos alguns exemplos usando a mediana...
 - O objetivo aqui é fazer algo como “estimar o fundo” removendo da imagem filtrada os detalhes que queremos encontrar.
 - Quando fizermos a subtração da limiarização adaptativa, o que restará serão exatamente os objetos de interesse!

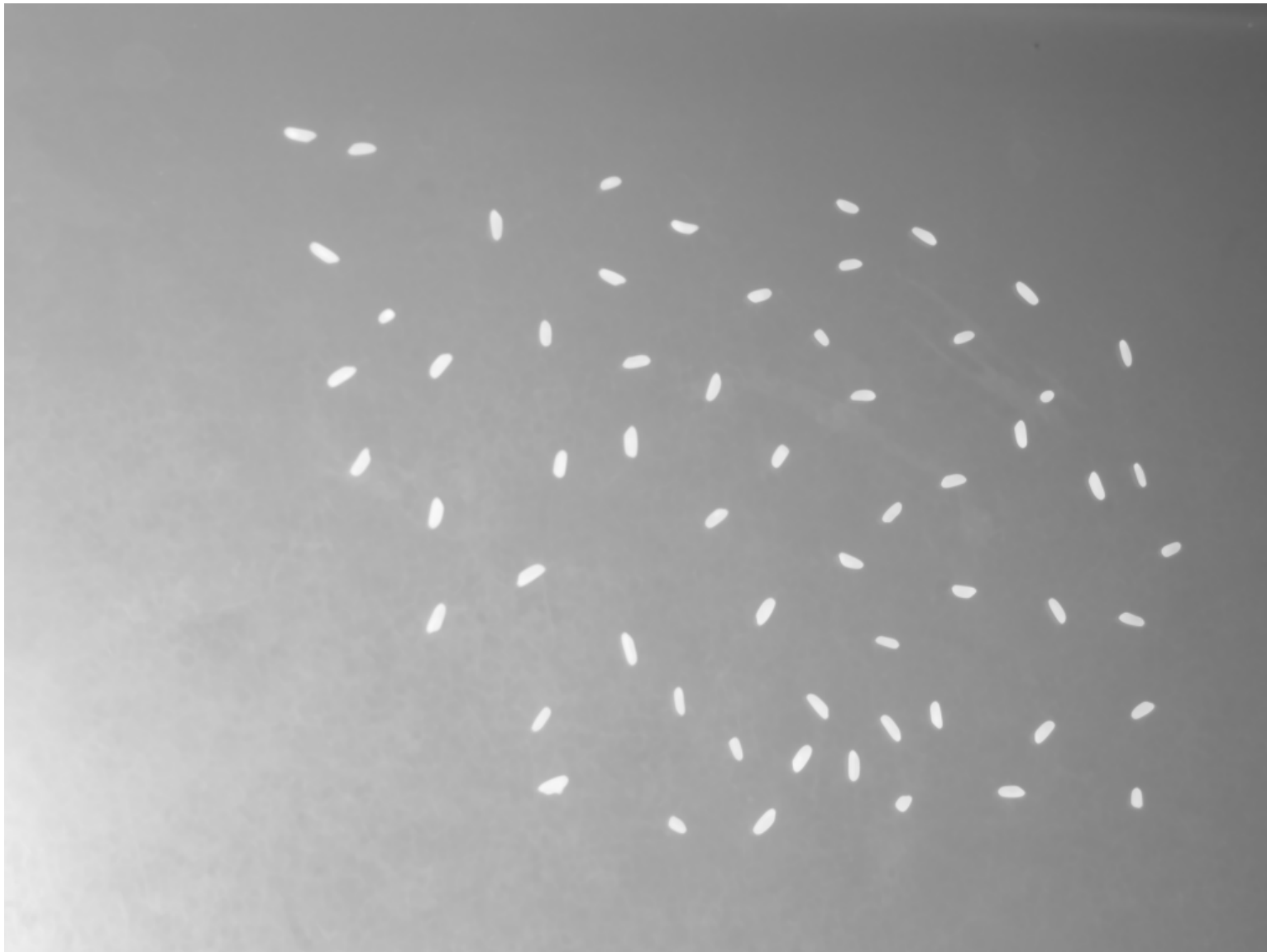
Filtro da mediana

- Não entraremos em muitos detalhes sobre como implementar o filtro da mediana, mas...
 - Como seria um algoritmo ingênuo?
 - O filtro da mediana é separável?
 - Como é o custo computacional, comparado ao filtro da média?
 - Como fica uma imagem filtrada pelo filtro da mediana?
- (Voltaremos a este filtro em outra aula).

Imagem original...



Após filtro da mediana 21x21



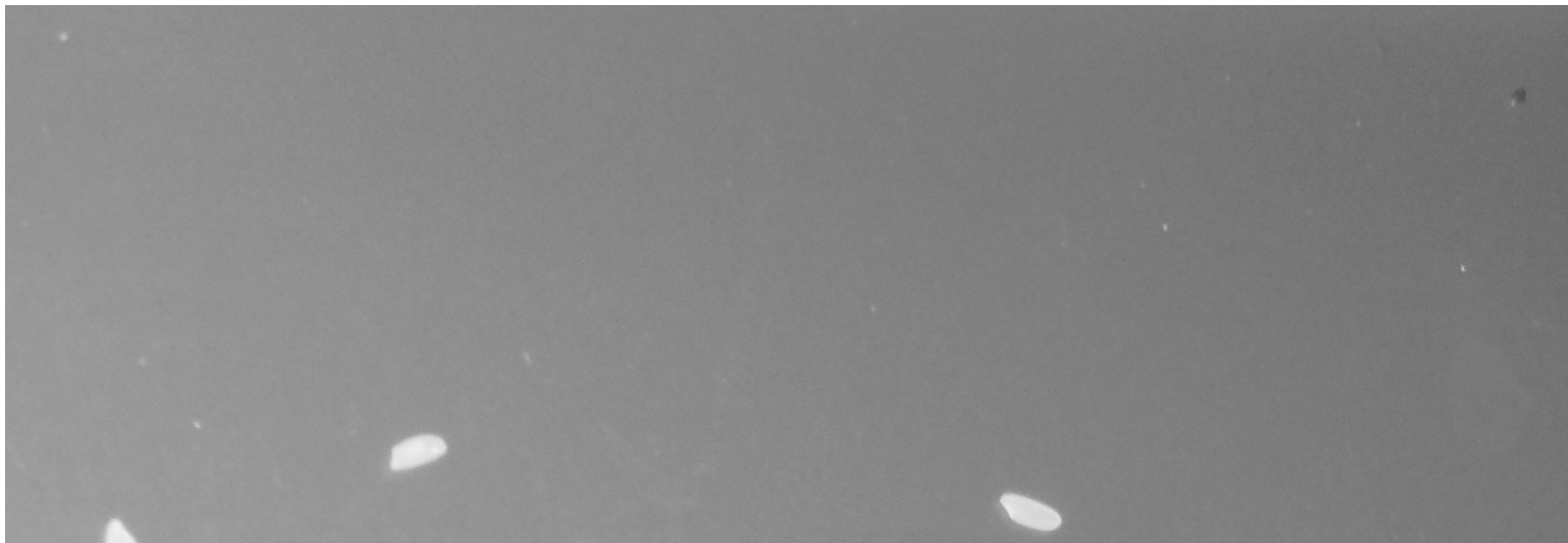


Imagem original



Após filtro da mediana 7x7



Após filtro da mediana 15x15



Após filtro da mediana 51x51

