Matlab Optimized Programming – v2.0

THE UNIVERSITY OF IOWA

```
% slow.m
A=zeros(10000);
tic
[xs ys] = size(A);

for i=1:xs
  for j=1:ys
    A(i,j)=A(i,j)+100;
  end
end
toc
```

```
% slow2.m
A=zeros(10000);
tic
[xs ys] = size(A);

for j=1:ys
  for i=1:xs
    A(i,j)=A(i,j)+100;
  end
end
toc
```

```
% fast.m
A=zeros(10000);
tic
A=A+100;
toc
```

+4276.5%

+455.8%

+687.4%

→ 19.647684 seconds.

→ 3.534881 seconds.

→ 0.448932 seconds.

Accessing matrix elements

```
>> a=magic(3)

a =
     8     1     6
     3     5     7
     4     9     2
```

```
>> a(:)

ans =
     8
     3
     4
     1
     5
     9
     6
     7
     2
```

```
>> a(ones(3)>0)

ans =
     8
     3
     4
     1
     5
     9
     6
     7
     2
```

```
>> a(a>4)

ans =
     8
     5
     9
     6
     7
```

Accessing matrix elements

```
>> a=magic(3)

a =
     8     1     6
     3     5     7
     4     9     2
```

```
>> a(diag(ones(3,1))>0)

ans =
     8
     5
     2
```

Conversion vector → matrix

```
>> a=magic(3)

a =
     8     1     6
     3     5     7
     4     9     2
```

```
>> b=a(ones(3)>0)

b =
     8
     3
     4
     1
     5
     9
     6
     7
     2
```

```
>> c = reshape(b, [3 3])

c =
     8     1     6
     3     5     7
     4     9     2
```

Concatenation of vectors

```
>> a=[3 5 0]'

a =
     3
     5
     0
```

```
>> b=[9 7 1]'

b =
     9
     7
     1
```

```
>> A=[a b]

A =
     3     9
     5     7
     0     1
```

```
>> A=[a' ; b']

A =
     3     5     0
     9     7     1
```

Sums along the dimension of a Matrix

The University of Iowa

```
>> A=[a' ; b']

A =
     3     5     0
     9     7     1
```

```
>> sum(A,1)

ans =
    12    12     1
```

```
>> sum(A,2)

ans =
     8
    17
```

→ “min” and “max” functions work similar; see help

Reading, Writing, and Displaying Images

The University of Iowa

- Read an image:

  >> I = imread('moon.tif');

- Display an Image:

  >> imshow(I, [])

  >> imshow(log(double(I)+1), [])

- Write an image:

  >> imwrite(I, 'moon2.png')

Exploring Image Data

The University of Iowa

- min/max value of image:

  >> min(I(:))

  >> max(I(:))

- Show coordinates and gray-value of a pixel

  >> impixelinfo

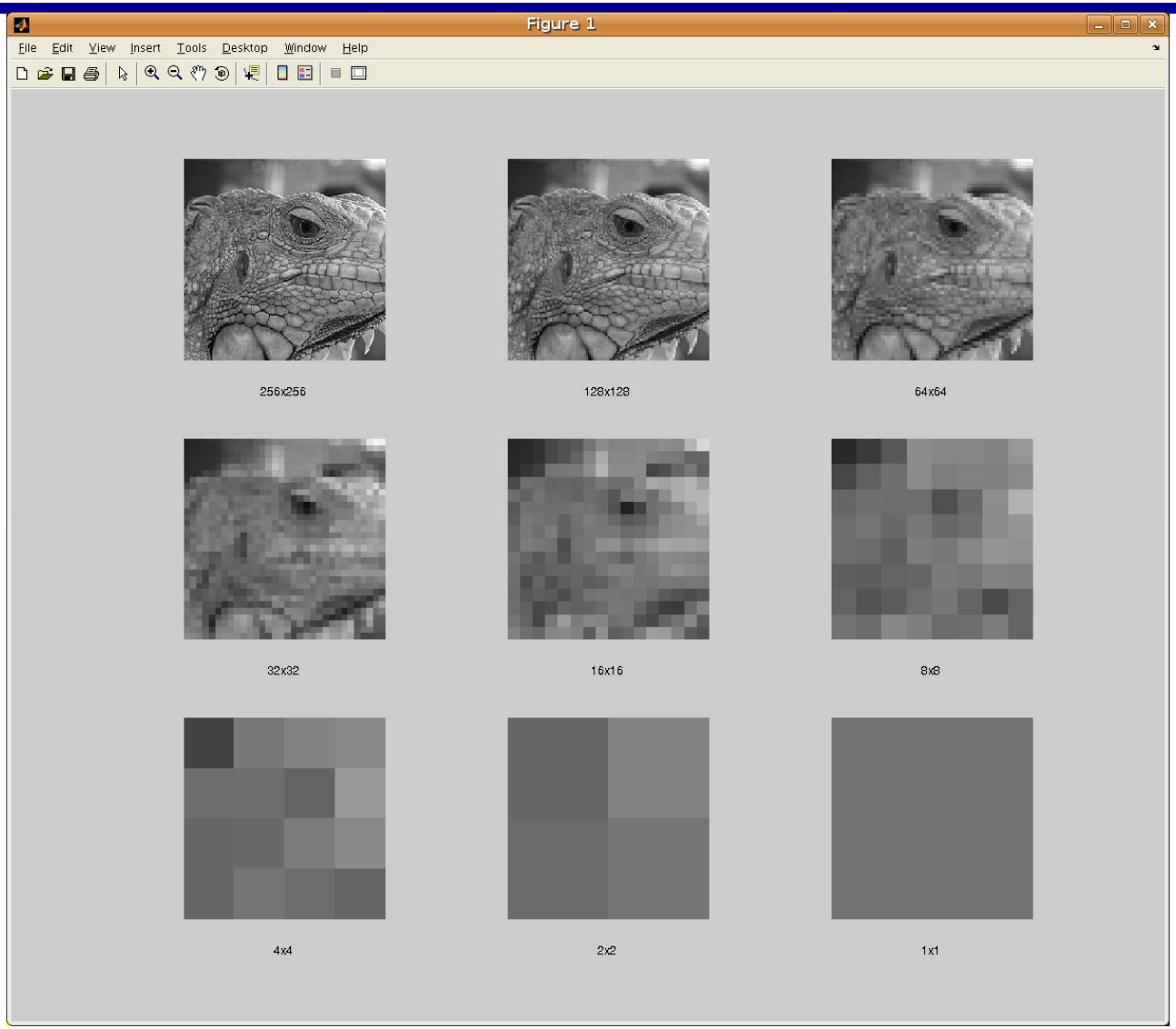
- Measure distances (Euclidean)

  >> imdistline

- Display a gray-value profile:

  >> improfile

Example: Sampling



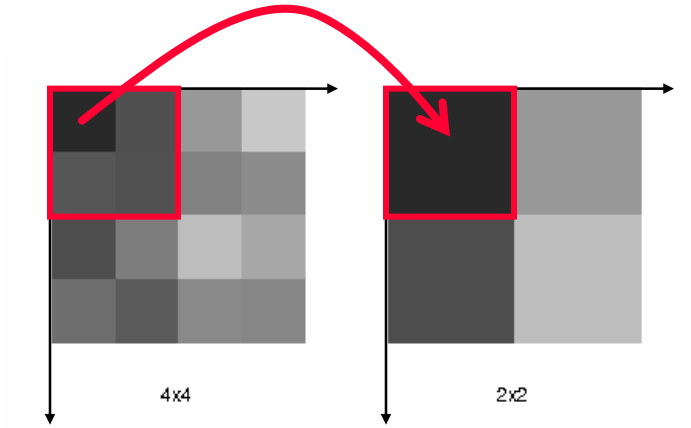
Sampling Functions

THE UNIVERSITY OF IOWA

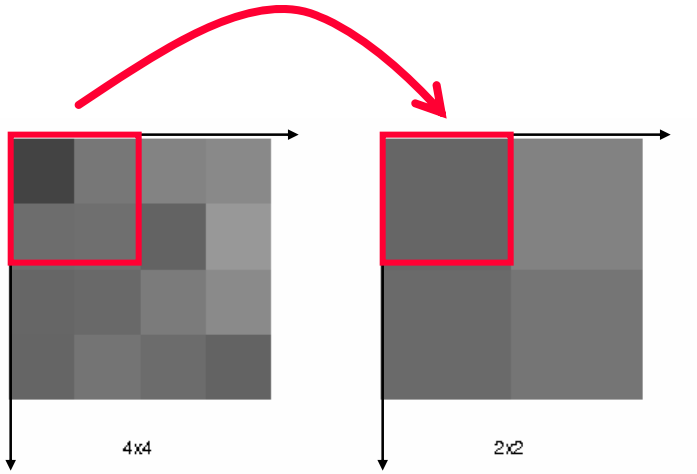
sample_im1.m

sample_im2.m

upper left pixel



mean value



Test Framework: "test_sample.m"

THE UNIVERSITY OF IOWA

```
close all

flag=1;

I = imread('after-384x288.jpg');
I = I(1:256, 1+50:256+50);

if flag
  subplot(3,3,1)
else
  figure(1)
end
imshow(I,[0 255])
xlabel( sprintf('%dx%d', size(I,1), size(I,2) ))

pause
```

```
for i=1:8
   I2 = sample_im1(I);

   if flag
     subplot(3,3,i+1)
   else
     figure(i+1)
   end
   imshow(I2,[0 255])
   xlabel( sprintf('%dx%d', size(I2,1), size(I2,2) ))

   I=I2;

   pause
end
```

Solution 1: ”sample_im1.m”

THE UNIVERSITY OF IOWA

```
function I2 = sample_im1(I)

[xs ys] = size(I);

if( xs~=ys )
  error(‘X- and Y-size must be the same!')
end

if( mod(xs,2)~=0 )
  error(‘X- and Y-size must be even!')
end

a = mod(1:xs, 2);
A = (a'*a)>0;
I2 = reshape(I(A), [xs/2 ys/2]);
```

Solution 2: “sample_im2.m”

The University of Iowa

```
function I2 = sample_im2(I)

[xs ys] = size(I);
…
a = mod(1:xs, 2);
b = abs(a-1);

A1 = (a'*a)>0;
A2 = (a'*b)>0;
A3 = (b'*a)>0;
A4 = (b'*b)>0;

S = mean( [I(A1) I(A2) I(A3) I(A4)], 2);

I2 = reshape(S, [xs/2 ys/2]);
```

Solution 2: “sample_im2.m”

A1 =

1	0	1	0	1	0
0	0	0	0	0	0
1	0	1	0	1	0
0	0	0	0	0	0
1	0	1	0	1	0
0	0	0	0	0	0

A2 =

0	1	0	1	0	1
0	0	0	0	0	0
0	1	0	1	0	1
0	0	0	0	0	0
0	1	0	1	0	1
0	0	0	0	0	0

A3 =

0	0	0	0	0	0
1	0	1	0	1	0
0	0	0	0	0	0
1	0	1	0	1	0
0	0	0	0	0	0
1	0	1	0	1	0

A4 =

0	0	0	0	0	0
0	1	0	1	0	1
0	0	0	0	0	0
0	1	0	1	0	1
0	0	0	0	0	0
0	1	0	1	0	1