



北京交通大学

# 图像处理与机器学习

Digital Image Processing and Machine Learning

主讲人：黄琳琳

电子信息工程学院



# 第三章 形态学处理

- ◆ 基本概念
- ◆ 集合论基础
- ◆ 数学形态学处理



# 数学形态学处理

- ◆ 膨胀 (dilation)
- ◆ 腐蚀 (erosion)
- ◆ 开运算 (opening)
- ◆ 闭运算 (Closing)

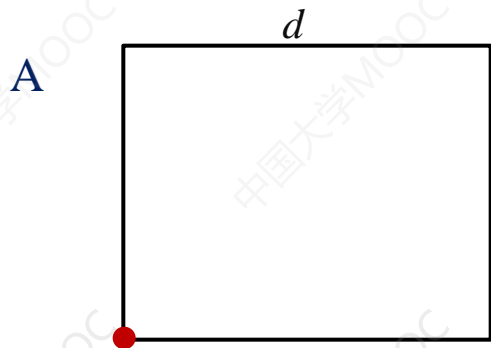


# 数学形态学处理

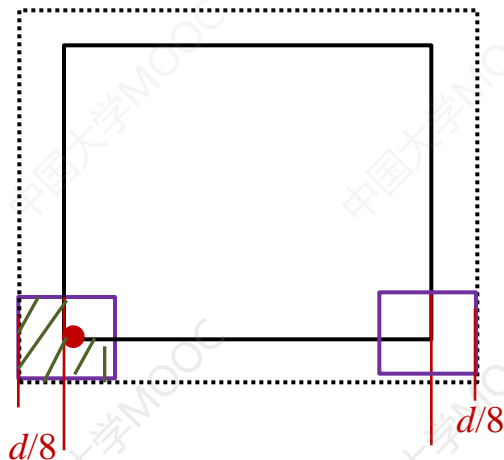
## ◆ 膨胀 (dilation)

-- 集合A与B, A被B膨胀的定义  $A \oplus B = \left\{ z \mid \left( \overset{\wedge}{B} \right)_z \cap A \neq \phi \right\}$

对B的反射  $\hat{B}$  进行**平移**, 使之与A的交集不为空的**点的集合**



保证B的反射的平移  
与A的交集不为空



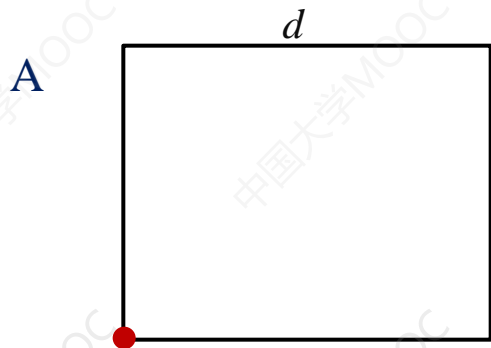


# 数学形态学处理

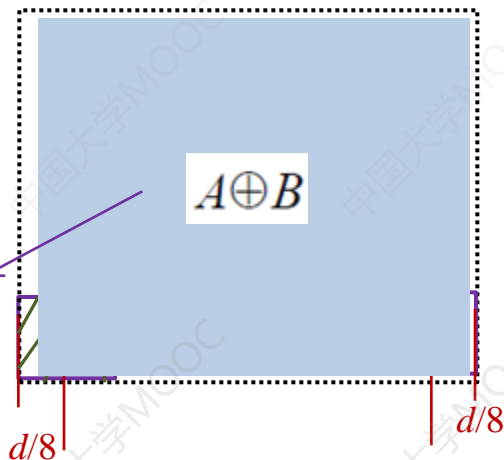
## ◆ 膨胀 (dilation)

-- 集合A与B, A被B膨胀的定义  $A \oplus B = \left\{ z \mid \left( \overset{\wedge}{B} \right)_z \cap A \neq \phi \right\}$

对B的反射  $\hat{B}$  进行**平移**, 使之与A的交集不为空的**点的集合**



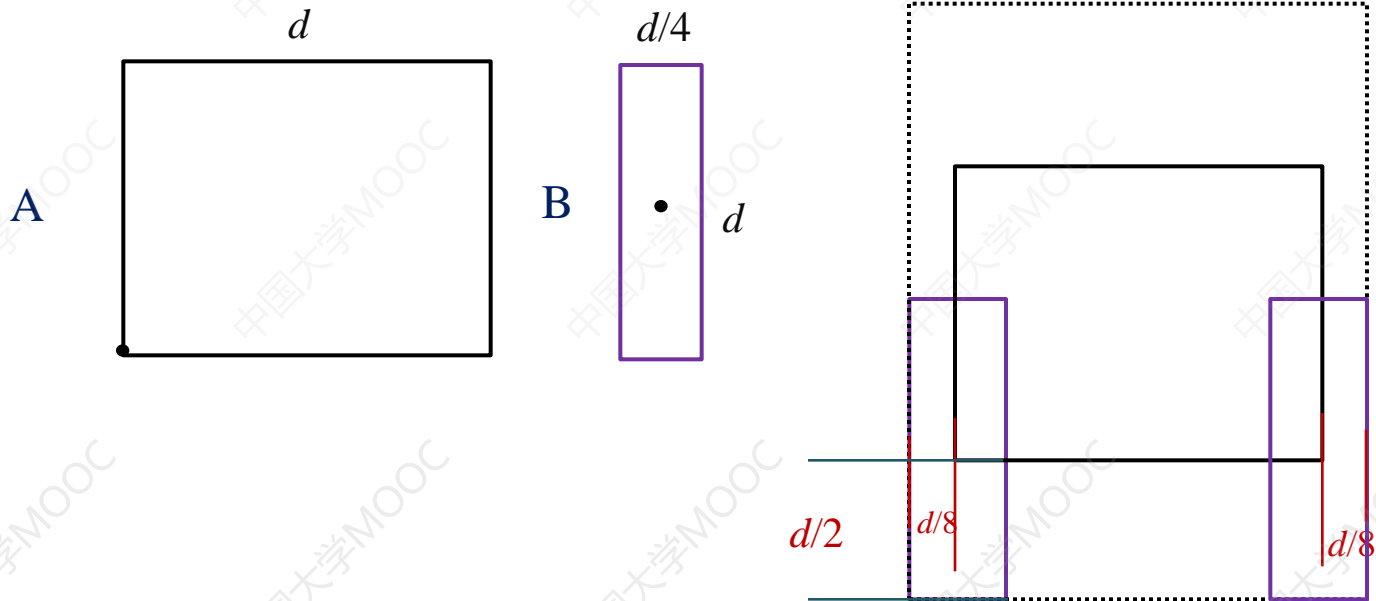
保证B的反射的平移  
与A的交集不为空





## -- 集合A与B， A被B膨胀的定义

$$A \oplus B = \left\{ z \mid \left( \overset{\wedge}{B} \right)_z \cap A \neq \phi \right\}$$



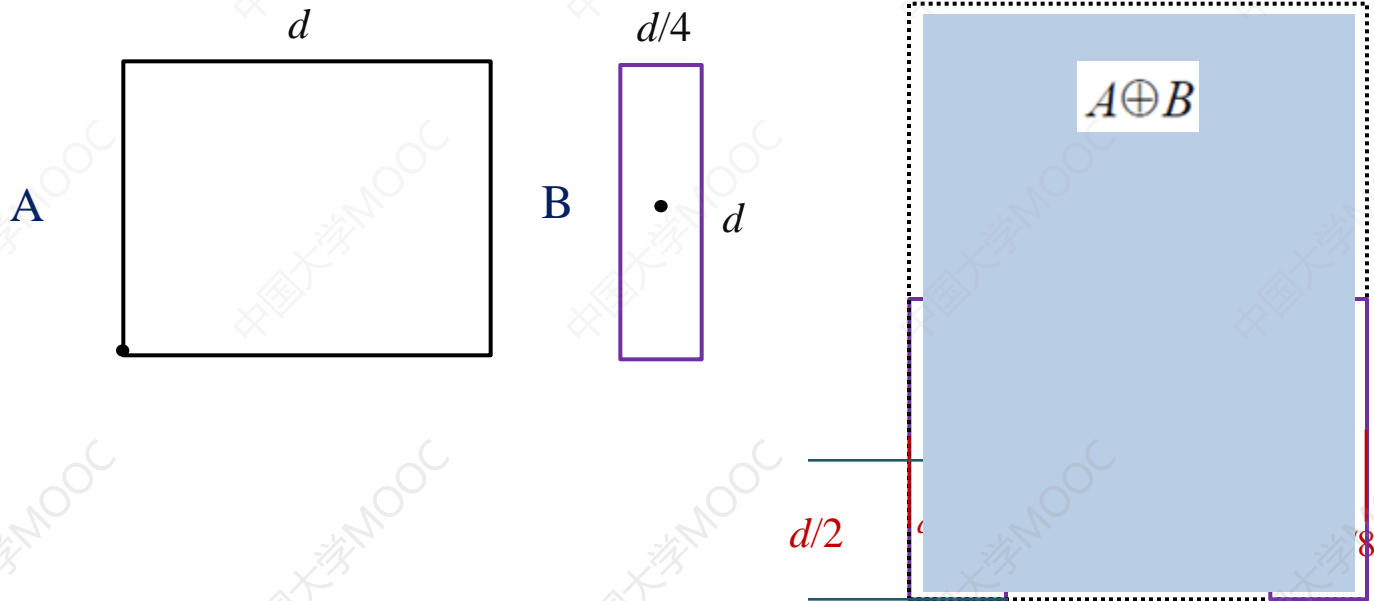


# 数学形态学处理

## ◆ 膨胀 (dilation)

-- 集合A与B, A被B膨胀的定义

$$A \oplus B = \left\{ z \mid \left( \hat{B} \right)_z \cap A \neq \phi \right\}$$

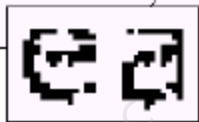




# 数学形态学处理

## ◆ 膨胀 (dilation)

Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.



A



B



$A \oplus B$



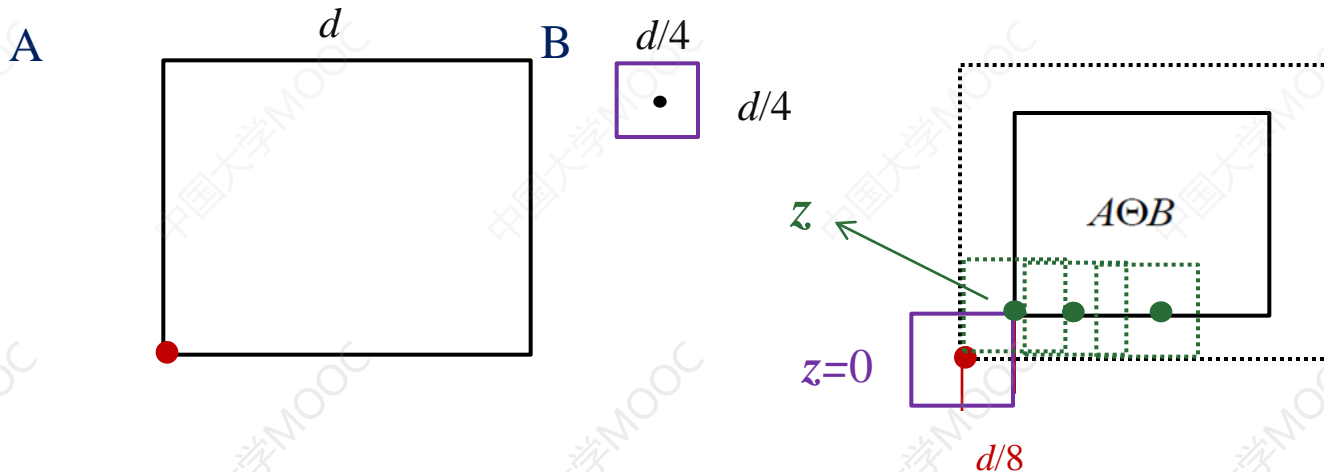


# 数学形态学处理

## ◆ 腐蚀 (dilation and erosion)

-- 集合A与B, A被B腐蚀的定义  $A \ominus B = \{z \mid (B)_z \subseteq A\}$

将B平移  $z$  之后,  $(B)_z$  被包含在A的点的集合



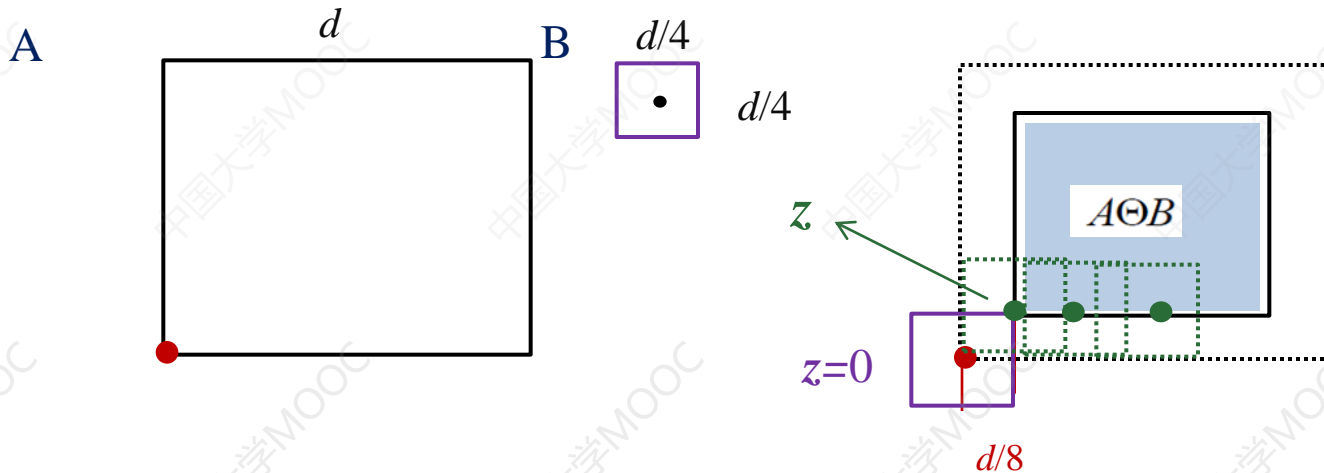


# 数学形态学处理

## ◆ 腐蚀 (dilation and erosion)

-- 集合A与B，A被B腐蚀的定义  $A \ominus B = \{z \mid (B)_z \subseteq A\}$

将B平移  $z$  之后， $(B)_z$  被包含在A的点的集合



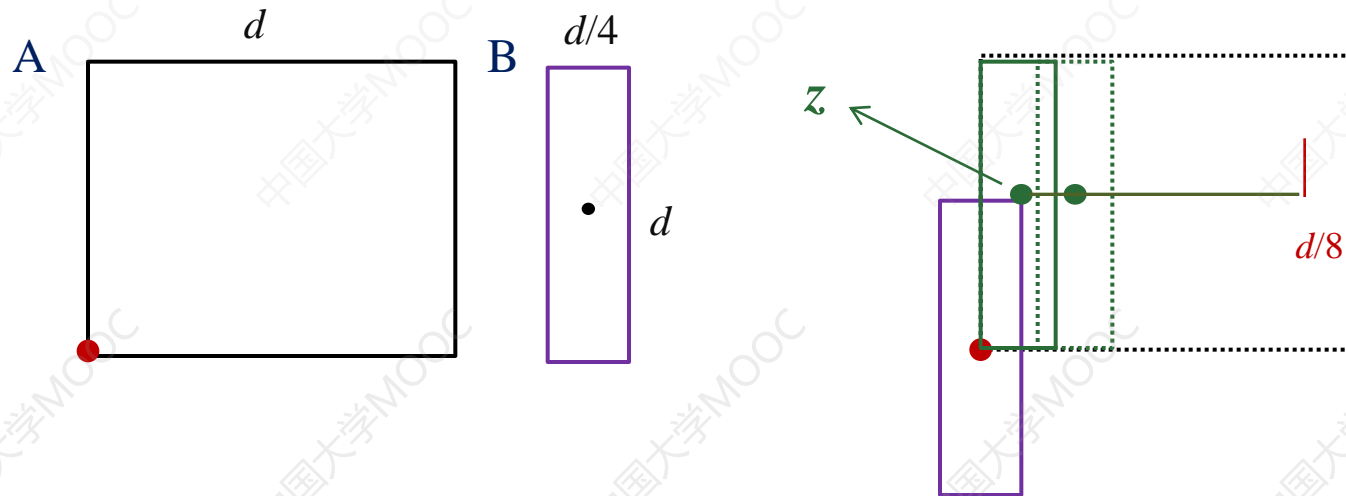


# 数学形态学处理

## ◆ 腐蚀 (dilation and erosion)

-- 集合A与B, A被B腐蚀的定义  $A \ominus B = \{z \mid (B)_z \subseteq A\}$

将B平移  $z$  之后,  $(B)_z$  被包含在A的点的集合



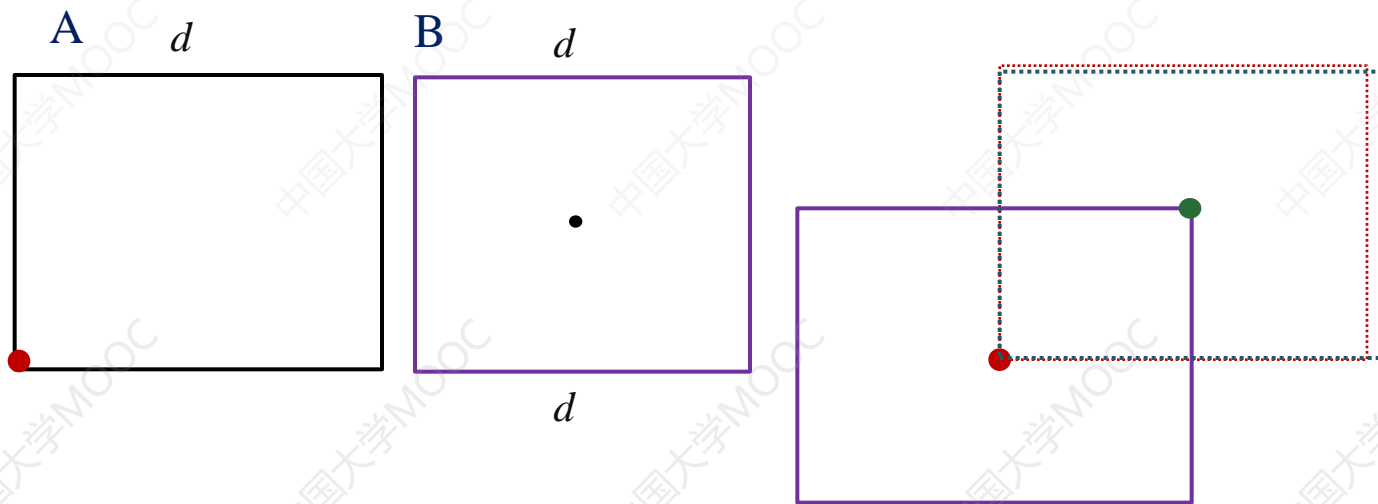


# 数学形态学处理

## ◆ 腐蚀 (dilation and erosion)

-- 集合A与B，A被B腐蚀的定义  $A \ominus B = \{z \mid (B)_z \subseteq A\}$

将B平移  $z$  之后， $(B)_z$  被包含在A的点的集合

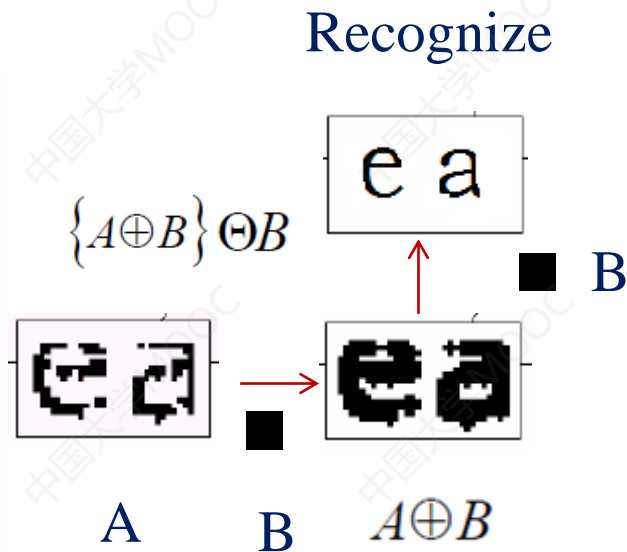
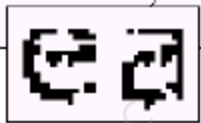




# 数学形态学处理

## ◆ 膨胀 (dilation)

Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.

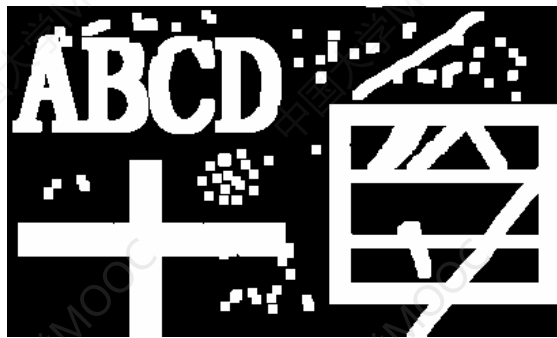




# 数学形态学处理

## ◆ 膨胀

- 将与物体接触所有背景点合并到该物体中
- 使边界向外部扩张的过程
- 可以用来填补物体中的空洞





# 数学形态学处理

## ◆ 腐蚀

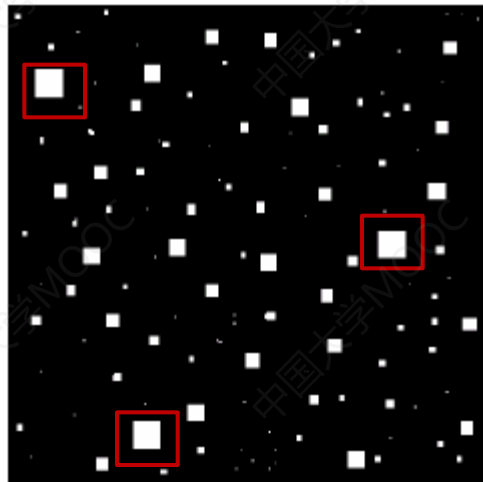
- 一种消除边界点，
- 使边界向内部收缩的过程
- 用来消除小且无意义的物体





# 数学形态学处理

包含边长为1,3,5,7,9  
和15像素正方形



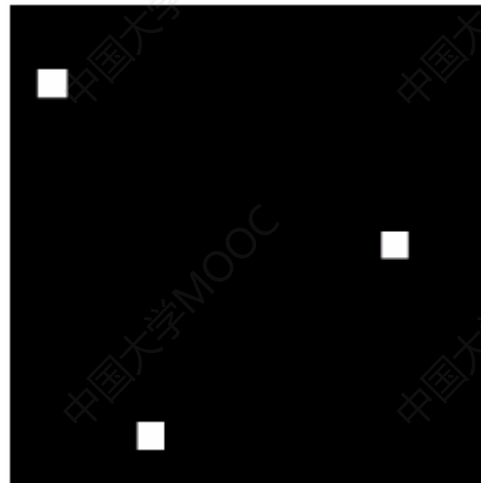
图a 二值图像

使用 $13 \times 13$ 像素大小  
的结构元素腐蚀原图



图b a被腐蚀图像

使用 $13 \times 13$ 像素大小的  
结构元素膨胀图b



图c b被膨胀图像

使用腐蚀消除图像的细节部分，产生滤波器的作用





# 数学形态学处理

## ◆ 图像形态学处理运算实现

-- 二值图像：0,1代表两个不同的灰度级

-- 逻辑运算：与、或、非

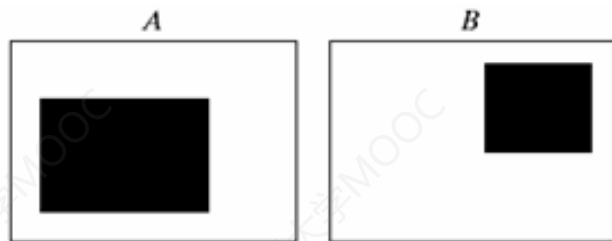
$p$	$q$	$p \text{ AND } q$ (also $p \cdot q$ )	$p \text{ OR } q$ (also $p + q$ )	NOT ( $p$ ) (also $\bar{p}$ )
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0



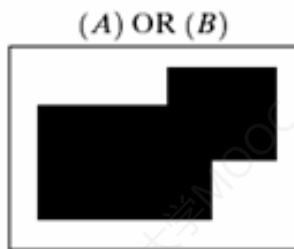
# 数学形态学处理

## ◆ 二值图像的逻辑运算

-- 集合A与B并集、交集、补集

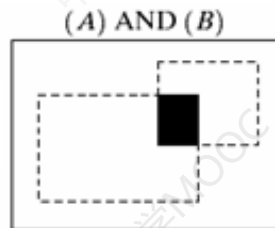


或运算 (or)



膨胀

与运算 (and)



腐蚀



# 数学形态学处理

## ◆ 形态学运算

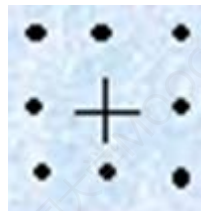
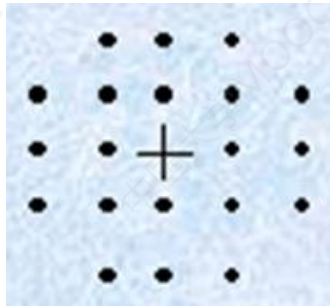
- 邻域运算
- 结构元素：特殊定义的邻域 (structure element)
- 在每个像素位置上, 结构元素与二值图像对应的区域
- 进行特定的逻辑运算
- 逻辑运算的结果为输出图像的相应像素取值



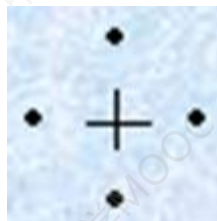
# 数学形态学处理

## ◆ 几种简单的结构元素

-- 圆形、方形、菱形等



1	1	1
1	1	1
1	1	1



0	1	0
1	1	1
0	1	0



# 数学形态学处理

## ◆ 腐蚀运算

-- 给定二值图像  $I(x, y)$

-- 结构元素的二值模板  $T(i, j)$

1	1	1
1	1	1
1	1	1

-- 输出二值图像  $E(x, y)$

$$E(x, y) = (I \odot T)(x, y) = \bigwedge_{i=0, j=0}^{m, n} [I(x+i, y+j) \& T(i, j)]$$



# 数学形态学处理

## ◆ 腐蚀运算的编程实现

$$E(x, y) = (I \odot T)(x, y) = \bigwedge_{i=0, j=0}^{m, n} [I(x+i, y+j) \& T(i, j)]$$



图形点(gray=255)在其3x3 邻域  
只要有若干个背景点  
则该点设为背景点(0)。





# 数学形态学处理

## ◆ 膨胀运算

-- 给定二值图像  $I(x, y)$

-- 结构元素的二值模板  $T(i, j)$

-- 输出二值图像  $E(x, y)$

$$E(x, y) = (I \oplus T)(x, y) = \underset{i=0, j=0}{\overset{m, n}{OR}} [I(x+i, y+j) \& T(i, j)]$$



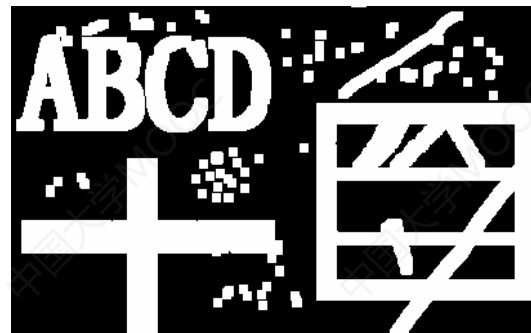
# 数学形态学处理

## ◆ 膨胀运算的编程实现

$$E(x, y) = (I \oplus T)(x, y) = \bigvee_{i=0, j=0}^{m, n} [I(x+i, y+j) \& T(i, j)]$$



背景点(gray=0)在其3x3 邻域  
只要有若干个图形点  
则该点设为图形点(255)。







# 谢 谢

本课程所引用的一些素材为主讲老师多年的教学积累，来源于多种媒体及同事和同行的交流，难以一一注明出处，特此说明并表示感谢！