

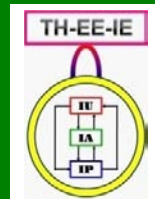
图象工程（上）

图 象 处 理

（第3版）

章毓晋

清华大学电子工程系 100084 北京



第2单元 图象恢复

- 第5章 图象消噪和恢复
- 第6章 图象校正和修补
- 第7章 图象投影重建

图象恢复认为图象（质量）在某种情况/条件下退化或恶化了，现在需要根据相应的退化模型和知识重建或恢复原始的图象。图象恢复技术要将图象退化的过程模型化，并根据确定的图象退化模型对原始图象进行复原，以获得原来期望的效果。



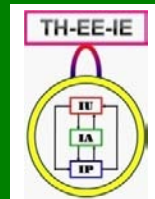
第6章 图象校正和修补

6.1 图象仿射变换

6.2 几何失真校正

6.3 图象修复

6.4 区域填充



6.1 图象仿射变换

{基本坐标变换的扩展}

6.1.1 一般仿射变换

6.1.2 特殊仿射变换

6.1.3 变换间的联系



6.1.1 一般仿射变换

1. 定义

一个非奇异线性变换接一个平移变换
矩阵形式

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

分块矩阵形式

$$\mathbf{v}' = \mathbf{H}_A \mathbf{v} = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{v}$$

一个平面上的仿射变换有6个自由度

6.1.1 一般仿射变换

1. 定义

$$A_1 = \begin{bmatrix} 1 & 1/2 \\ 1/2 & 1 \end{bmatrix}$$

$$t_1 = \begin{bmatrix} 4 \\ -2 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 1/2 & 1 \\ 1 & 1 \end{bmatrix}$$

$$t_2 = \begin{bmatrix} -2 \\ 1 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} 3/2 & 1/2 \\ 1/2 & 1 \end{bmatrix}$$

$$t_3 = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$$

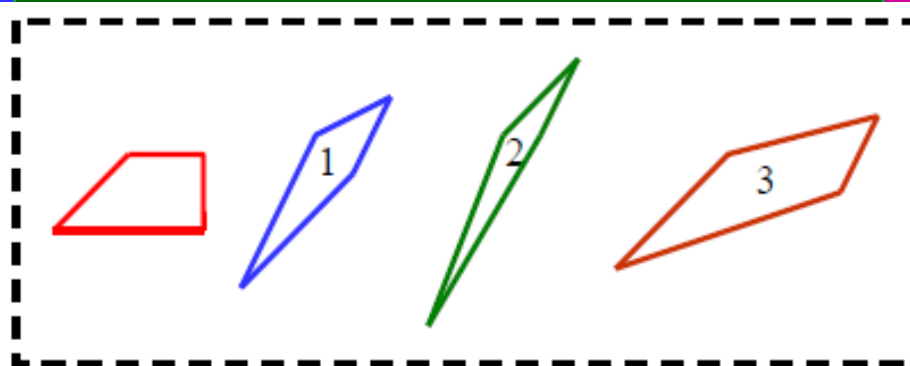


图 6.1.1 对多边形目标进行仿射变换得到的结果

6.1.1 一般仿射变换

2. 分解

可以把A分解成两个基本变换的组合：一个非各向同性放缩和一个旋转

$$A = R(\theta)R(-\phi)DR(\phi)$$

$$D = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

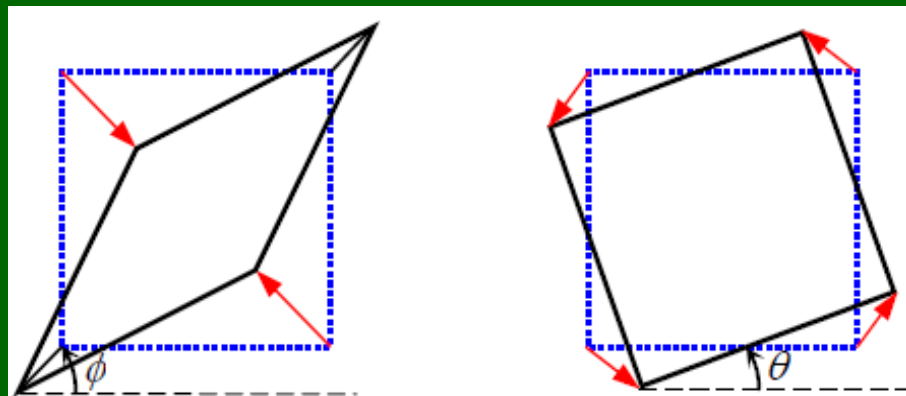
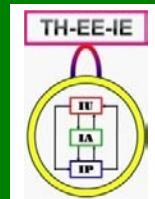


图 6.1.2 仿射变换的分解结果



6.1.1 一般仿射变换

2. 分解

平移、放缩、旋转和剪切变换的一种综合

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} 1 & J_x \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}$$

$$\begin{aligned} R &= \begin{bmatrix} 1 & J_x \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \\ &= \begin{bmatrix} 1 & J_x + \tan(\theta/2) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\sin \theta & 1 \end{bmatrix} \begin{bmatrix} 1 & \tan(\theta/2) \\ 0 & 1 \end{bmatrix} \end{aligned}$$

$$\begin{cases} S_x = \sqrt{a_{11}^2 + a_{21}^2} \\ S_y = \frac{\det(A)}{S_x} \\ J_x = \frac{a_{11}a_{12} + a_{21}a_{22}}{\det(A)} \\ \tan \theta = -\frac{a_{21}}{a_{11}} \end{cases}$$

6.1.1 一般仿射变换

3. 系数

从 (x, y) 到 (x', y') 和从 (x', y') 到 (x, y) 的变换

$$\begin{aligned} x' &= S_x x + J_x y + T_x \\ y' &= J_y x + S_y y + T_y \end{aligned}$$

$$\begin{bmatrix} x' - T_x \\ y' - T_y \end{bmatrix} = \begin{bmatrix} S_x & J_x \\ J_y & S_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{1}{D} \begin{bmatrix} S_y & -J_x \\ -J_y & S_x \end{bmatrix} \begin{bmatrix} x' - T_x \\ y' - T_y \end{bmatrix}$$

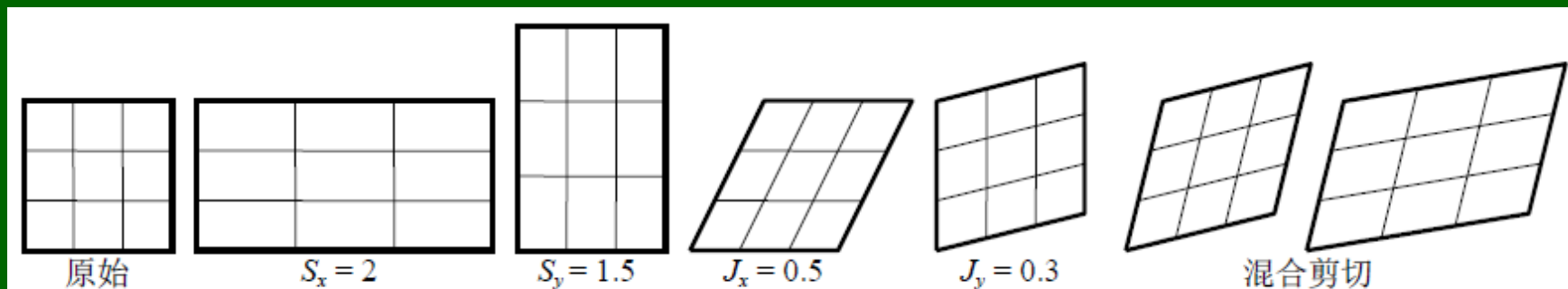


图 6.1.3 4 个系数 S_x 、 S_y 、 J_x 和 J_y 在仿射变换中的作用示意



6.1.1 一般仿射变换

4. 性质

- (1) 仿射变换将有限点映射为有限点，即仿射变换能建立一对一的关系
- (2) 仿射变换仍将直线映射为直线
- (3) 仿射变换将平行直线映射为平行直线
- (4) 当区域 P 和 Q 是没有退化的三角形（即面积不为0），那么存在一个惟一的仿射变换 A 可将 P 映射为 Q ，即 $Q = A(P)$ ——3点映射变换
- (5) 仿射变换会导致区域面积的变化

6.1.1 一般仿射变换

4. 性质

(5) 仿射变换会导致区域面积的变化

- 仿射变换可分解为包括剪切变换的多个变换
- 正方形受到沿X方向的剪切作用后变成菱形
- 在短对角线方向有所压缩而在长对角线方向有所拉伸

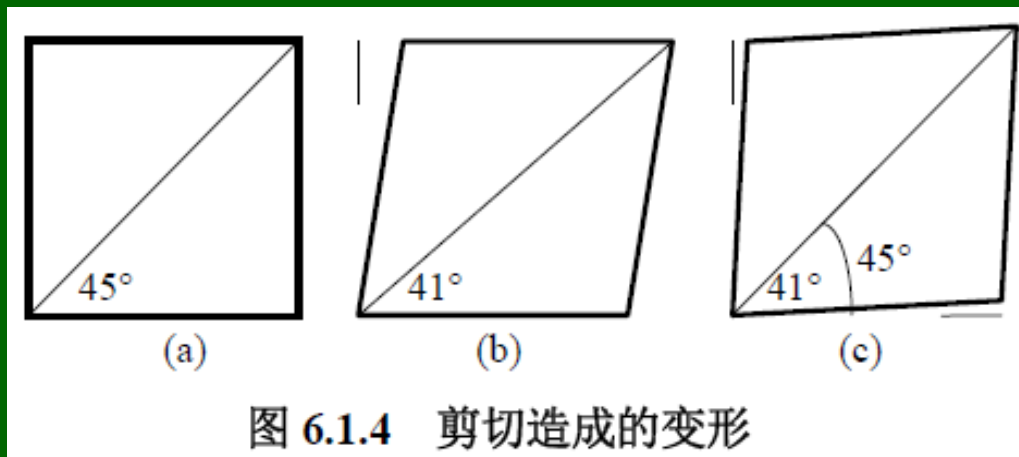


图 6.1.4 剪切造成的变形

6.1.1 一般仿射变换

4. 性质

(5) 仿射变换会导致区域面积的变化

- 剪切后单方向的伸缩会导致面积变化

$$S_{\text{area}} = S_x S_y - J_x J_y$$

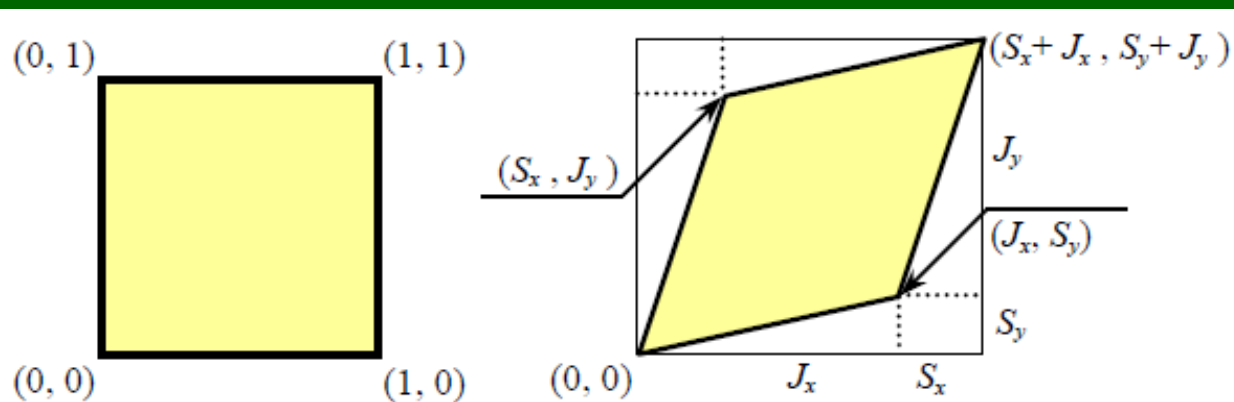


图 6.1.5 剪切前和剪切后的正方形

6.1.2 特殊仿射变换

1. 相似变换

矩阵表达

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s \cos \theta & s \sin \theta & t_x \\ -s \sin \theta & s \cos \theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

分块矩阵形式

$$\mathbf{v}' = \mathbf{H}_S \mathbf{v} = \begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{v}$$

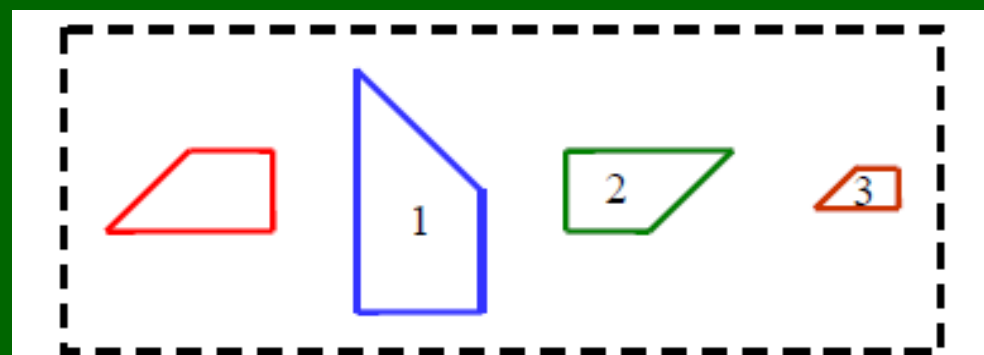


图 6.1.6 对区域进行相似变换得到的结果

6.1.2 特殊仿射变换

2. 等距变换 (刚体变换)

矩阵表达

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} e \cos \theta & \sin \theta & t_x \\ -e \sin \theta & \cos \theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

分块矩阵形式

$$\mathbf{v}' = \mathbf{H}_I \mathbf{v} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{v}$$

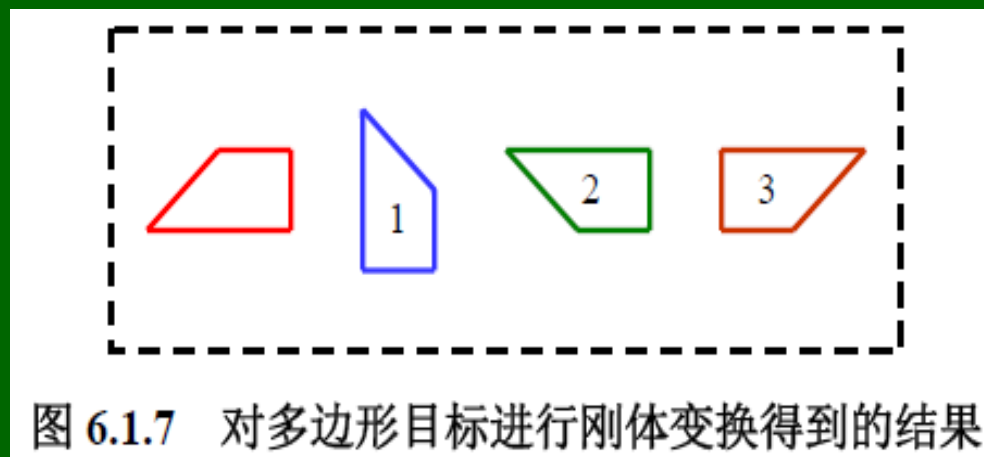


图 6.1.7 对多边形目标进行刚体变换得到的结果

6.1.2 特殊仿射变换

3. 欧氏变换

矩阵表达

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & t_x \\ -\sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

分块矩阵形式

$$\mathbf{v}' = \mathbf{H}_E \mathbf{v} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{v}$$

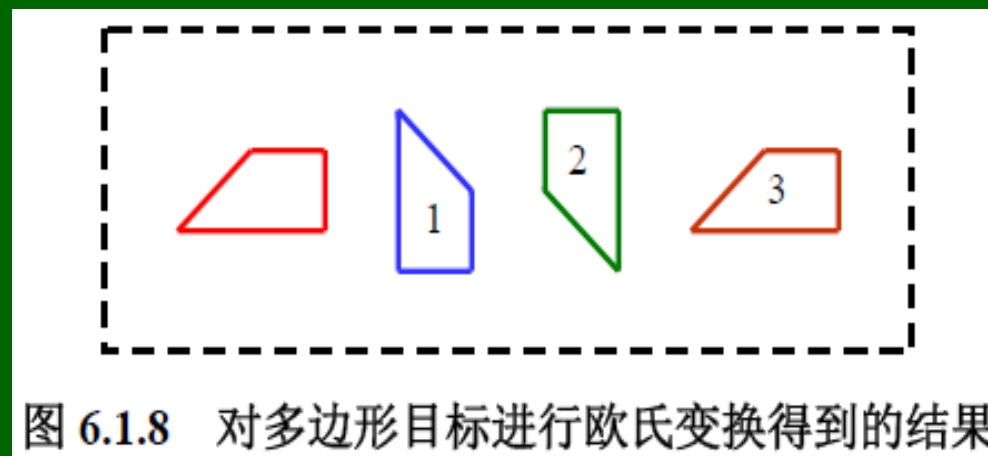


图 6.1.8 对多边形目标进行欧氏变换得到的结果

6.1.3 变换间的联系

1. 变换的层次

构成一个体系
逐步专门化（或者说它们依次退化）
下一层变换是
上一层变换的一个特例

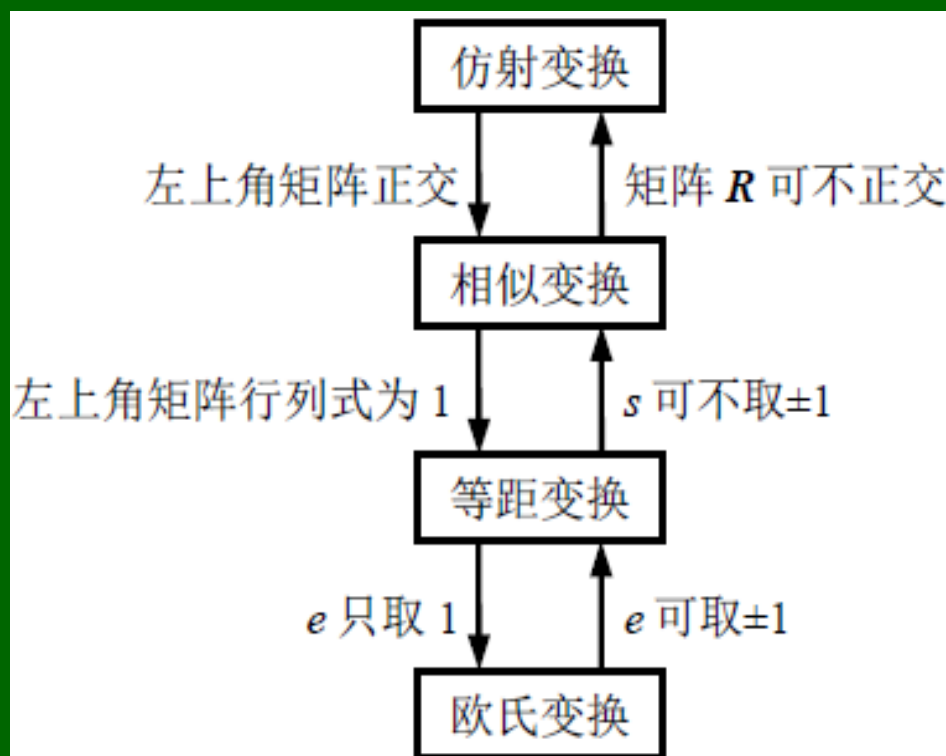
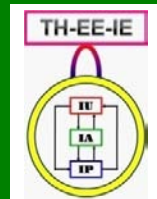


图 6.1.9 四种变换的层次体系



6.2 几何失真校正

{既对象素进行重新排列以恢复原空间关系，
也对空间变换后的象素赋予相应的灰度值以恢复
原位置的灰度值}

6.2.1 空间变换

6.2.2 灰度插值



6.2.1 空间变换

几何失真模型

图象 $f(x, y)$ 受几何形变的影响变成失真图象

$$g(x', y'): \quad x' = s(x, y) \quad y' = t(x, y)$$

线性失真

$$s(x, y) = k_1 x + k_2 y + k_3$$

$$t(x, y) = k_4 x + k_5 y + k_6$$

二次失真

(非线性)

$$s(x, y) = k_1 + k_2 x + k_3 y + k_4 x^2 + k_5 xy + k_6 y^2$$

$$t(x, y) = k_7 + k_8 x + k_9 y + k_{10} x^2 + k_{11} xy + k_{12} y^2$$

6.2.1 空间变换

约束对应点方法

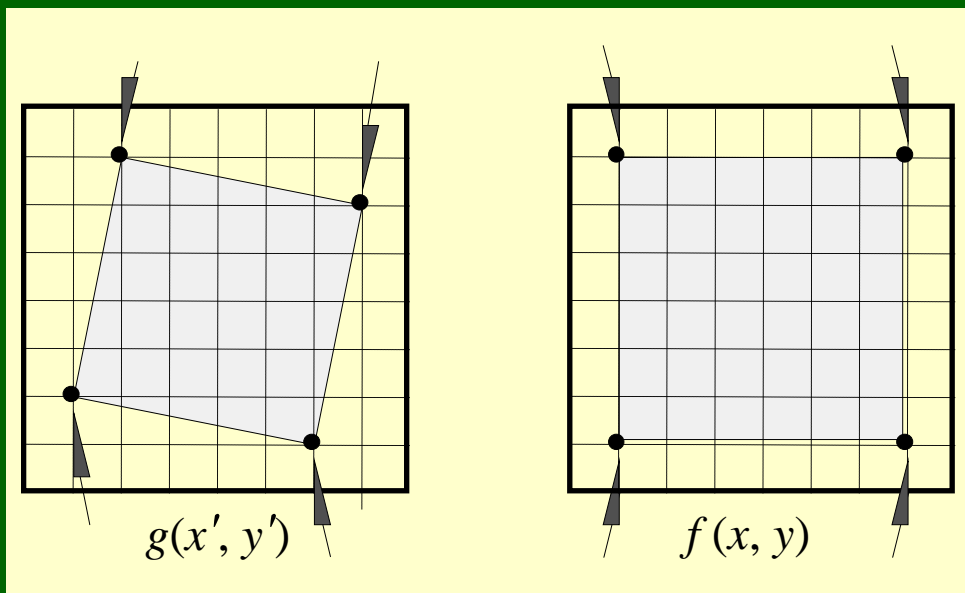
在输入图（失真图）和输出图（校正图）上找一些其位置确切知道的点，然后利用这些点建立两幅图间其它点空间位置的对应关系

选取四边形顶点

$$x' = k_1x + k_2y + k_3xy + k_4$$

$$y' = k_5x + k_6y + k_7xy + k_8$$

四组对应点解八个系数



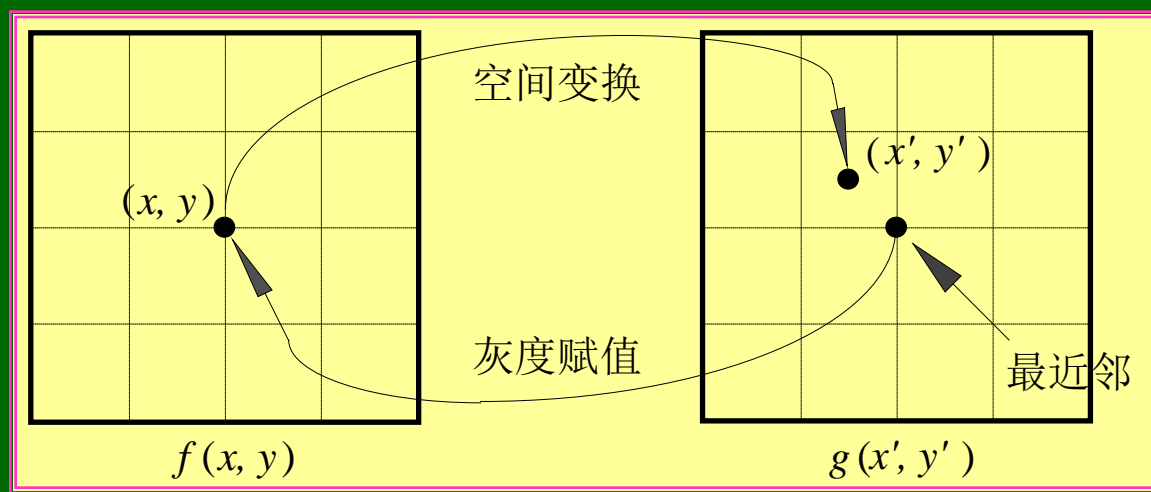
6.2.2 灰度插值

- ◆ (x, y) 的值总是整数，但 (x', y') 的值可能不是整数
- ◆ 用整数处的像素值来计算在非整数处的像素值

最近邻插值

也常称为零阶插值（计算量小，但精度差）

将离 (x', y') 点最近的像素的灰度值作为 (x', y') 点的灰度值赋给原图 (x, y) 处像素

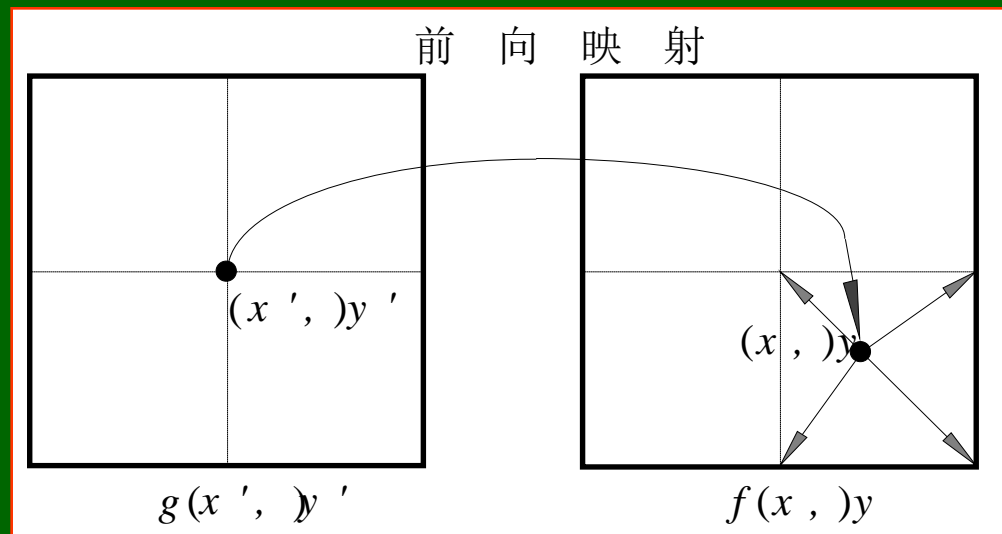


6.2.2 灰度插值

前向映射

一个失真图的象素映射到不失真图的四个象素之间
最后灰度是由许多失真图的象素的贡献之和所决定

失真图 →
非失真图



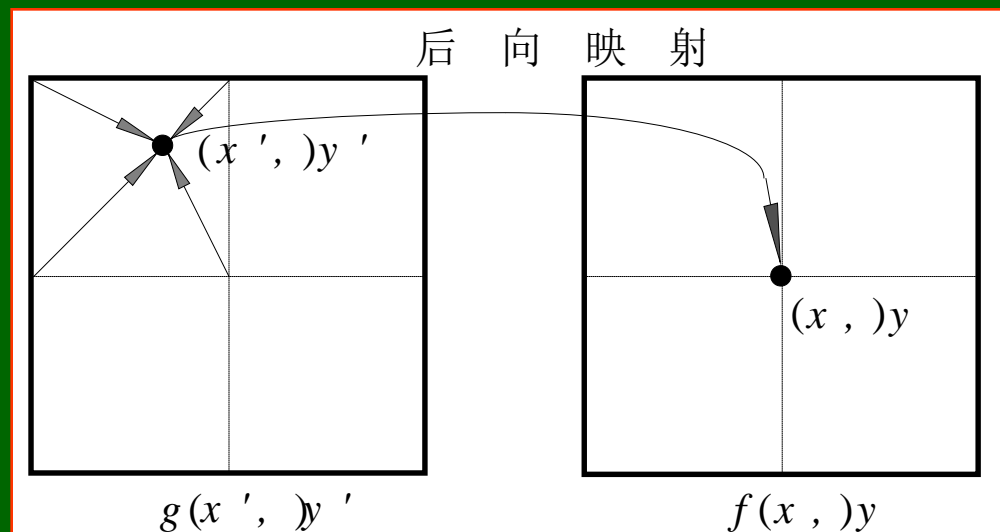
6.2.2 灰度插值

后向映射

实际失真图中四个象素之间的某个位置对应不失真图的某个象素，则先根据插值算法计算出该位置的灰度，再将其映射给不失真图的对应象素

非失真图

→ 失真图



6.2.2 灰度插值

前向映射和后向映射对比

后向映射既可以避免在不失真图象中产生空洞，也不需要重复计算多个失真图象中像素的贡献之和，所以实际中用得更为广泛

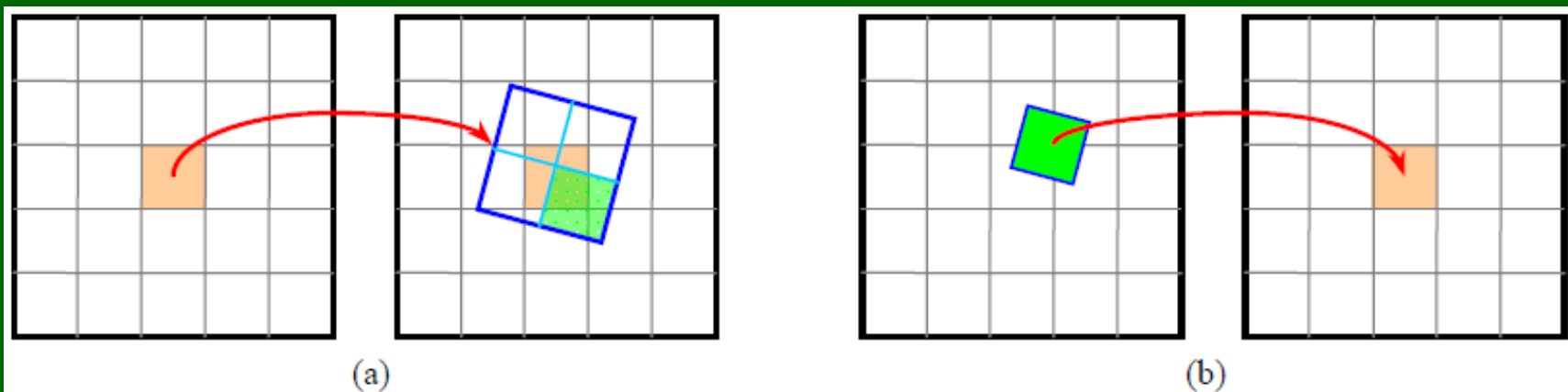


图 6.2.4 前向映射和后向映射对比

6.2.2 灰度插值

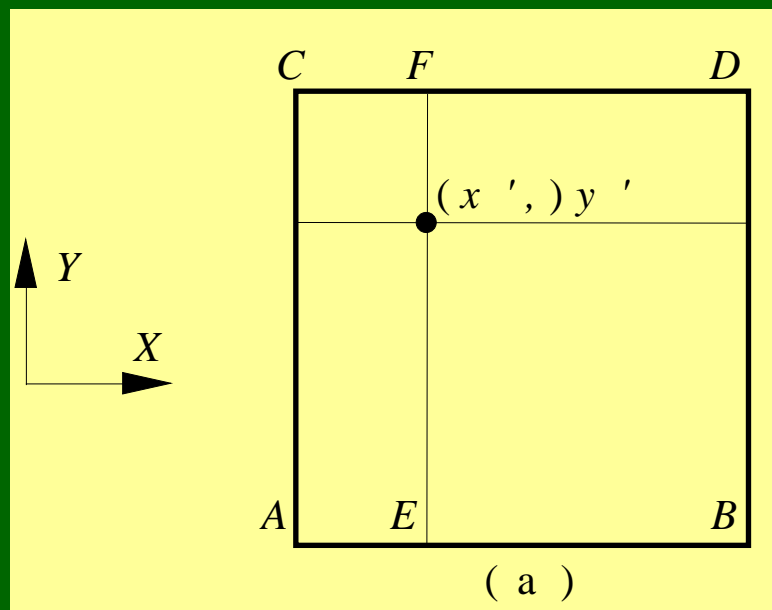
双线性插值

利用 (x', y') 点的四个最近邻像素 A 、 B 、 C 、 D ，灰度值分别为 $g(A)$ 、 $g(B)$ 、 $g(C)$ 、 $g(D)$

$$g(E) = (x' - i)[g(B) - g(A)] + g(A)$$

$$g(F) = (x' - i)[g(D) - g(C)] + g(C)$$

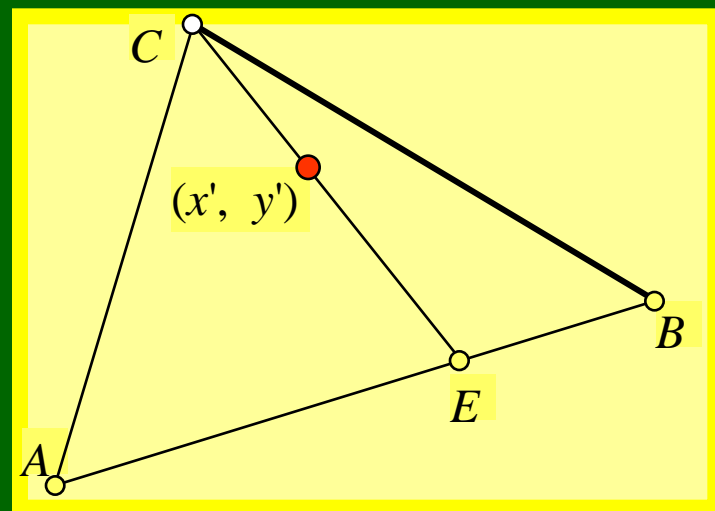
$$g(x', y') = (y' - j)[g(F) - g(E)] + g(E)$$



6.2.2 灰度插值

双线性插值

利用 (x', y') 点的三个不共线近邻像素A、B、C，灰度值分别为 $g(A)$ 、 $g(B)$ 、 $g(C)$ 。先计算 $g(E)$



$$g(E) = \frac{x_E - x_B}{x_A - x_B} g(A) + \frac{x_A - x_E}{x_A - x_B} g(B)$$

$$g(x', y') = \frac{x' - x_C}{x_E - x_C} g(E) + \frac{x_E - x'}{x_E - x_C} g(C)$$

6.2.2 灰度插值

双线性插值

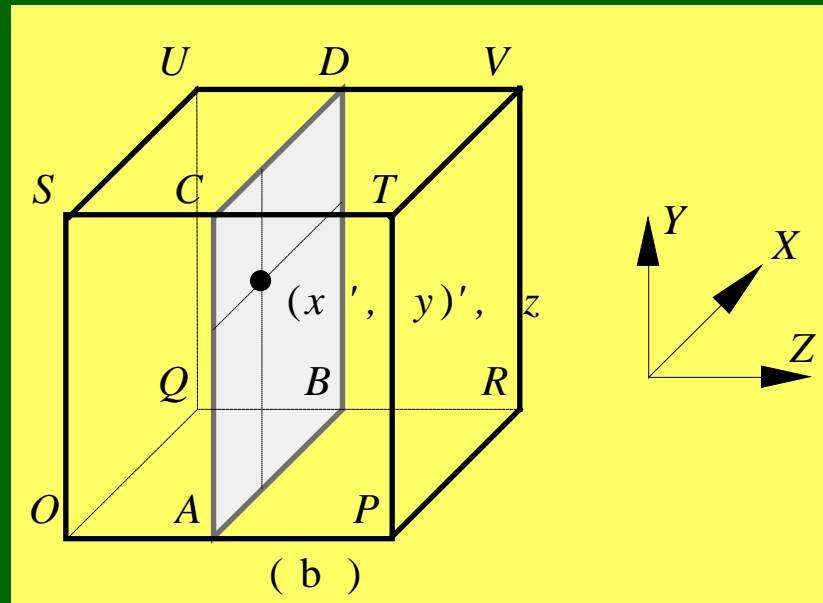
推广到3-D

$$g(A) = (z' - k)[g(P) - g(O)] + g(O)$$

$$g(B) = (z - k)[g(R) - g(Q)] + g(Q)$$

$$g(C) = (z - k)[g(T) - g(S)] + g(S)$$

$$g(D) = (z - k)[g(V) - g(U)] + g(U)$$



问题转化为2-D (继续用前面插值方法进行)

6.2.2 灰度插值

三次线性插值

$$g(x', y') = \sum W_x W_y g(\cdot)$$

$$d_x < 1$$

$$W_x = 1 - 2d_x^2 + d_x^3$$

$$d_x > 1$$

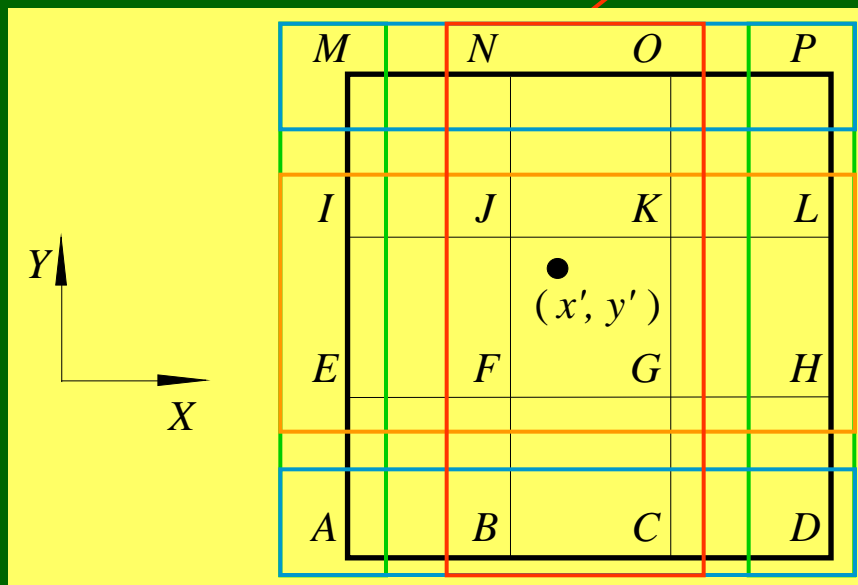
$$W_x = 4 - 8d_x + 5d_x^2 - d_x^3$$

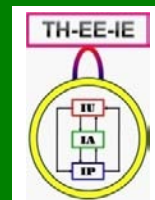
$$d_y < 1$$

$$W_y = 1 - 2d_y^2 + d_y^3$$

$$d_y > 1$$

$$W_y = 4 - 8d_y + 5d_y^2 - d_y^3$$

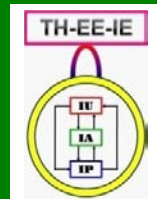




6.3 图象修复

图象中部分区域发生缺损或缺失，像素灰度急剧改变，使图象不完整的情况：

- ① 在采集有遮挡的场景图象或扫描有破损的老图片时产生的部分内容的缺失
- ② 在图象加工中去除特定区域（无关景物）后留下的空白；
- ③ 图象上覆盖文字或受到干扰（照片撕裂或有划痕）导致的变化
- ④ 对图象进行有损压缩时而造成的部分信息丢失
- ⑤ 在（网络上）传输数据时由于网络故障所导致的像素丢失



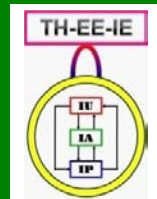
6.3 图象修复

{ 图象的退化反映在象素的位置变化以及灰度变化上 }

6.3.1 图象修补原理

6.3.2 全变分模型

6.3.3 混合模型



6.3.1 图象修补原理

图象修补

基于不完整的图象和对原始图象的先验知识，通过采用相应的方法纠正或校正前述区域缺损问题，以达到恢复图象原貌的目的

✓ 修复（插补）

修补尺度较小的区域，利用局部结构信息

✓ 补全（填充）

修补尺度较大的区域，考虑整图纹理信息

➤ 相互之间在尺度上并没有严格的界限

6.3.1 图象修补原理

图象修补

病态的问题，解是不确定的

- ①领域复杂性：需修补区域随应用而不相同
- ②图象复杂性：性质在不同尺度表现不同
- ③模式复杂性：考虑视觉上有意义的模式



图 6.3.1 模式复杂性示例

6.3.1 图象修补原理

图象修补模型

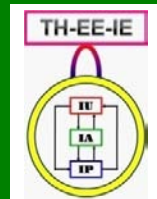
原始图象 $f(x, y)$, 其分布的空间区域用 F 表示
待修补图象 $g(x, y)$, 其分布的空间区域也是 F

$$[g(x, y)]_{F-D} = \{H[f(x, y)] + n(x, y)\}_{F-D}$$

待修补部分为 $d(x, y)$
其空间区域用 D 表示
靶区域 D
源区域 $F-D$



图 6.3.2 图像修补中各区域示意



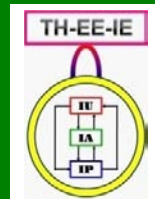
6.3.2 全变分模型

用于去除划痕或尺寸较小靶区域的修复技术
基于偏微分方程或变分模型，两者可以借助
变分原理相互等价推出

通过对靶区域进行逐个像素的扩散来达到修
复图象的目的

沿着等光强线（相等灰度值的线）由源区域
向靶区域延伸扩散，扩散时可借助**全变分模型**
（TV）来恢复图象中的缺失信息

优点是可以很好地保持图象中的线性结构，
缺点是不一定能保持图象细节



6.3.2 全变分模型

扩散的代价函数

$$R[f] = \iint_F |\nabla f(x, y)| dx dy$$

约束（去除噪声）

$$\frac{1}{\|F - D\|} \iint_{F-D} |f - g|^2 dx dy = \sigma^2$$

有约束问题转化成无约束问题

$$E[f] = \iint_F |\nabla f(x, y)| dx dy + \frac{\lambda}{2} \iint_{F-D} |f - g|^2 dx dy$$

6.3.2 全变分模型

引入扩展的拉格朗日因子 λ_D

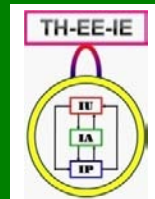
$$\lambda_D(r) = \begin{cases} 0 & r \in D \\ \lambda & r \in (F - D) \end{cases}$$

新的泛函式

$$J[f] = \iint_F |\nabla f(x, y)| dx dy + \frac{\lambda_D}{2} \iint_F |f - g|^2 dx dy$$

能量梯度下降方程

$$\frac{\partial f}{\partial t} = \nabla \cdot \left[\frac{\nabla f}{|\nabla f|} \right] + \lambda_D (f - g)$$



6.3.3 混合模型

代价函数的梯度项改为梯度平方项

$$R[f] = \iint_F |\nabla f(x, y)|^2 dx dy$$

借助扩展拉格朗日因子得到泛函

$$J[f] = \iint_F |\nabla f(x, y)|^2 dx dy + \frac{\lambda_D}{2} \iint_F |f - g|^2 dx dy$$

✓ 取两种模型的加权和的混合模型：调和模型

$$R_h[f] = \iint_F h |\nabla f(x, y)| + \frac{(1-h)}{2} |\nabla f(x, y)|^2 dx dy$$



6.3.3 混合模型

调和模型的泛函 ($h=1$ 时为全变分模型)

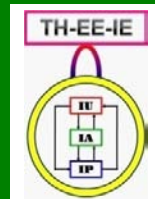
$$J_h[f] = \iint_F h |\nabla f(x, y)| + \frac{(1-h)}{2} |\nabla f(x, y)|^2 dx dy + \frac{\lambda_D}{2} \iint_F |f - g|^2 dx dy$$

✓ 混合模型: p -调和模型

$$R_p[f] = \iint_F |\nabla f(x, y)|^p dx dy$$

p -调和模型的泛函 ($p=2$ 时为调和模型)

$$J_p[f] = \iint_F |\nabla f(x, y)|^p dx dy + \frac{\lambda_D}{2} \iint_F |f - g|^2 dx dy$$

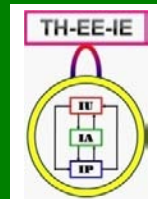


6.4 区域填充

? 修复方法将缺失区域周围的信息向中心扩散，对尺度较大的缺失区域，扩散会造成一定的模糊，且模糊程度随缺失区域的尺度增加而增加

? 修复方法没有考虑缺失区域内部的纹理特性，将缺失区域周围的纹理特性直接移入缺失区域内。由于对尺度比较大的缺失区域，内外纹理特性可能有较大的差别，导致修复结果不很理想

(1) 将图象分解为结构部分和纹理部分，对结构性强的部分用扩散方法进行插补，而对纹理明显的部分则借助纹理合成的技术进行填充

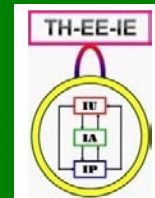


6.4 区域填充

(2) 在图象未退化部分选择一些样本块，用这些样本块来替代拟填充区域边界处的图象块（这些块的未退化部分与所选样本块有接近的特性），并逐步向拟填充区域内部递进填充

➤ 基于第1种思路的方法是一种混合的方法，扩散的方法借助了结构信息，但要完全用纹理合成来填补大面积靶区域仍有一定风险和难度

➤ 基于第2种思路的方法常称为基于样本的图象填充方法。这类方法直接用源区域中的信息（包括纹理信息）来填补靶区域

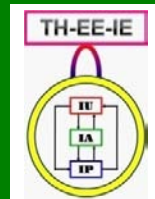


6.4 区域填充

{基于样本的方法及其改进}

6.4.1 基于样本的方法

6.4.2 结合稀疏表达的方法



6.4.1 基于样本的方法

使用保持原状的空间区域去估计和填充待修补部分中缺失的信息

(1) 计算图象块的优先权

填充图象块的工作从外向里进行

先填充具有较强连续边缘（人对边缘信息更敏感）的区域和其中已知信息较多的区域

以边界点 p 为中心的图象块 $P(p)$ 的优先权值

$$P(p) = C(p) \cdot D(p)$$

$C(p)$: 置信度项

$D(p)$: 数据项

算法优先沿着等照度线的方向修复



6.4.1 基于样本的方法

(2) 传播纹理和结构信息

确定出具有最高优先权的图象块（靶区域）
从源区域中选图象块数据来填充它

要使两个图象块中已填充像素的平方差的和最小。这样填充的结果可将纹理和结构都从源区域传播到靶区域中

(3) 更新置信度值

当一个图象块被用新像素值填充后，其置信度值也要用新像素所在图象块的置信度值来更新。这可帮助测量锋线上图象块间相对的置信度

6.4.1 基于样本的方法

T 代表需填补的区域，两个 S (S_1 和 S_2) 表示的区域均为可以用于填补的区域，两个 B 均代表需填补区域的边界（包括 B_1 和 B_2 两段）

剥洋葱皮，根据由外往里、逐圈往内的顺序

原始图象 需填补图象 候选填补块 填补结果

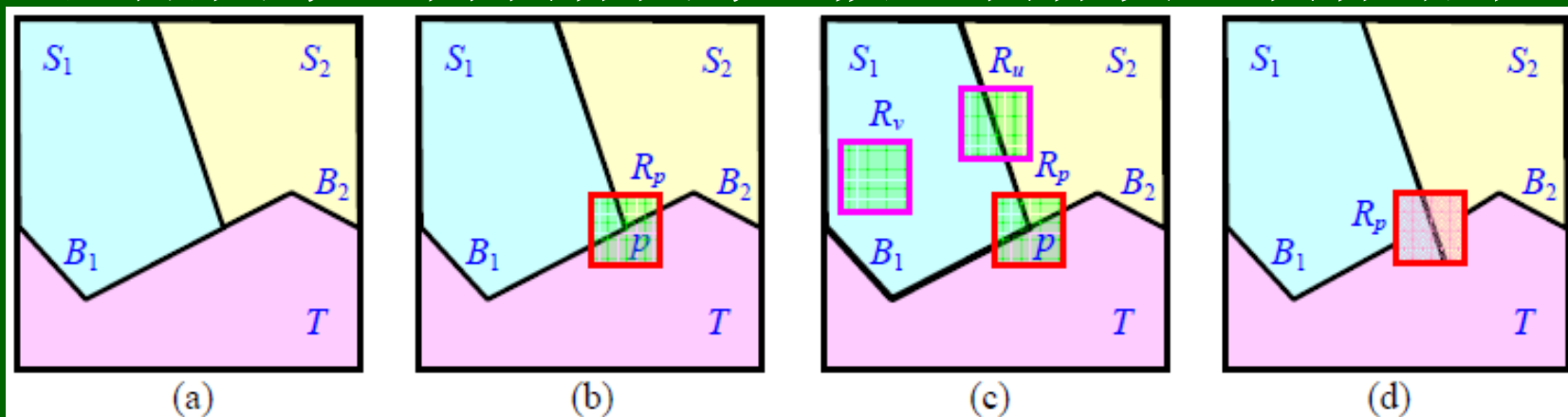


图 6.4.1 基于样本纹理合成的结构传播示意



6.4.2 结合稀疏表达的方法

借助稀疏表达减少基于样本方法的计算量

1. 稀疏表达简介

采用过完备的冗余基代替一般的完备正交基
图象所有像素组成的矢量 $f \in \mathbf{R}^N$ 的超完备表达

$$f = Zx$$

x 表示像素 $f(x, y)$ 的坐标, $x \in \mathbf{R}^K$ 是图象 f 在过完备字典 Z 上线性展开的系数

当 $\|x\|_0 \ll N$, 也就是 x 集合中非零元个数远小于 N 时, 则可以称上述表达是稀疏表达



6.4.2 结合稀疏表达的方法

2. 稀疏表达算法

计算图象的稀疏系数向量，而为此得首先构建一组合适的基，直接在图象中进行采样或者直接使用完整的图象块作为基

计算定义在 S 上的 f_p 在 \mathbf{Z}^S 上展开的系数矢量 \mathbf{x} ，然后根据得到的系数 \mathbf{x} 更新 f_p 中属于 T 部分的像素

$$\hat{\mathbf{x}} = \arg \min_x \left\{ \left\| \mathbf{f}_p^S - \mathbf{Z}^S \cdot \mathbf{x} \right\|_2^2 + \lambda \left\| \mathbf{x} \right\|_1 \right\}$$

$$\hat{f}_p^i = \begin{cases} f_p^i & i \notin I \\ (\mathbf{Z} \cdot \hat{\mathbf{x}})^i & i \in I \end{cases}$$

6.4.2 结合稀疏表达的方法

2. 稀疏表达算法

基于样本的算法在删除不需要景物的同时有可能引入另一些不需要的内容（每次总是选取最相似的一个图象块进行填充，这有可能会把源区域中的一些内容引入到靶区域中）

原始图象 需填补图象 基于样本 稀疏表达



图 6.4.2 6.4.1 小节算法与利用稀疏表达算法的比较

6.4.2 结合稀疏表达的方法

3. 稀疏表达算法改进

(1) 针对填充顺序的改进 (避免单一方向)

图6.4.3

图6.4.4

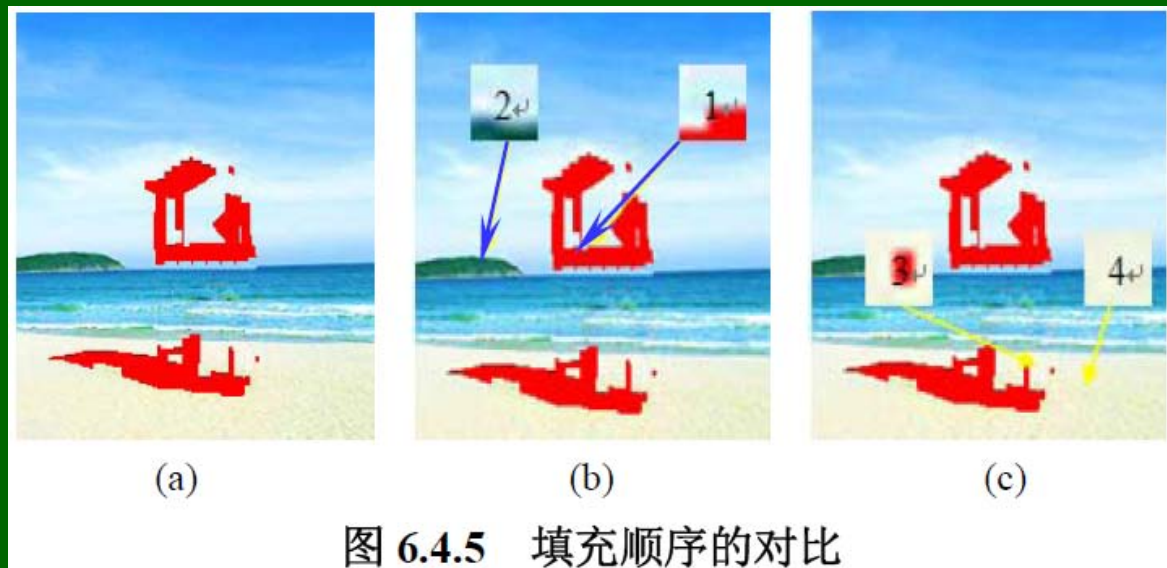


图 6.4.5 填充顺序的对比

(2) 针对填充误差的改进 (限定图象块数)

(3) 针对当前图象块的改进 (考虑方差)



联系信息



- ☞ 通信地址：北京清华大学电子工程系
- ☞ 邮政编码：100084
- ☞ 办公地址：清华大学，罗姆楼，6层305室
- ☞ 办公电话：(010) 62798540
- ☞ 传真号码：(010) 62770317
- ☞ 电子邮件：zhang-yj@tsinghua.edu.cn
- ☞ 个人主页：oa.ee.tsinghua.edu.cn/~zhangyujin/