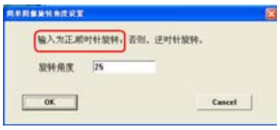
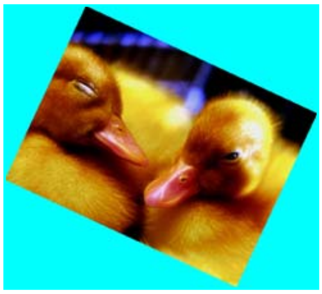
数字图像处理
Digital Image Processing

第03章 图像的几何变换与应用

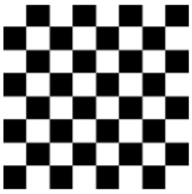
张朝晖
河北师范大学
2020年9月23日

图像的几何变换 --彩色图像旋转示例(线性变换)





图像的几何变换 －棋盘图像涡流变换示例(非线性)



3

1.图像几何变换的理论基础

一般意义上，图像几何变换主要包括：

(1) 几何变换模型

(2) 空间采样与灰度插值

数字图像的几何变换——空间变换：

- 体现了变换前后的两幅图像中点与点之间的空间映射关系
- 是图像几何变形的基础
- 广泛用于遥感图像的几何校正、医学成像、计算机视觉以及电影、电视和媒体广告等的影像特技处理中。

1.1 几何变换模型

描述了变换前后两幅图像中各像素空间坐标的映射关系。

设：变换前输入图像的像素坐标 (x,y)

变换后输出图像的像素坐标 (u,v)

两图像坐标系之间的映射关系用函数表示：

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} U(x,y) \\ V(x,y) \end{bmatrix}$$

前向映射：将输入图像的任意像素坐标(x,y)映射到输出图像中的相应位置(u,v)。是从整数集(输入像素的坐标)到实数集(输出的像素坐标)的映射。

输入图像的采样点是均匀分布的；但映射后输出图像的采样点就不一一定是均匀分布的。

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} X(u,v) \\ Y(u,v) \end{bmatrix}$$

逆向映射：将输出图像中空间均匀分布的像素坐标(整数坐标)映射到输入图像中的相应位置（浮点坐标），此种映射可确保输出图像任何像素在输入图像坐标系都有对应位置。

该方法易实现、多在实际中应用。

以仿射变换为例，掌握一些基本的线性几何变换模型

仿射变换(*affine transform*)：

$$\begin{cases} x = a_1 u + a_2 v + a_3 \\ y = b_1 u + b_2 v + b_3 \end{cases}$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

其中仿射变换矩阵 $\begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ 0 & 0 & 1 \end{bmatrix}$ 必为非奇异矩阵

仿射变换的基本变换

平移(translation)

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & a_3 \\ 0 & 1 & b_3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \qquad \begin{cases} x = u + a_3 \\ y = v + b_3 \end{cases}$$

水平镜像(horizontal mirror)

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & wide-1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \qquad \begin{cases} x = u \\ y = wide - 1 - v \end{cases}$$

垂直镜像(vertical mirror)

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & height-1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \qquad \begin{cases} x = height - 1 - u \\ y = v \end{cases}$$

转置(transpose)

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \qquad \begin{cases} x = v \\ y = u \end{cases}$$

放缩(scaling)

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} C_u & 0 & 0 \\ 0 & C_v & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \qquad \begin{cases} x = u \cdot C_u \\ y = v \cdot C_v \end{cases}$$

旋转(rotation)

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \qquad \begin{cases} x = u \cdot \cos\alpha + v \sin\alpha \\ y = -u \sin\alpha + v \cdot \cos\alpha \end{cases}$$

水平错切(horizontal shear)

保持变换前后的纵坐标不变，某个位置在变换后相对于变换前的水平错切量关于其变换前的纵坐标成比例关系。

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ S_u & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \qquad \begin{cases} x = u \\ y = v + S_u u \end{cases}$$

垂直错切(vertical shear)

保持变换前后的横坐标不变，某个位置在变换后相对于变换前的垂直错切量关于其变换前的横坐标成比例关系。

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & S_v & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \qquad \begin{cases} x = u + S_v v \\ y = v \end{cases}$$

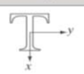
将上述各种运算等号两侧同时转置，有：

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}^T = \left\{ \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \right\}^T$$

$$\begin{bmatrix} x & y & 1 \end{bmatrix} = \begin{bmatrix} u & v & 1 \end{bmatrix} \begin{bmatrix} a_1 & b_1 & 0 \\ a_2 & b_2 & 0 \\ a_3 & b_3 & 1 \end{bmatrix}$$

变换矩阵 $\begin{bmatrix} a_1 & b_1 & 0 \\ a_2 & b_2 & 0 \\ a_3 & b_3 & 1 \end{bmatrix}$ 具体为：

教材51页

Transformation Name	Affine Matrix, T	Coordinate Equations	Example
Identity	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = w$	
Scaling	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = c_x v$ $y = c_y w$	
Rotation	$\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v\cos\theta - w\sin\theta$ $y = v\cos\theta + w\sin\theta$	
Translation	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$	$x = v + t_x$ $y = w + t_y$	
Shear (vertical)	$\begin{bmatrix} 1 & 0 & 0 \\ s_v & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v + s_v w$ $y = w$	
Shear (horizontal)	$\begin{bmatrix} 1 & s_h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = s_h v + w$	

1.2 空间采样与灰度插值

输出图像I_2

每个像素的整数坐标(u,v)

坐标变换

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} X(u,v) \\ Y(u,v) \end{bmatrix}$$

输入图像I_1

浮点坐标(x,y)

灰度/颜色插值

$I_1(x,y)$

像素值赋值

$I_2(u,v)$ <== $I_1(x,y)$

常见的灰度插值方法：

- 近邻法：在输入图像中距离浮点位置最近的整数坐标像素值为输出图像的像素值。
- 双线性内插法(bilinear)：利用与输入图像中浮点位置最为邻近的2*2邻域整数坐标像素值线性加权
- 双三次卷积法(bicubic)：利用与输入图像中浮点位置最为邻近的4*4邻域整数坐标像素值加权

方法1--近邻法(*nearest neighbour*)

已知：某图像f中的浮点坐标$(x, y) = (x_0 + \Delta x, y_0 + \Delta y)$

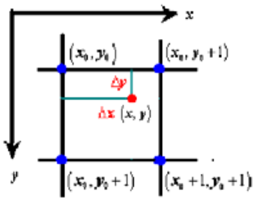
确定：$f(x, y)$

方法：首先计算与(x, y)距离最近的整数坐标(x', y')

$$x' = \begin{cases} x_0 & 若0 \le |\Delta x| < 0.5 \\ x_0 + 1 & 若0.5 \le |\Delta x| < 1 \end{cases}$$

$$y' = \begin{cases} y_0 & 若0 \le |\Delta y| < 0.5 \\ y_0 + 1 & 若0.5 \le |\Delta y| < 1 \end{cases}$$

所以 $\boxed{f(x, y) \Leftarrow f(x', y')}$



该方法最简单。

当图像中所包含的像素之间灰度级存在细微变化的结构时，该方法会在图像中产生人为加工的痕迹。

方法2：双线性内插法(*bilinear interpolation*)

已知：某图像f中的浮点坐标$(x, y) = (x_0 + \Delta x, y_0 + \Delta y)$

确定：$f(x, y)$

方法：要进行三个位置的线性插值运算。

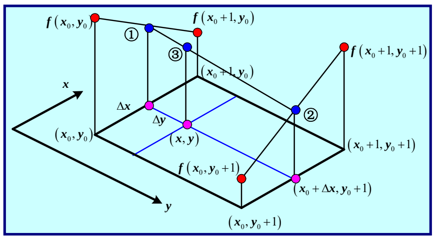
分别计算：

行向、列向两个方向的灰度线性插值

$f(x_0 + \Delta x, y_0)$

$f(x_0 + \Delta x, y_0 + 1)$

$f(x_0 + \Delta x, y_0 + \Delta y)$

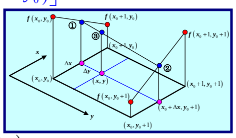


计算$f(x_0 + \Delta x, y_0)$：

$$\frac{f(x_0 + \Delta x, y_0) - f(x_0, y_0)}{f(x_0 + 1, y_0) - f(x_0, y_0)} = \frac{(x_0 + \Delta x) - x_0}{(x_0 + 1) - x_0}$$

$$\Rightarrow f(x_0 + \Delta x, y_0) = \left[f(x_0 + 1, y_0) - f(x_0, y_0)\right] \cdot \Delta x + f(x_0, y_0)$$

$$\Rightarrow f(x_0 + \Delta x, y_0) = \left[(1 - \Delta x) \quad \Delta x\right] \begin{bmatrix} f(x_0, y_0) \\ f(x_0 + 1, y_0) \end{bmatrix}$$



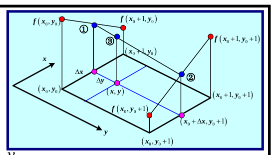
计算$f(x_0 + \Delta x, y_0 + 1)$：

$$\frac{f(x_0 + \Delta x, y_0 + 1) - f(x_0, y_0 + 1)}{f(x_0 + 1, y_0 + 1) - f(x_0, y_0 + 1)} = \frac{(x_0 + \Delta x) - x_0}{(x_0 + 1) - x_0}$$

$$\Rightarrow f(x_0 + \Delta x, y_0 + 1) = \left[(1 - \Delta x) \quad \Delta x\right] \begin{bmatrix} f(x_0, y_0 + 1) \\ f(x_0 + 1, y_0 + 1) \end{bmatrix}$$

15

计算 $f(x, y) = f(x_0 + \Delta x, y_0 + \Delta y)$：



$$\frac{f(x_0 + \Delta x, y_0 + \Delta y) - f(x_0 + \Delta x, y_0)}{f(x_0 + \Delta x, y_0 + 1) - f(x_0 + \Delta x, y_0)} = \frac{(y_0 + \Delta y) - y_0}{(y_0 + 1) - y_0}$$

$$\Rightarrow f(x_0 + \Delta x, y_0 + \Delta y) = \left[f(x_0 + \Delta x, y_0) \quad f(x_0 + \Delta x, y_0 + 1)\right] \begin{bmatrix} (1 - \Delta y) \\ \Delta y \end{bmatrix}$$

又由于

$$\left[f(x_0 + \Delta x, y_0) \quad f(x_0 + \Delta x, y_0 + 1)\right] = \left[(1 - \Delta x) \quad \Delta x\right] \begin{bmatrix} f(x_0, y_0) & f(x_0, y_0 + 1) \\ f(x_0 + 1, y_0) & f(x_0 + 1, y_0 + 1) \end{bmatrix}$$

所以

$$f(x, y) = f(x_0 + \Delta x, y_0 + \Delta y) = \left[(1 - \Delta x) \quad \Delta x\right] \begin{bmatrix} f(x_0, y_0) & f(x_0, y_0 + 1) \\ f(x_0 + 1, y_0) & f(x_0 + 1, y_0 + 1) \end{bmatrix} \begin{bmatrix} (1 - \Delta y) \\ \Delta y \end{bmatrix}$$

16

综上，利用**双线性插值**，计算得：

$$\boxed{f(x_0 + \Delta x, y_0 + \Delta y) = A \cdot B \cdot C}$$

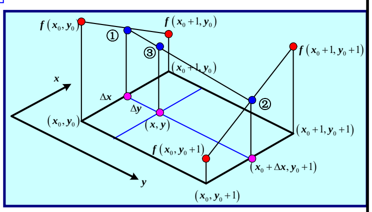
其中：$A = \left[(1 - \Delta x) \quad \Delta x\right]$

$$B = \begin{bmatrix} f(x_0, y_0) & f(x_0, y_0 + 1) \\ f(x_0 + 1, y_0) & f(x_0 + 1, y_0 + 1) \end{bmatrix}$$

$$C = \left[(1 - \Delta y) \quad \Delta y\right]^T$$

效果好于近邻法。

但该方法具有平滑作用，可能造成细节的退化。



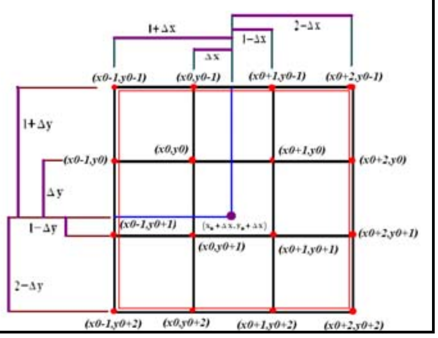
方法3：双三次插值法(*bicubic interpolation*)

已知：某图像f中的浮点坐标$(x, y) = (x_0 + \Delta x, y_0 + \Delta y)$

确定：$f(x, y)$

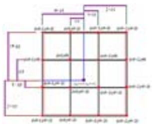
三次卷积运算中，需要用到该浮点坐标周围16个整数坐标点的像素值。计算量明显增加。

但该方法的插值效果最好。



步骤：

[1]确定$\Delta x, \Delta y$以及x_0, y_0



[2]根据**权值**函数$\omega(d)=\begin{cases}1-2|d|^2+|d|^3 & ,|d|<1 \\ 4-8|d|+5|d|^2-|d|^3 & ,1\le|d|<2 \\ 0 & ,|d|\ge 2\end{cases}$

估计$W_Y=\left[\omega(1+\Delta y)\quad \omega(\Delta y)\quad \omega(1-\Delta y)\quad \omega(2-\Delta y)\right]$

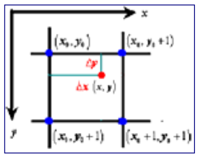
以及$W_X=\left[\omega(1+\Delta x)\quad \omega(\Delta x)\quad \omega(1-\Delta x)\quad \omega(2-\Delta x)\right]^t$

[3]求$f(x,y)$: $f(x,y)=f(x_0+\Delta x, y_0+\Delta y)=W_Y\cdot F\cdot W_X$

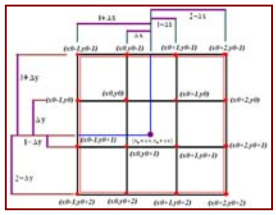
其中

$$F=\begin{bmatrix} f(x_0-1,y_0-1) & f(x_0,y_0-1) & f(x_0+1,y_0-1) & f(x_0+2,y_0-1) \\ f(x_0-1,y_0) & f(x_0,y_0) & f(x_0+1,y_0) & f(x_0+2,y_0) \\ f(x_0-1,y_0+1) & f(x_0,y_0+1) & f(x_0+1,y_0+1) & f(x_0+2,y_0+1) \\ f(x_0-1,y_0+2) & f(x_0,y_0+2) & f(x_0+1,y_0+2) & f(x_0+2,y_0+2) \end{bmatrix}$$

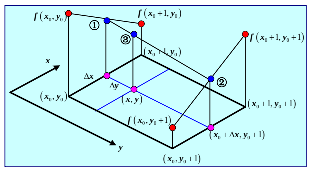
三种典型的灰度插值法图示。



近邻法



三次卷积法



双线性内插法

20

例：输入图像的**顺时针21°** **旋转变换**(坐标变换+灰度插值)

插值方式：近邻法、双线性、双三次样条

1.3 数字图像的几何变换基本实现步骤

已知： 变换前的**输入图像**$I_1(x,y)$

逆向几何变换模型$\begin{cases}x=f_1(u,v)\\ y=f_2(u,v)\end{cases}$

前向几何变换模型$\begin{cases}u=g_1(x,y)\\ v=g_2(x,y)\end{cases}$

以及某种选定的**灰度插值算法**、**无效位置标记值LABEL**

求取： 包含变换后**图像**$I_2(u,v)$完整内容的**输出图像**$I_3(X,Y)$

> ➢图像几何变换
>
> **输入**一幅待处理的图像，以及几何变换模型，**生成**变换后的输出图像。

思考：

(1)前向变换的意义

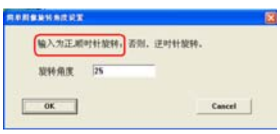
(2)逆向变换的意义

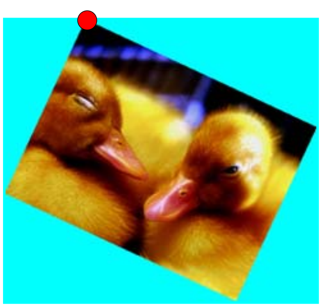
(3)$I_1(x,y), I_2(u,v), I_3(X,Y)$三个图像坐标系的关系

(4)**输出图像**I_3的大小？图像内容如何填充(有效位置、无效位置)

22

彩色图像的几何变换(旋转变换)





步骤：

STEP1.利用**前向几何变换模型**$\begin{cases}u=g_1(x,y)\\ v=g_2(x,y)\end{cases}$

估算**输入图像**$I_1(x,y)$的四个顶点位置$\{(x_i,y_i), i=1,2,3,4\}$

在**图像**$I_2(u,v)$坐标系的对应位置$\{(u_i,v_i), i=1,2,3,4\}$

STEP2.估计能有效包含**图像**I_2内容的**输出图像**$I_3(X,Y)$的行数H、列数W

$u_{\max}=\max\{u_i, i=1,2,3,4\}$　$u_{\min}=\min\{u_i, i=1,2,3,4\}$

$v_{\max}=\max\{v_i, i=1,2,3,4\}$　$v_{\min}=\min\{v_i, i=1,2,3,4\}$

则 $\begin{cases}H=\text{ceil}(u_{\max}-u_{\min}+1)\\ W=\text{ceil}(v_{\max}-v_{\min}+1)\end{cases}$

STEP3.估计**输出图像**$I_3(X,Y)$坐标系原点$(0,0)$在**图像**$I_2(u,v)$坐标系的位置(u_0,v_0): $(u_0,v_0)=(u_{\min}, v_{\min})$

24

STEP4.逐行、逐列遍历**输出图像**I_3的每个像素(X,Y)：

(1)确定其在**图像**I_2所在坐标系的位置$(u,v)=(X+u_0, Y+v_0)$

(2)利用逆向模型计算(u,v)在**输入图像**I_1的对应位置

$(x,y)=(f_1(u,v),\ f_2(u,v))$

(3)若(x,y)为**输入图像**I_1**有效位置**，则基于选定的灰度插值方法，得到取值$I_1(x,y)$，将其赋给$I_3(X,Y)$；

否则，以**无效标记值**为$I_3(X,Y)$赋值。

STEP5.显示并保存**输出图像**I_3

25

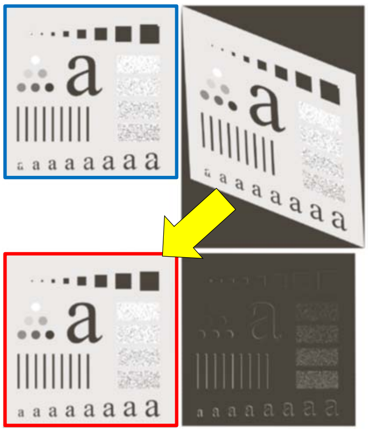
2.图像几何变换的应用--图像配准与拼接

➢图像配准

空间坐标变换关系未知，但变换前后的图像已知；基于变换前、后两幅图像，估计空间坐标几何变换模型；再基于估计的变换模型，以其中一幅图像坐标系为基准，对另外一幅图像进行几何变换。

➢ 图像配准要用到同名点(tie-points)

26



图像配准示例：

左上，参考图像；

右上，输入图像；

左下，配准后的输入图像；

右下，配准误差图像

27

图像配准(Image Registration)的基本步骤

已知：$\begin{cases} \text{参考图像 } M_1\text{行}\times N_1\text{列 } I_1(x,y) \\ \text{输入图像 } M_2\text{行}\times N_2\text{列 } I_2(u,v) \\ \text{同名点对}\{(x_i,y_i),(u_i,v_i)\mid i=1,\dots,n\} \end{cases}$ $\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$

求取：配准的图像对$\begin{cases} \boldsymbol{RegI}_1(x,y) \\ \boldsymbol{RegI}_2(x,y) \end{cases}$ 均为M_1行$\times N_1$列

$$\begin{cases} u = a_1x+a_2y+a_3 \\ v = b_1x+b_2y+b_3 \end{cases} \quad \text{或} \quad [u_i \quad v_i] = [x_i \quad y_i \quad 1]\begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \end{bmatrix}$$

$$\Rightarrow \underbrace{\begin{bmatrix} u_1 & v_1 \\ u_2 & v_2 \\ \vdots & \vdots \\ u_n & v_n \end{bmatrix}}_{U} = \underbrace{\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{bmatrix}}_{X} \underbrace{\begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \end{bmatrix}}_{A} \qquad U = XA$$

$$\Rightarrow X^TU = X^TXA \qquad \Rightarrow \boxed{A=(X^TX)^{-1}X^TU}$$

仿射变换模型的参数估计(伪逆法/最小二乘法)

28

图像配准步骤：

STEP1. 基于同名点对，估计前向仿射变换模型参数$A = \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \end{bmatrix}$

STEP2. 在参考图像$I_1(x,y)$**坐标系，生成配准结果**

$for\ \ x = 0 : M_1 - 1$

$\quad for\ \ y = 0 : N_1 - 1$

$\quad\quad$(1)前向变换：由(x,y)确定(u,v) $\quad [u \quad v] = [x \quad y \quad 1]A$

$\quad\quad$(2)若$u \in [0, M_2-1]$并且$v \in [0, N_2-1]$ 则$\begin{cases} \boldsymbol{RegI}_1(x,y) \Leftarrow \boldsymbol{I}_1(x,y) \\ \boldsymbol{RegI}_2(x,y) \Leftarrow \boldsymbol{I}_2(u,v) \end{cases}$

$\quad\quad$其中$\boldsymbol{RegI}_2(x,y)$**由输入图像**$\boldsymbol{I}_2(u,v)$**灰度插值得到。**

$\quad\quad$否则$\begin{cases} \boldsymbol{RegI}_1(x,y) \Leftarrow 255(\text{无效标记值}) \\ \boldsymbol{RegI}_2(x,y) \Leftarrow 255(\text{无效标记值}) \end{cases}$

$\quad endfor$

$endfor$

STEP3. 输出配准图像对$(\boldsymbol{RegI}_1$及$\boldsymbol{RegI}_2$的显示、保存)

29

图像拼接(Image Mosaicing)的基本步骤

已知：$\begin{cases} \text{参考图像 } M_1\text{行}\times N_1\text{列 } I_1(x,y) \\ \text{输入图像 } M_2\text{行}\times N_2\text{列 } I_2(u,v) \\ \text{同名点对}\{(x_i,y_i),(u_i,v_i)\mid i=1,\dots,n\} \end{cases}$ $\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$

求取：拼接图像 $\boldsymbol{I}_3(X,Y)$ $\quad M_3$行$\times N_3$列

前向映射 $\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$ 或 $[u \quad v \quad 1] = [x \quad y \quad 1]\begin{bmatrix} a_1 & b_1 & 0 \\ a_2 & b_2 & 0 \\ a_3 & b_3 & 1 \end{bmatrix}$

逆向映射 $\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} c_1 & c_2 & c_3 \\ d_1 & d_2 & d_3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$ 或 $[x \quad y \quad 1] = [u \quad v \quad 1]\begin{bmatrix} c_1 & d_1 & 0 \\ c_2 & d_2 & 0 \\ c_3 & d_3 & 1 \end{bmatrix}$

并且 $\begin{bmatrix} c_1 & d_1 & 0 \\ c_2 & d_2 & 0 \\ c_3 & d_3 & 1 \end{bmatrix} \begin{bmatrix} a_1 & b_1 & 0 \\ a_2 & b_2 & 0 \\ a_3 & b_3 & 1 \end{bmatrix} = I$

图像拼接步骤：

STEP1. 基于同名点对，估计前向仿射变换模型参数A及逆向变换模型参数B. $A=\begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \end{bmatrix}$ $B=\begin{bmatrix} c_1 & d_1 \\ c_2 & d_2 \\ c_3 & d_3 \end{bmatrix}$

STEP2. 确定拼接图像的行数M_3及列数N_3.

(1)基于**逆向变换模型**，确定输入图像$I_2(u,v)$的四个顶点在参考图像坐标系的浮点坐标；

(2)根据上述浮点坐标以及参考图像$I_1(x,y)$的四个顶点，确定8个坐标位置中的行列坐标最小及最大值：

$x_{\min}, x_{\max}, y_{\min}, y_{\max}$

(3)估计**拼接图像最小外接矩形在参考图像坐标系的位置**，其中：

左上角$(x_{\min}, y_{\min})$及右下角$(x_{\max}, y_{\max})$

(4)估计拼接图像**的行数M_3及列数N_3**：

$$M_3 = ceil(x_{\max} - x_{\min} + 1) \quad N_3 = ceil(y_{\max} - y_{\min} + 1)$$

图像拼接步骤：

STEP3. 在拼接图像坐标系生成拼接图像.

for $X = 0 : M_3 - 1$ 　拼接图像的像素在拼接图像坐标系的坐标(X,Y)

　for $Y = 0 : N_3 - 1$

　　(1)估算(X,Y)在参考图像I_1的对应位置(x,y): 　拼接图像的像素在参考图像坐标系的坐标(x,y)

　　　$(x,y) = (X + x_{\min}, Y + y_{\min})$

　　(2A)如果(x,y)落入参考图像I_1的有效覆盖区域，即

　　　$x \in [0,\ M_1 - 1]$并且$y \in [0,\ N_1 - 1]$

　　　则$I_3(X,Y) \Leftarrow I_1(x,y)$

　　(2A)否则将(x,y)前向变换至输入图像I_2所在坐标系的对应位置(u,v)。　拼接图像的像素在输入图像坐标系的坐标(u,v)

　　　A.如果(u,v)落入输入图像I_2有效覆盖区域，即

　　　$u \in [0,\ M_2 - 1]$并且$v \in [0,\ N_2 - 1]$

　　　则$I_3(X,Y) \Leftarrow I_2(u,v)$

　　　B.否则$I_3(X,Y) \Leftarrow 255$(无效位置标记值)

　end for

end for

STEP4.显示、保存最终的拼接图像

32

本章小结

图像的几何变换：

(1)什么是数字图像的几何变换？

常见线性几何变换模型有哪些？

如何用齐次变换矩阵表示这些线性变换？

(2)理解三种常见的灰度插值方式及原理？

(3)掌握图像几何变换步骤。

给定几何变换模型，如何对图像进行几何变换？

(4)掌握图像的几何配准与几何拼接原理。

如何实现图像的几何配准与几何拼接？

(5)能编程实现图像的几何变换、几何裁剪；两幅图像的几何配准、拼接

33