《数字图像处理》期末速通教程

3. 图像基本运算

3.1 图像的几何变换

可参考: https://learnopengl-cn.github.io/01%20Getting%20started/07%20Transformations/

3.1.1 齐次坐标

(看一眼) p42 齐次坐标.

[**例 3.1.1.1**] 用齐次坐标表示图像矩阵的好处.

[**答**]

(1) 统一表示平移、旋转和缩放:

在二维平面中, 平移操作不能用普通的矩阵乘法表示. 而通过引入齐次坐标（增加一个维度), 可以使用一个统一的矩阵表示包括平移、旋转、缩放在内的所有仿射变换. 齐次坐标的第三个分量通常设置为1, 这使得平移变换可以通过矩阵乘法实现.

(2) 方便组合变换:

通过矩阵乘法可以方便地组合多个变换. 比如, 一个图像的缩放、旋转和平移可以用一个矩阵表示, 而不需要分别计算. 这样可以简化变换的计算和应用.

(3) 简化计算:

在齐次坐标下, 通过矩阵乘法可以统一处理不同的几何变换, 避免了对不同类型变换的单独处理, 从而简化了计算流程和代码实现.

(4) 处理透视变换:

齐次坐标可以自然地处理透视变换, 这是普通二维坐标系无法实现的. 透视变换对于图像的三维效果处理、摄像机视角变换等场景非常重要.

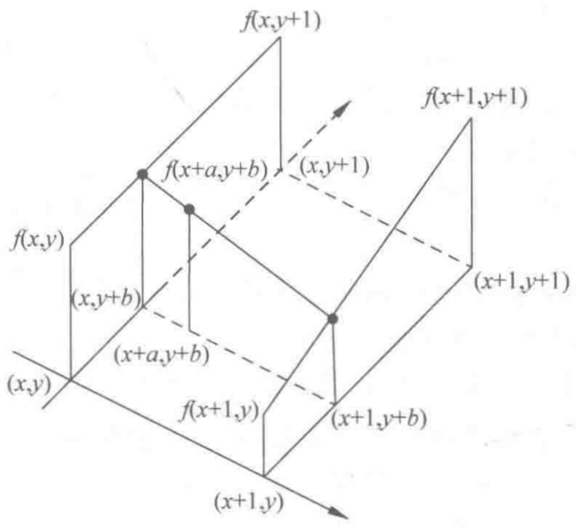
(5) 增强数值稳定性:

使用齐次坐标进行变换计算, 特别是在涉及多次变换组合的情况下, 可以减少由于直接数值计算带来的误差累积问题, 提高变换计算的数值稳定性.

3.1.2 图像的插值

(看一眼) p43 ~ 44 最近邻插值、双三次插值.

[**定理 3.1.2.1**] [**双线性插值**] 如下图, 插值点 $(x+a,y+b)\ \ (x,y\in\mathbb{N};0\leq a,b\leq 1)$ 处的像素值 $f(x+a,y+b)$ 可由原图像中的四个点 (x,y) 、$(x+1,y)$ 、$(x,y+1)$ 和 $(x+1,y+1)$ 处的像素值决定, 即 $\begin{cases}f(x,y+b)=f(x,y)+b[f(x,y+1)-f(x,y)]\\ f(x+1,y+b)=f(x+1,y)+b[f(x+1,y+1)-f(x+1,y)]\\ f(x+a,y+b)=f(x,y+b)+a[f(x+1,y+b)-f(x,y+b)]\end{cases}$.



[**证**]

(1) 在过点 (x,y) 和点 $(x,y+1)$ 且垂直于 xy 平面的平面上, x 坐标不变.

设过点 $(y,f(x,y))$ 和点 $(y+1,f(x,y+1))$ 的直线 $l_1=k_1y+m_1$,

则 $k_1=\dfrac{f(x,y+1)-f(x,y)}{(y+1)-y}=f(x,y+1)-f(x,y)$, $m_1=f(x,y)-k_1y$,

进而 $f(x,y+b)=k_1(y+b)+m_1=k_1(y+b)+f(x,y)-k_1y$

$=f(x,y)+b[f(x,y+1)-f(x,y)]$.

(2) 在过点 $(x+1,y)$ 和点 $(x+1,y+1)$ 且垂直于 xy 平面的平面上, x 坐标不变.

类似地, 有 $f(x+1,y+b)=f(x+1,y)+b[f(x+1,y+1)-f(x+1,y)]$.

(3) 在过点 $(x,y+b)$ 和点 $(x+1,y+b)$ 且垂直于 xy 平面的平面上, y 坐标不变.

类似地, 有 $f(x+a,y+b)=f(x,y+b)+a[f(x+1,y+b)-f(x,y+b)]$.

[**推论**] 设插值点 (x,y) 在原图中原图像中的四个邻近点分别为点 (x_1,y_1) 、点 (x_1,y_2) 、点 (x_2,y_1) 和点 (x_2,y_2) , 其中 $x_2=x_1+1$, $y_2=y_1+1$, 则

$$\begin{aligned}f(x,y)=&f(x_1,y_1)\cdot(x_2-x)(y_2-y)+f(x_2,y_1)\cdot(x-x_1)(y_2-y)\\&+f(x_1,y_2)\cdot(x_2-x)(y-y_1)+f(x_2,y_2)\cdot(x-x_1)(y-y_1)\end{aligned}\ .$$

[**注**] 最近邻插值会产生锯齿. 双线性插值可抗锯齿, 新图像边缘较平滑.

3.1.3 图像的位置变换

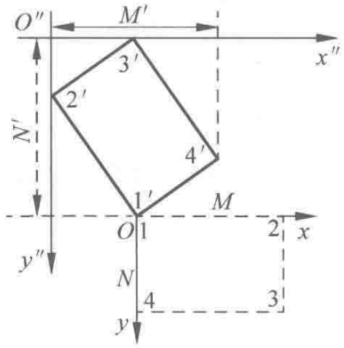
(看一眼) p44 ~ 45 图像的平移、镜像.

[**定理 3.1.3.1**] 将点 (x,y) 绕原点旋转 θ 角(逆时针为正)得到点 (x',y') .

(1) 旋转正变换: $\begin{cases} x'=x\cdot\cos\theta+y\cdot\sin\theta \\ y'=-x\cdot\sin\theta+y\cdot\cos\theta \end{cases}$.

(2) 旋转逆变换: $\begin{cases} x=x'\cdot\cos\theta-y'\cdot\sin\theta \\ y=x'\cdot\sin\theta+y'\cos\theta \end{cases}$.

[**定义 3.1.3.1**] [**图像的旋转**] 如下图, 将图像绕图像原点(图像的左上角)逆时针旋转 θ 角得到新图像.



设原图像坐标系为 xOy , 新图像坐标系为 $x''O''y''$.

设原图像的四个顶点分别为 (x_1,y_1) 、(x_2,y_2) 、(x_3,y_3) 、(x_4,y_4) , 新图像的四个顶点分别为 (x_1',y_1') 、(x_2',y_2') 、(x_3',y_3') 、(x_4',y_4') .

设原图像的大小为 $M\times N$, 则其四个顶点在 xOy 坐标系下的坐标分别为 $(x_1,y_1)=(0,0)$, $(x_2,y_2)=(M-1,0)$, $(x_3,y_3)=(M-1,N-1)$, $(x_4,y_4)=(0,N-1)$.

新图像的四个顶点在 xOy 坐标系下的坐标分别为:

$$\begin{cases} (x_1',y_1')=(0,0) \\ (x_2',y_2')=((M-1)\cdot\cos\theta,-(M-1)\cdot\sin\theta) \\ (x_3',y_3')=((M-1)\cdot\cos\theta+(N-1)\cdot\sin\theta,-(M-1)\cdot\sin\theta+(N-1)\cdot\cos\theta) \\ (x_4',y_4')=((N-1)\cdot\sin\theta,(N-1)\cdot\cos\theta) \end{cases}.$$

(1) 确定新图像的大小.

设 $\begin{cases} maxx'=\max\{x_1',x_2',x_3',x_4'\} \\ minx'=\min\{x_1',x_2',x_3',x_4'\} \end{cases}$, $\begin{cases} maxy'=\max\{y_1',y_2',y_3',y_4'\} \\ miny'=\min\{y_1',y_2',y_3',y_4'\} \end{cases}$,

则新图像的宽度 M' 和高度 N' 分别为 $\begin{cases} M'=\mathrm{round}(maxx'-minx'+1) \\ N'=\mathrm{round}(maxy'-miny'+1) \end{cases}$, 其中 $\mathrm{round}(x)$ 表示对 x 四舍五入.

(2) 坐标变换.

将新图像中的像素点 (x'',y'') $(x''\in[0,M'-1],y''\in[0,N'-1])$ 作平移变换 $\begin{cases} x'=x''+minx' \\ y'=y''+miny' \end{cases}$, 变换到 xOy 坐标系.

(3) 旋转逆变换.

对每个像素点 (x',y') 作旋转逆变换 $\begin{cases} x=x'\cdot\cos\theta-y'\cdot\sin\theta \\ y=x'\cdot\sin\theta+y'\cos\theta \end{cases}$, 变换到原图像中对应的像素点.

(4) 为新图像赋值.

按对应关系直接为新图像的像素点赋值, 或用插值方法赋值.

[注] 绕任意点 P 旋转时, 先作平移变换, 将 P 平移至原点, 绕原点旋转后, 再作平移逆变换.

[例 3.1.3.1] 将图像 $f(x,y)=\begin{bmatrix} 59 & 60 & 58 \\ 61 & 59 & 57 \\ 62 & 56 & 55 \end{bmatrix}$ 绕图像原点逆时针旋转 30° .

[1] 采用最邻近插值.

[2] 采用双线性插值.

[解]

[1] (1) 计算新图像的大小.

原图像的四个顶点旋转后的坐标分别为:

$$\begin{cases} (x_1',y_1')=(0,0) \\ (x_2',y_2')=(2\cdot\cos 30^\circ,-2\cdot\sin 30^\circ)\approx(1.732,-1) \\ (x_3',y_3')=(2\cdot\cos 30^\circ+2\cdot\sin 30^\circ,-2\cdot\sin 30^\circ+2\cdot\cos 30^\circ)\approx(2.732,0.732) \\ (x_4',y_4')=(2\cdot\sin 30^\circ,2\cdot\cos 30^\circ)\approx(1,1.732) \end{cases}.$$

$maxx'=2.732, minx'=0, maxy'=1.732, miny'=-1$.

新图像的大小 $\begin{cases} M'=\mathrm{round}(maxx'-minx'+1)=\mathrm{round}(3.732)=4 \\ N'=\mathrm{round}(maxy'-miny'+1)=\mathrm{round}(3.732)=4 \end{cases}$.

(2) 坐标变换.

将新图像中的像素点 (x'',y'') 作平移变换, 变换到原像素坐标系 (x',y') , 再作旋转逆变换, 变换到原图像中的对应像素点.

(x'',y'')	(x',y')	(x,y)	(x'',y'')	(x',y')	(x,y)
$(0,0)$	$(0,-1)$	$(0.5,-0.866)$	$(2,0)$	$(2,-1)$	$(2.232,0.134)$
$(0,1)$	$(0,0)$	$(0,0)$	$(2,1)$	$(2,0)$	$(1.732,1)$
$(0,2)$	$(0,1)$	$(-0.5,0.866)$	$(2,2)$	$(2,1)$	$(1.232,1.866)$
$(0,3)$	$(0,2)$	$(-1,1.732)$	$(2,3)$	$(2,2)$	$(0.732,2.732)$
$(1,0)$	$(1,-1)$	$(1.366,-0.366)$	$(3,0)$	$(3,-1)$	$(3.098,0.634)$
$(1,1)$	$(1,0)$	$(0.866,0.5)$	$(3,1)$	$(3,0)$	$(2.598,1.5)$
$(1,2)$	$(1,1)$	$(0.366,1.366)$	$(3,2)$	$(3,1)$	$(2.098,2.366)$
$(1,3)$	$(1,2)$	$(-0.134,2.232)$	$(3,3)$	$(3,2)$	$(1.598,3.232)$

(3) 求每个点 (x,y) 对应的最邻近点, 并赋值.

① 若最邻近点超出原图像的范围, 或新图像中的点在原图像中无对应点, 则赋背景色 255 .

② 若点未超出图像范围, 但坐标非整数, 则采用最近邻插值(四舍五入).

(x,y)	最近邻点	像素值	(x,y)	最近邻点	像素值
$(0.5,-0.866)$	$(1,-1)$	越界, 255	$(2.232,0.134)$	$(2,0)$	58
$(0,0)$	$(0,0)$	59	$(1.732,1)$	$(2,1)$	57
$(-0.5,0.866)$	$(-1,1)$	越界, 255	$(1.232,1.866)$	$(1,2)$	56
$(-1,1.732)$	$(-1,2)$	越界, 255	$(0.732,2.732)$	$(1,3)$	越界, 255
$(1.366,-0.366)$	$(1,0)$	60	$(3.098,0.634)$	$(3,1)$	越界, 255
$(0.866,0.5)$	$(1,1)$	59	$(2.598,1.5)$	$(3,2)$	越界, 255
$(0.366,1.366)$	$(0,1)$	61	$(2.098,2.366)$	$(2,2)$	55
$(-0.134,2.232)$	$(0,2)$	62	$(1.598,3.232)$	$(2,3)$	越界, 255

将上表竖着抄成 4×4 的矩阵, 即得新图像 $g(x,y)=\begin{bmatrix}255 & 60 & 58 & 255\\ 59 & 59 & 57 & 255\\ 255 & 61 & 56 & 55\\ 255 & 62 & 255 & 255\end{bmatrix}$.

[2] 以求新图像的点 $(1,2)$ 处的像素值为例, 其对应的原图点 $(0.366,1.366)$ 位于点 $(0,1)$ 、点 $(1,1)$ 、点 $(0,2)$ 、点 $(1,2)$ 间.

$x=0,y=1,a=0.366,b=0.366$.

$f(0,1.366)=f(0,1)+0.366\cdot[f(0,2)-f(0,1)]=61+0.366\times(62-61)\approx 61.366$.

$f(1,1.366)=f(1,1)+0.366\cdot[f(1,2)-f(1,1)]=59+0.366\times(56-59)=57.902$.

$f(0.366,1.366)=f(0,1.366)+0.366\cdot[f(1,1.366)-f(0,1.366)]\approx 60$.

3.1.4 图像的形状变换

(了解) p53 ~ 54 图像的错切.

[**例 3.1.4.1**] 将图像 $f(x,y)=\begin{bmatrix}1&2&3&4&5&6\\7&8&9&10&11&12\\13&14&15&16&17&18\\19&20&21&22&23&24\\25&26&27&28&29&30\\31&32&33&34&35&36\end{bmatrix}$ 按比例 $k_x=0.75$, $k_y=0.6$ 缩小.

[1] 采用最邻近插值.

[2] 采用双线性插值.

[解]

[1] (1) 计算新图像的大小.

新图像的大小 $\begin{cases}M'=k_x\cdot M=4.5\approx 5\\N'=k_y\cdot N=3.6\approx 4\end{cases}$.

(2) 将新图像的点 (x',y') 作缩放变换的逆变换 $\begin{cases}x=\dfrac{x'}{k_x}\\y=\dfrac{y'}{k_y}\end{cases}$, 变换到原图的点.

下面省略四舍五入.

$\left[\dfrac{0}{0.75},\dfrac{1}{0.75},\dfrac{2}{0.75},\dfrac{3}{0.75},\dfrac{4}{0.75}\right]=[0,1,3,4,5]$,

即新图的列 $x'\in[0,4]$ 对应原图的列 $x\in[0,1,3,4,5]$.

$\left[\dfrac{0}{0.6},\dfrac{1}{0.6},\dfrac{2}{0.6},\dfrac{3}{0.6}\right]=[0,2,3,5]$,

即新图的行 $y'\in[0,3]$ 对应原图的行 $y\in[0,2,3,5]$.

(3) 去掉原图像的第 2 列、第 1 行、第 4 行即得新图像 $g(x,y)=\begin{bmatrix}1&2&4&5&6\\13&14&16&17&18\\19&20&22&23&24\\31&32&34&35&36\end{bmatrix}$.

[2] 以求新图像中的点 $(2,2)$ 处的像素值为例, 其对应原图中的点 $\left(\dfrac{2}{0.75},\dfrac{2}{0.6}\right)\approx(2.67,3.33)$ 位于点 $(2,3)$ 、点 $(3,3)$ 、点 $(2,4)$ 和点 $(3,4)$ 间.

$x=2,y=3,a=0.67,b=0.33$.

$f(2,3.33)=f(2,3)+0.33\cdot[f(2,4)-f(2,3)]=21+0.33\times(27-21)\approx 22.98$.

$f(3,3.33)=f(3,3)+0.33\cdot[f(3,4)-f(3,3)]=22+0.33\times(28-22)\approx 23.98$.

$f(2.67,3.33)=f(2,3.33)+0.67\cdot[f(3,3.33)-f(2,3.33)]\approx 23.65\approx 24$.

[例 3.1.4.2] 将图像 $f(x,y)=\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\end{bmatrix}$ 按比例 $k_x=1.2$, $k_2=1.5$ 放大, 采用最近邻插值.

[解] (1) 新图像大小 $\begin{cases}M'=k_x\cdot M=3\times 1.2\approx 4\\ N'=k_y\cdot N=2\times 1.5=3\end{cases}$.

(2) 将新图像的点 (x',y') 作缩放变换的逆变换 $\begin{cases}x=\dfrac{x'}{k_x}\\ y=\dfrac{y'}{k_y}\end{cases}$, 变换到原图的点.

下面省略四舍五入.

$\left[\dfrac{0}{1.2},\dfrac{1}{1.2},\dfrac{2}{1.2},\dfrac{3}{1.2}\right]=[0,1,2,2]$,

即新图的列 $x'\in[0,3]$ 对应原图的列 $x\in[0,1,2,2]$,

其中 $\dfrac{3}{1.2}=2.5$ 四舍五入为 3 , 需与原图像的最大 x 值 2 取 $\min$.

$\left[\dfrac{0}{1.5},\dfrac{1}{1.5},\dfrac{2}{1.5}\right]=[0,1,1]$,

即新图的列 $y'\in[0,2]$ 对应原图的列 $y\in[0,1,1]$.

(3) 将原图像对应行列的像素值填入即得新图像:

$$g(x,y)=\begin{bmatrix}f(0,0) & f(1,0) & f(2,0) & f(2,0)\\ f(0,1) & f(1,1) & f(2,1) & f(2,1)\\ f(0,1) & f(1,1) & f(2,1) & f(2,1)\end{bmatrix}=\begin{bmatrix}1 & 2 & 3 & 3\\ 4 & 5 & 6 & 6\\ 4 & 5 & 6 & 6\end{bmatrix}.$$

3.2 图像的代数运算

(看一眼) p54 ~58 各种图像的代数运算和应用.

[定理 3.2.1] 多幅图像求均值可降低加性噪声. 具体地, 对图像 $f(x,y)$ 及其噪声图像集 $\{g_i(x,y)\}\ \ (i=1,\cdots,m)$, $g_i(x,y)=f(x,y)+n_i(x,y)$, 其中 $n_i(x,y)$ 是第 i 帧图像的噪声分布. 噪声分布 $n(x,y)$ 的均值为 0 , 方差为 σ_n^2 , 且 $n(x,y)$ 中不同点 (x,y) 处的噪声分布互不相关, 则 m 个图像的均值 $\overline{g}(x,y)=\dfrac{1}{m}\sum\limits_{i=1}^m g_i(x,y)$.

[定理 3.2.2] 对两个图像求均值可得到二次曝光效果. 具体地, 设两图像分别为 $f_1(x,y)$ 和 $f_2(x,y)$, 则它们的均值 $g(x,y)=\dfrac{1}{2}f_1(x,y)+\dfrac{1}{2}f_2(x,y)$.

3.3 邻域与模板运算

(看一眼) p58 ~ 60 邻域与模板运算的定义.
