

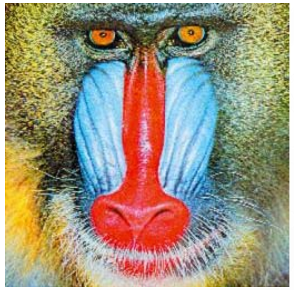


数据可视化
图像数据基本变换4
几何变换

庄吓海

大数据学院

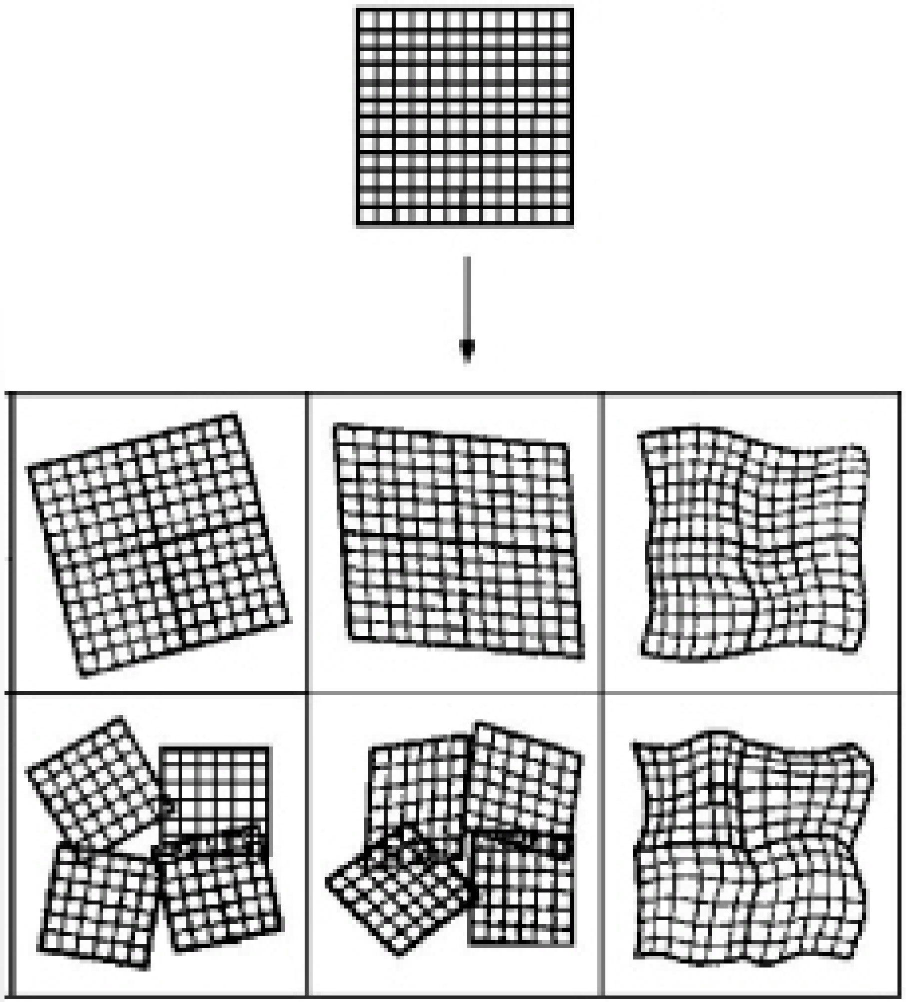
图像的几何变换



[Images courtesy of David J. Hawkes]

图像的几何变换

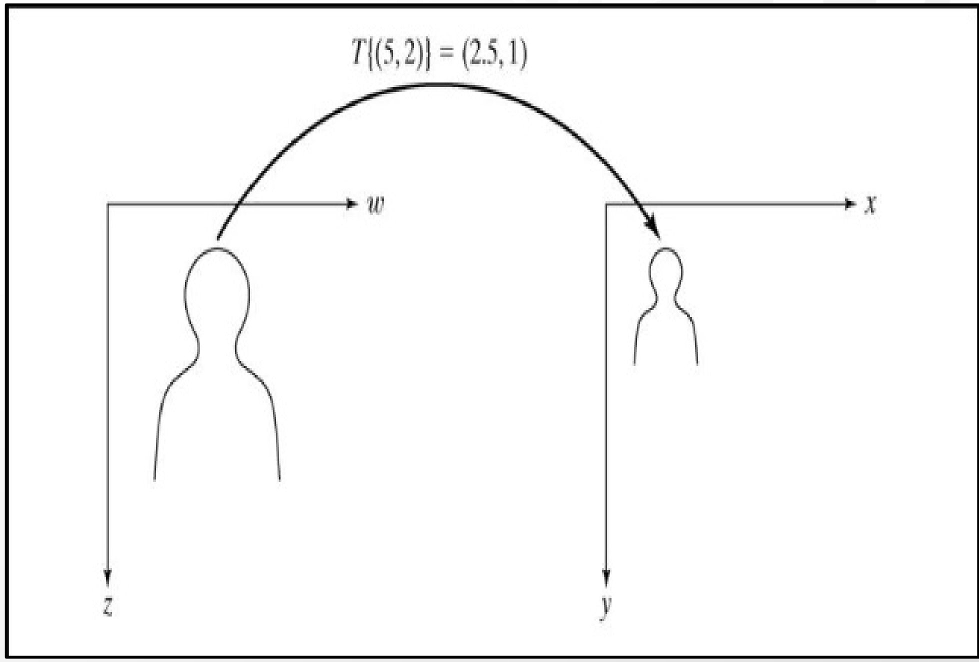
- Linear transformation
  - Rigid transformation
  - Similarity transformation
  - Affine transformation
- Nonlinear transformation
  - FFD
  - Locally affine
- Global VS Local



图像的几何变换

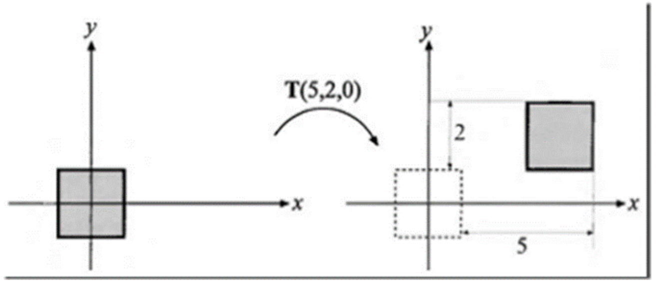
- $X' = T(X)$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \left| \begin{bmatrix} & T & \end{bmatrix} \right] \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

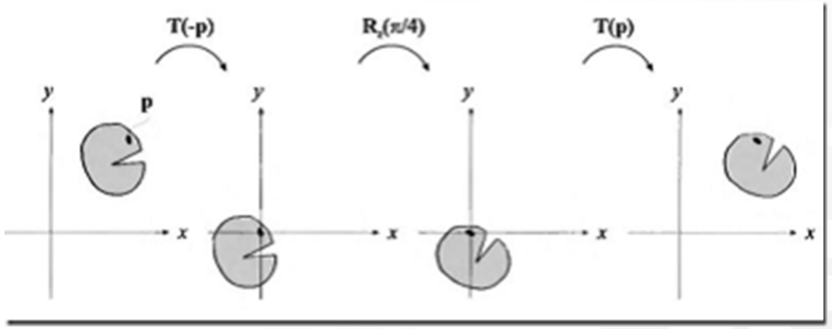


线性变换-刚性变换

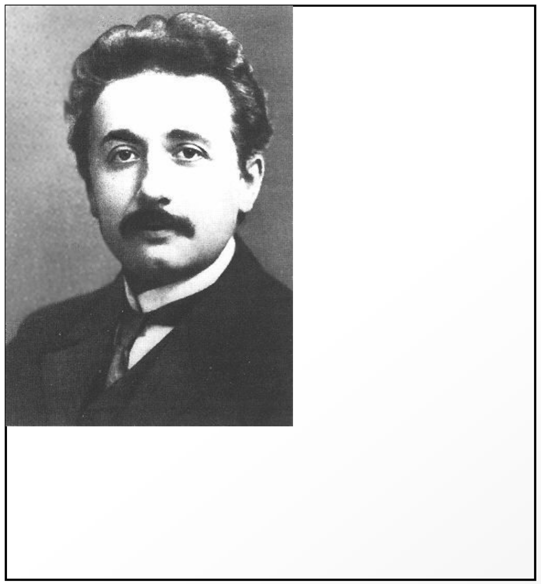
Rigid transformation preserves the angels and distances within the model

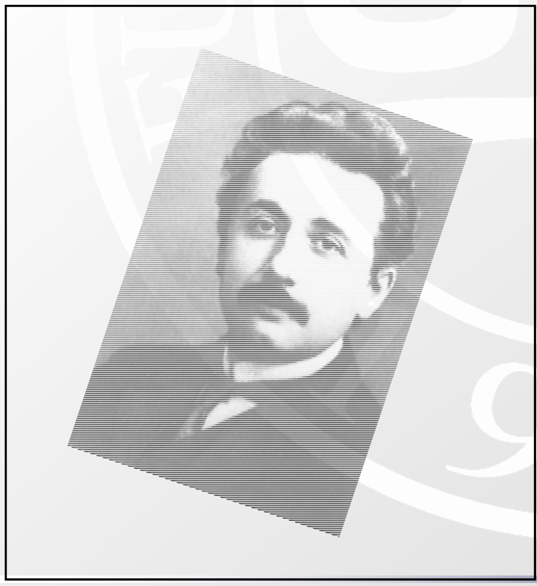


Translation



Rotation





线性变换-刚性变换

大数据学院 School of Data Science

- A 3D rigid body transform is defined by:

  --3 translations - in X, Y & Z directions

  --3 rotations - about X, Y & Z axes

- The order of the operations matters

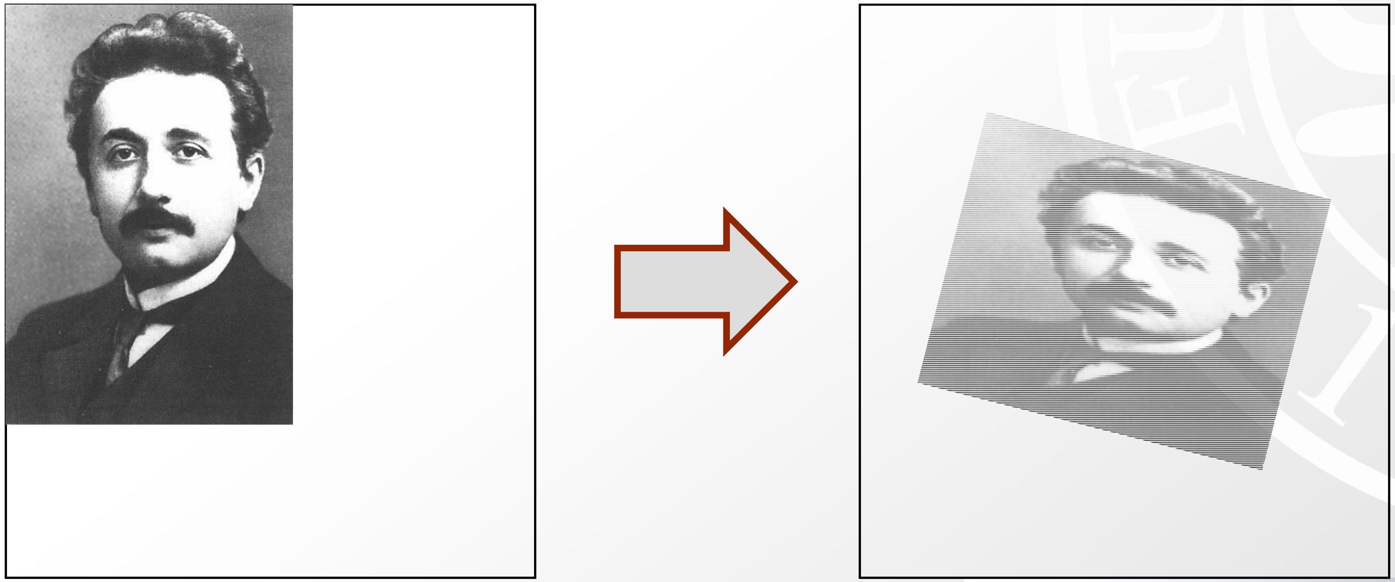
$$\begin{pmatrix} 1 & 0 & 0 & X_{trans} \\ 0 & 1 & 0 & Y_{trans} \\ 0 & 0 & 1 & Z_{trans} \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\Phi & \sin\Phi & 0 \\ 0 & -\sin\Phi & \cos\Phi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} \cos\Theta & 0 & \sin\Theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\Theta & 0 & \cos\Theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} \cos\Omega & \sin\Omega & 0 & 0 \\ -\sin\Omega & \cos\Omega & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Translations | Pitch about x axis | Roll about y axis | Yaw about z axis

线性变换–相似变换

Similarity transformation adds scaling $\{s\}$

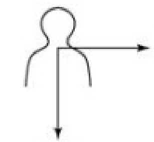
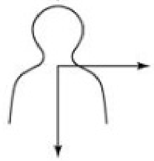
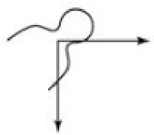
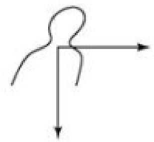
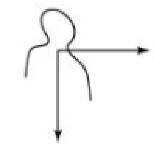
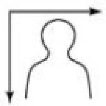
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_0 & 0 & 0 & 0 \\ 0 & s_1 & 0 & 0 \\ 0 & 0 & s_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} T_{rigid} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$



线性变换-仿射变换

Affine transformation applies a function between affine spaces which preserves points, straight lines and planes.

Parallel lines remain parallel

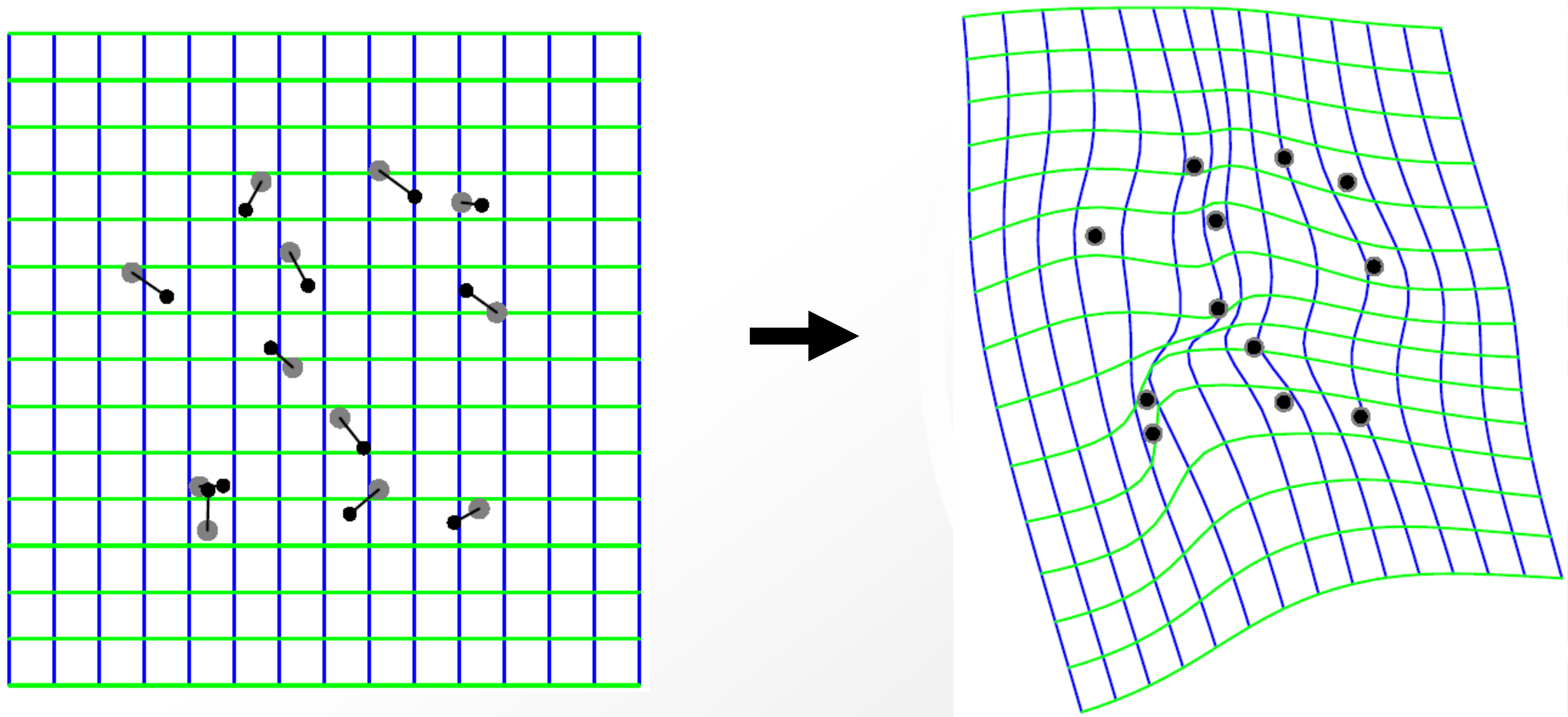
Type	Affine Matrix, T	Coordinate Equations	Diagram
Identity	$\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}$	$x = w$ $y = z$	
Scaling	$\begin{bmatrix}s_x&0&0\\0&s_y&0\\0&0&1\end{bmatrix}$	$x = s_x w$ $y = s_y z$	
Rotation	$\begin{bmatrix}\cos\theta&\sin\theta&0\\-\sin\theta&\cos\theta&0\\0&0&1\end{bmatrix}$	$x = w\cos\theta - z\sin\theta$ $y = w\sin\theta + z\cos\theta$	
Shear (horizontal)	$\begin{bmatrix}1&0&0\\\alpha&1&0\\0&0&1\end{bmatrix}$	$x = w + \alpha z$ $y = z$	
Shear (vertical)	$\begin{bmatrix}1&\beta&0\\0&1&0\\0&0&1\end{bmatrix}$	$x = w$ $y = \beta w + z$	
Translation	$\begin{bmatrix}1&0&0\\0&1&0\\\delta_x&\delta_y&1\end{bmatrix}$	$x = w + \delta_x$ $y = z + \delta_y$	

$$\begin{bmatrix}x'\\y'\\z'\\1\end{bmatrix}=\begin{bmatrix}e_{11}&e_{12}&e_{13}&e_{14}\\e_{21}&e_{22}&e_{23}&e_{24}\\e_{31}&e_{32}&e_{33}&e_{34}\\0&0&0&1\end{bmatrix}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}$$

非线性变换-FFD

大数据学院 School of Data Science

Thin-Plate Spline



非线性变换-FFD

- Thin-Plate Spline
- Free-form

(From [S.Y. Lee, K.Y Chwa, and S.Y. Shin, SIGGRAPH, 1995])

非线性变换-FFD

- 自由形变变换（Free-form deformation, FFD）

Compute lattice coordinates

(u,v,w)

Alter the control points

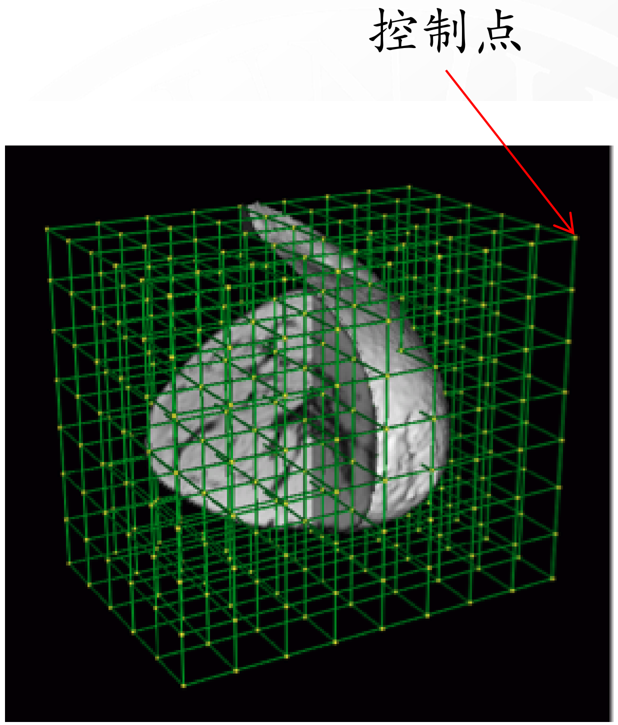
$\mathbf{p}_{ijk}$

Compute the deformed points

$\mathbf{Q}(u,v,w)$

$$\mathbf{Q}(u,v,w)=\sum_{ijk}\mathbf{p}_{ijk}B(u)B(v)B(w)$$

$$X' = X + Q$$



Free form deformation (FFD)

非线性变换-FFD

大数据学院 School of Data Science

- 自由形变变换（Free-form deformation, FFD）

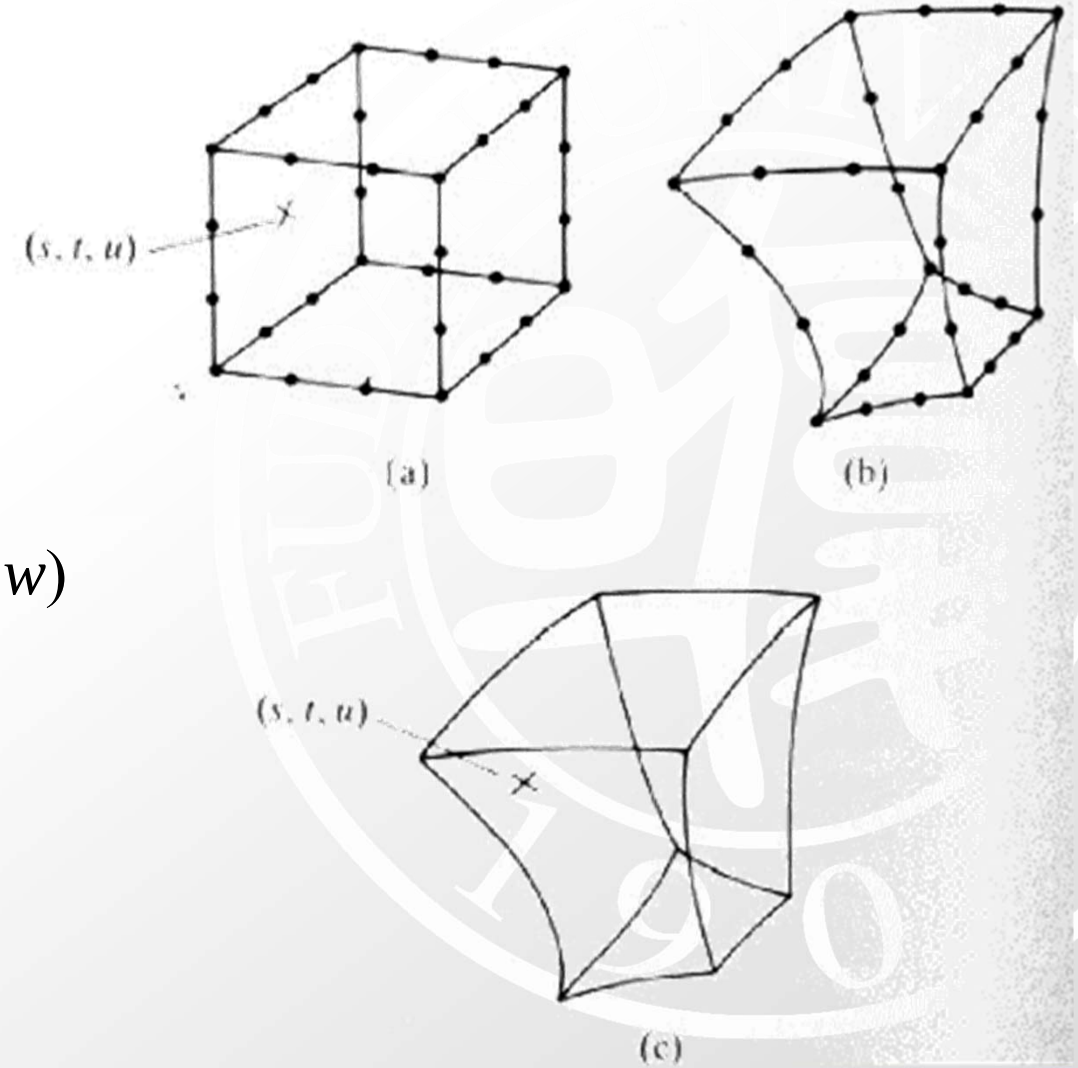
Compute lattice coordinates
(u,v,w)

Alter the control points
$\mathbf{p}_{ijk}$

Compute the deformed points
$\mathbf{Q}(u,v,w)$

$$\mathbf{Q}(u,v,w)=\sum_{ijk}\mathbf{p}_{ijk}B(u)B(v)B(w)$$

$$X' = X + Q$$



非线性变换-FFD

大数据学院 School of Data Science

- 自由形变变换（Free-form deformation, FFD）

Daniel Rueckert

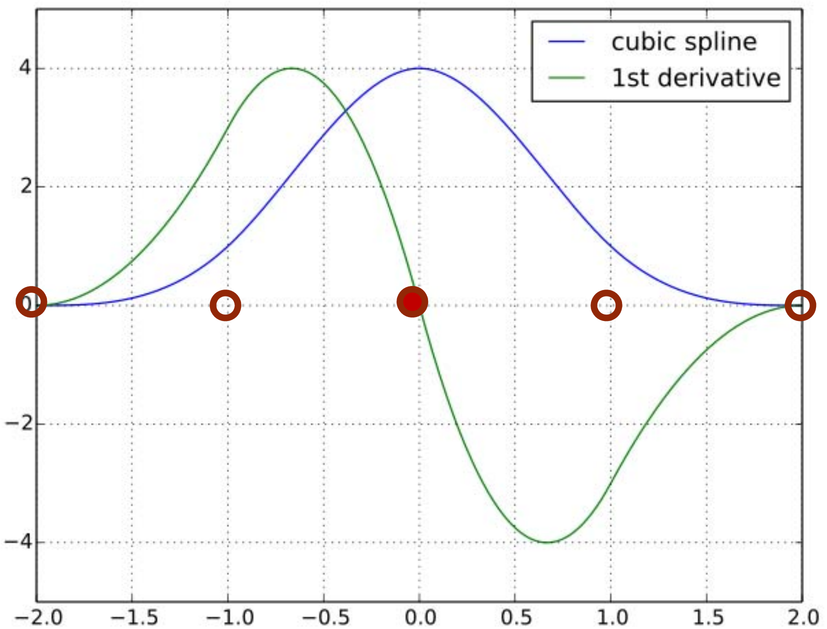
Follow

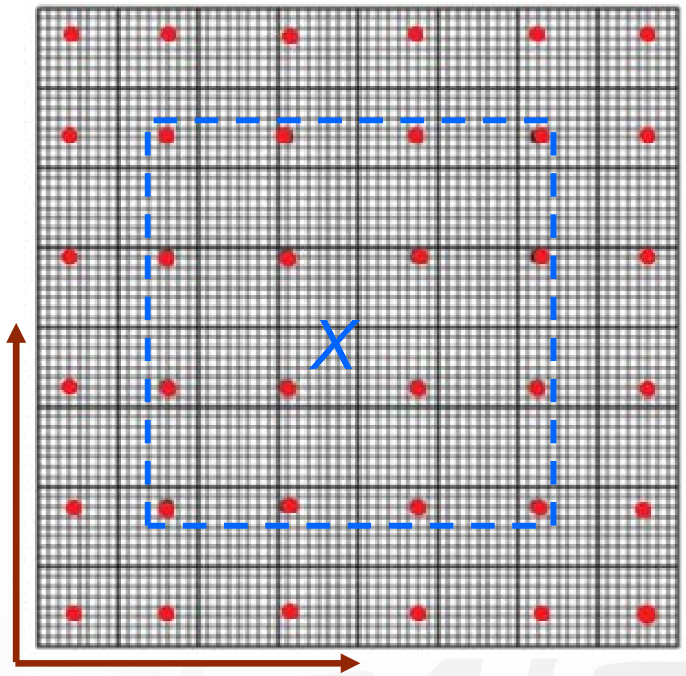
Professor of Visual Information Processing, Imperial College London
Medical Image Computing, Medical Image Analysis, Biomedical Image Analysis, Medical Imaging, Neuroimaging
Verified email at imperial.ac.uk - Homepage

Title 1–20	Cited by	Year
Nonrigid registration using free-form deformations: application to breast MR images D Rueckert, LI Sonoda, C Hayes, DLG Hill, MO Leach, DJ Hawkes IEEE transactions on medical imaging 18 (8), 712-721	4362	1999
Tract-based spatial statistics: voxelwise analysis of multi-subject diffusion data SM Smith, M Jenkinson, H Johansen-Berg, D Rueckert, TE Nichols, ... Neuroimage 31 (4), 1487-1505	3309	2006
Evaluation of 14 nonlinear deformation algorithms applied to human brain MRI registration A Klein, J Andersson, BA Ardekani, J Ashburner, B Avants, MC Chiang, ... Neuroimage 46 (3), 786-802	1267	2009
Automatic anatomical brain MRI segmentation combining label propagation and decision fusion RA Heckemann, JV Hajnal, P Aljabar, D Rueckert, A Hammers NeuroImage 33 (1), 115-126	714	2006
Multi-atlas based segmentation of brain images: atlas selection and its effect on accuracy P Aljabar, RA Heckemann, A Hammers, JV Hajnal, D Rueckert	559	2009

非线性变换-FFD

- 使用三次B-样条核函数
- $X' = X + T_{\text{local}}$





$$\mathbf{T}_{\text{local}}(x, y, z) = \sum_{l=0}^{3}\sum_{m=0}^{3}\sum_{n=0}^{3} B_l(u)B_m(v)B_n(w)\phi_{i+l,\, j+m,\, k+n} \quad (3)$$

where $i = \lfloor x/n_x \rfloor - 1$, $j = \lfloor y/n_y \rfloor - 1$, $k = \lfloor z/n_z \rfloor - 1$, $u = x/n_x - \lfloor x/n_x \rfloor$, $v = y/n_y - \lfloor y/n_y \rfloor$, $w = z/n_z - \lfloor z/n_z \rfloor$ and where B_l represents the lth basis function of the B-spline [22], [23]

$$B_0(u) = (1-u)^3/6$$
$$B_1(u) = (3u^3 - 6u^2 + 4)/6$$
$$B_2(u) = (-3u^3 + 3u^2 + 3u + 1)/6$$
$$B_3(u) = u^3/6.$$

D Rueckert, LI Sonoda, C Hayes, DLG Hill, MO Leach, DJ Hawkes: Nonrigid registration using free-form deformations: application to breast MR images. IEEE transactions on medical imaging 18 (8), 712-721

非线性变换-FFD

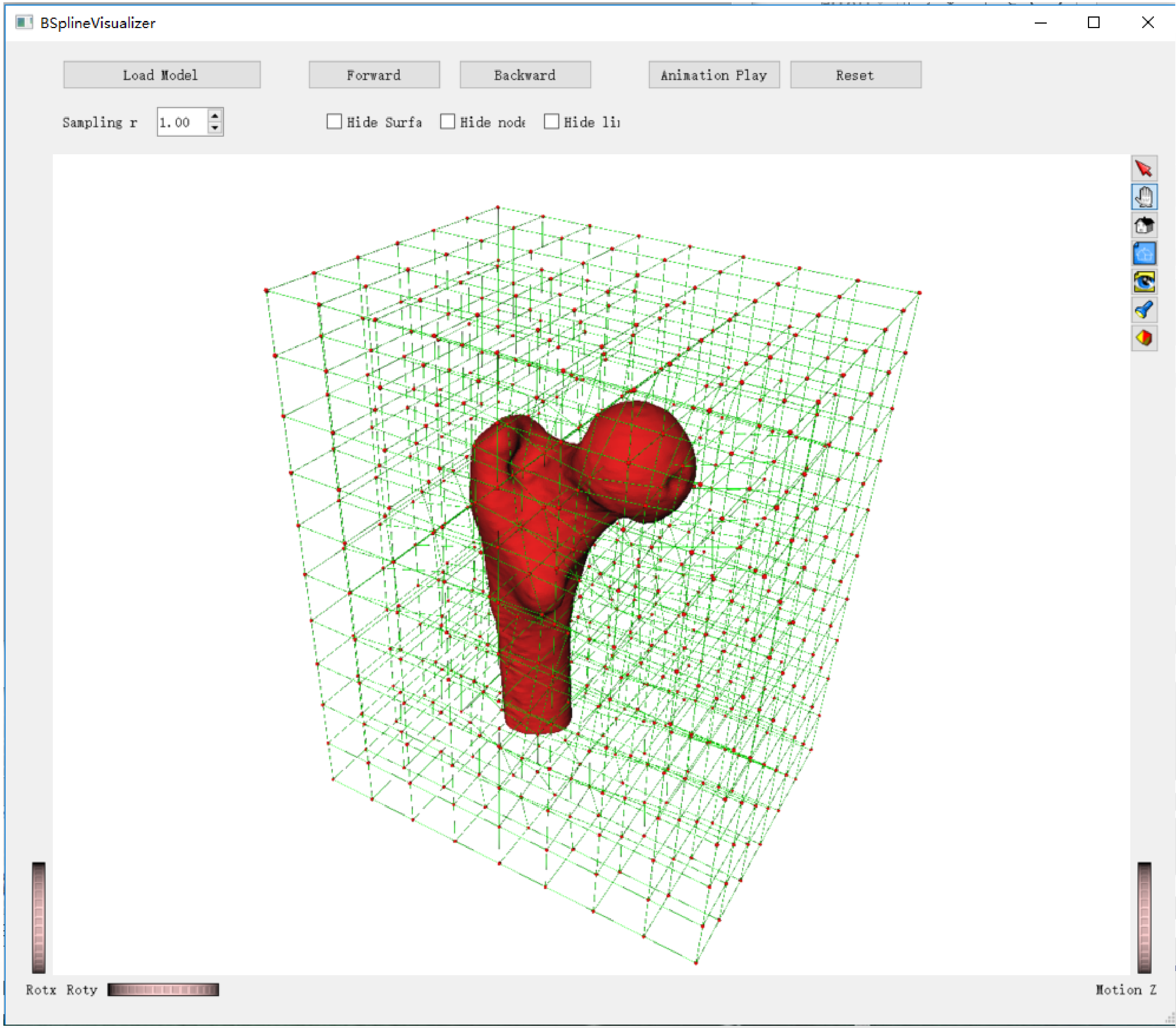
大数据学院 School of Data Science

```
void  zxhTransformFFDBase::TransformPhysToPhys( const float fFrom[],float fTo[])const
{
  int iGrid[ImageDimensionMax] = {0,0,0,0} ;
  float fu[ImageDimensionMax]={0,0,0,0}, afOffset[ImageDimensionMax]={0,0,0,0};
  for(int index=0;index<m_iDimension;++index)
  {
    iGrid[index]= static_cast<int>(floor(fFrom[index]));
    fu[index]   = fFrom[index]-floor(fFrom[index]);
  }
  if(m_iDimension==3)
  {
    float fBSpline_u[4]={0,0,0,0},fBSpline_v[4]={0,0,0,0},fBSpline_w[4]={0,0,0,0};
    for(int ite=0;ite<4;++ite)
    {
      fBSpline_u[ite]  = BSplinei(ite-1,fu[0]);
      fBSpline_v[ite]  = BSplinei(ite-1,fu[1]);
      fBSpline_w[ite]  = BSplinei(ite-1,fu[2]);
    }
    for(int i=-1;i<=2;i++)
    for(int j=-1;j<=2;j++)
    for(int k=-1;k<=2;k++)
    {
      float* poff  = GetCtrPnt(iGrid[0]+i,iGrid[1]+j,iGrid[2]+k);
      if(poff!=0)
      {
        float w=fBSpline_u[i+1]*fBSpline_v[j+1]*fBSpline_w[k+1];
        for(int idx=0;idx<m_iDimension;++idx)
          afOffset[idx]  += poff[idx]*w;
      }
    }
  }else{}
  for(int idx=0;idx<m_iDimension;++idx)
    fTo[idx]  = fFrom[idx]+afOffset[idx] ;
  return  ;
}
```

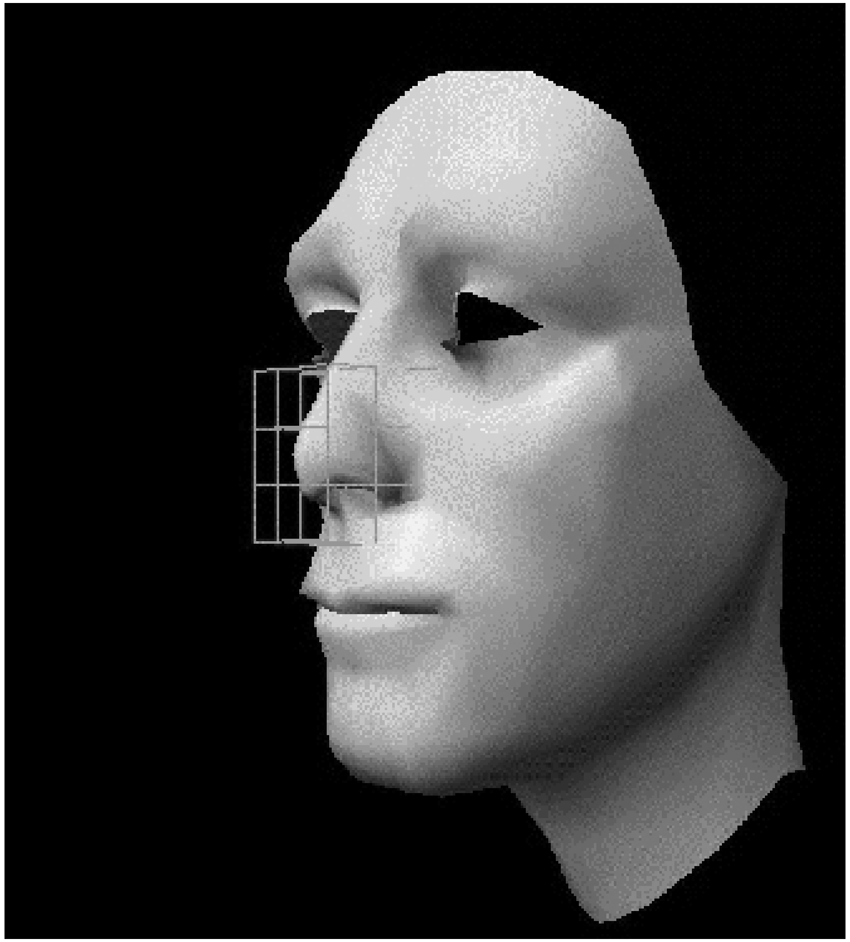
```
inline float  BSplinei(int iOrd,float u) const
{
  float v;
  switch(iOrd)
  {
  case -1:
    v=1-u;
    return v*v*v/6.0f;
  case 0:
    v=u*u*3.0f;
    return (v*u-v*2.0f+4.0f)/6.0f;
  case 1:
    v=3.0f*u*u;
    return (-v*u+v+3.0f*u+1.0f)/6.0f;
  case 2:return u*u*u/6.0f;
  }
  return 0;
};
```

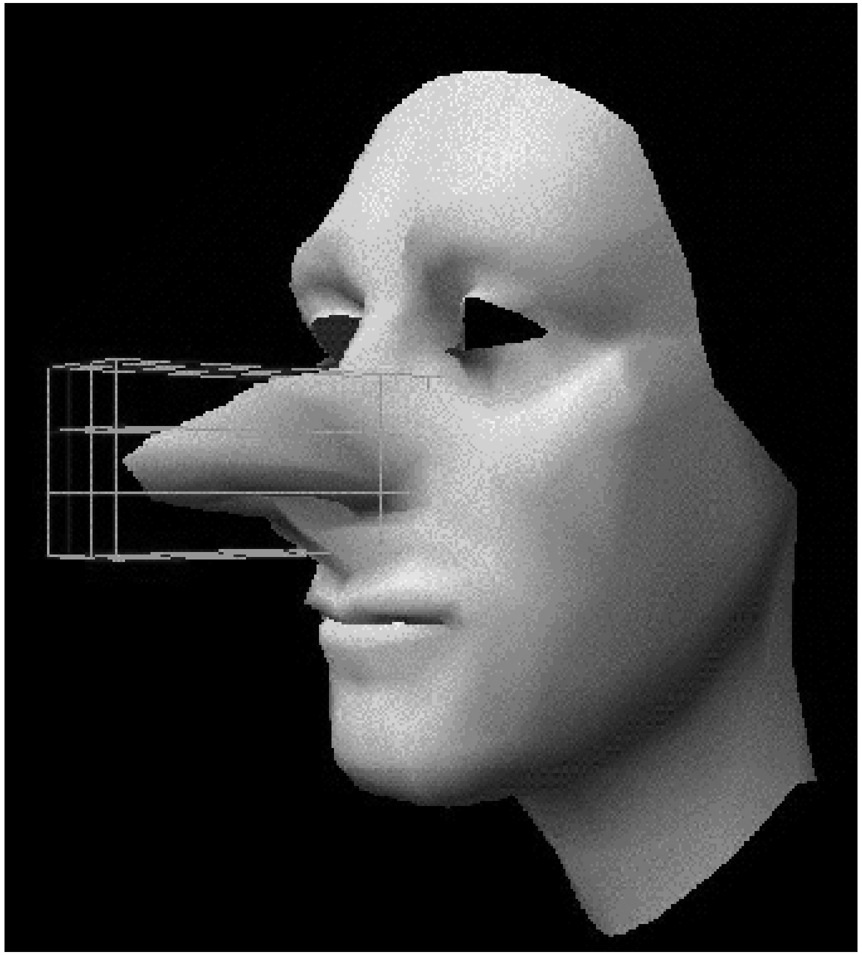
非线性变换-FFD

大数据学院
School of Data Science

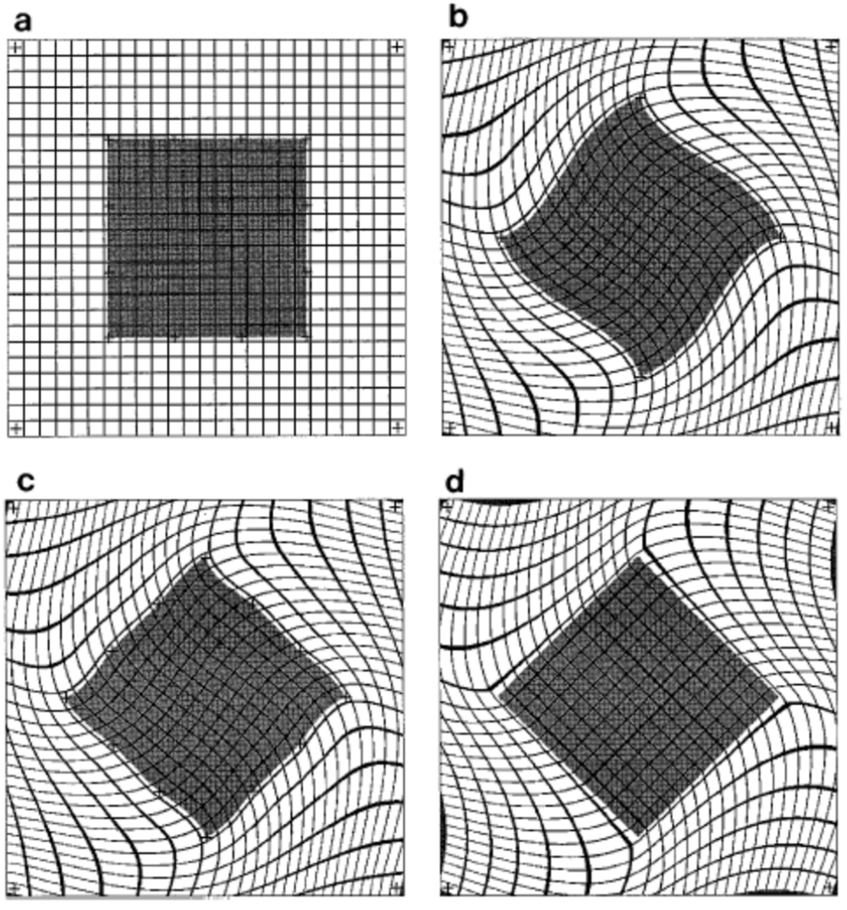


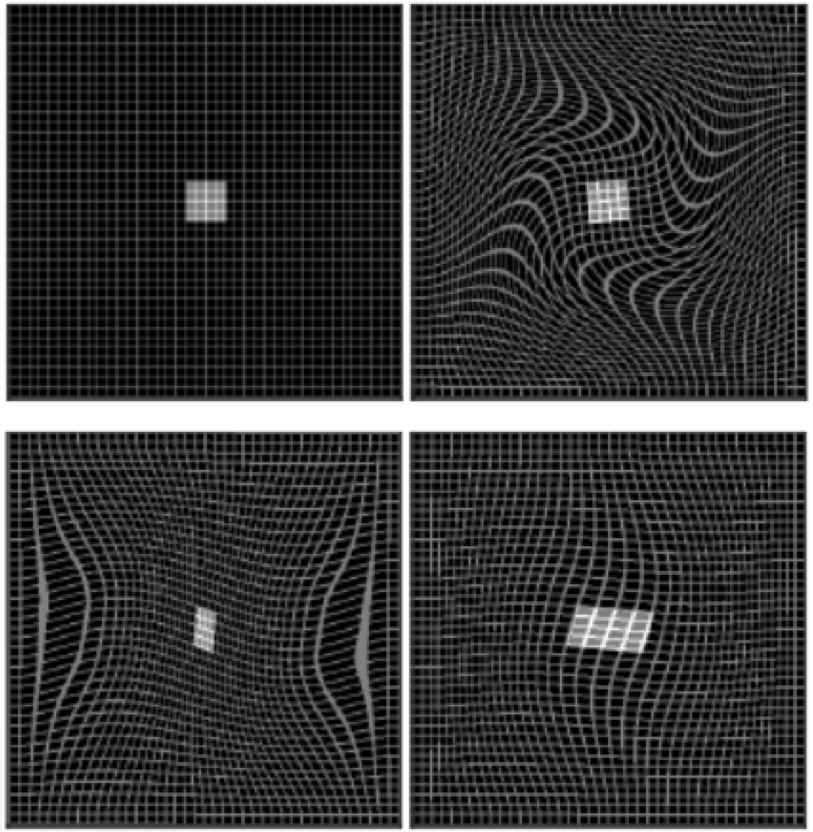
非线性变换-FFD



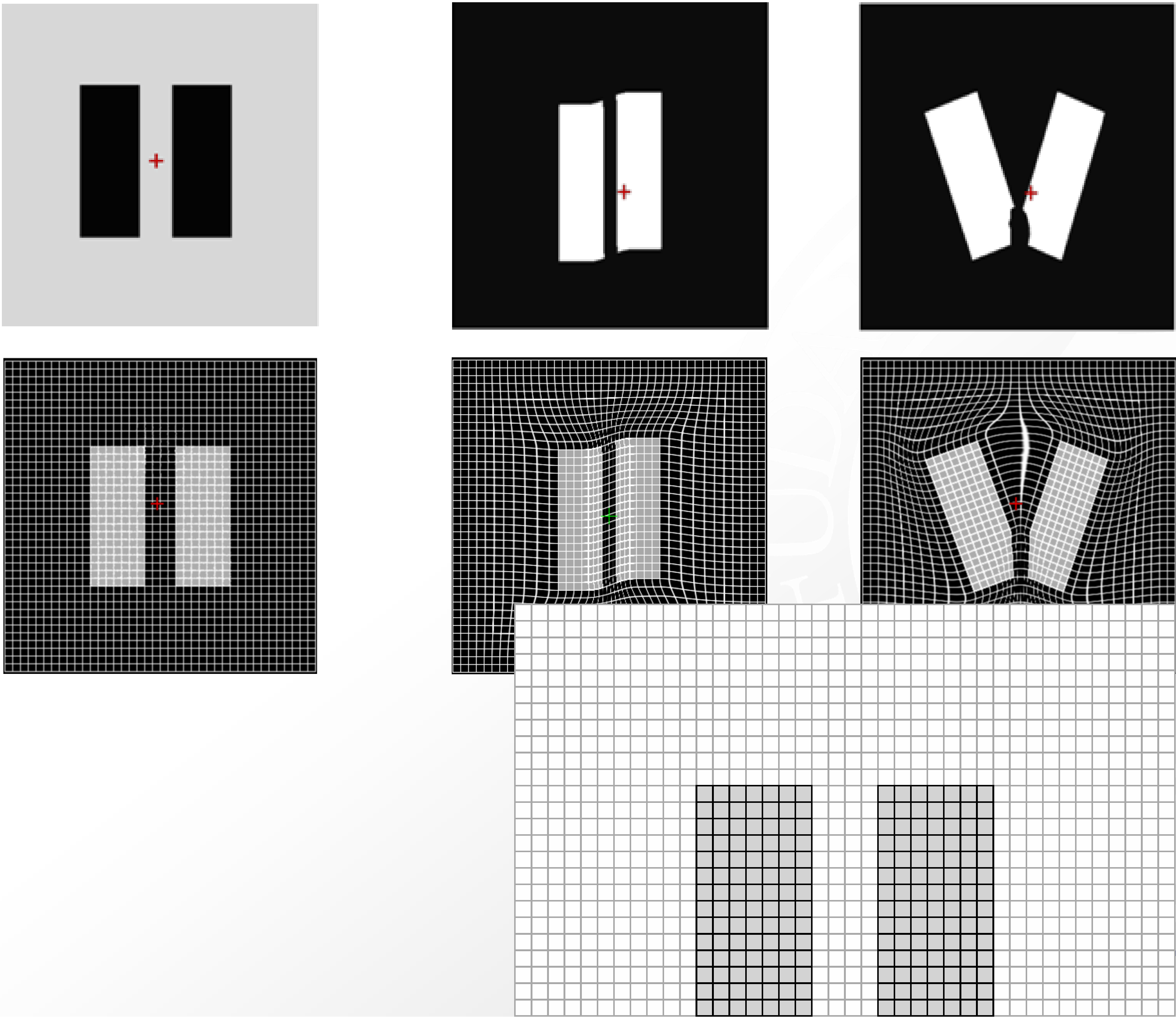


局部仿射





局部仿射



局部仿射

- 假设有n个局部区域（坐标点）U_i，每个区域对应的局部仿射变换为G_i，则局部仿射空间变换公式由下式计算

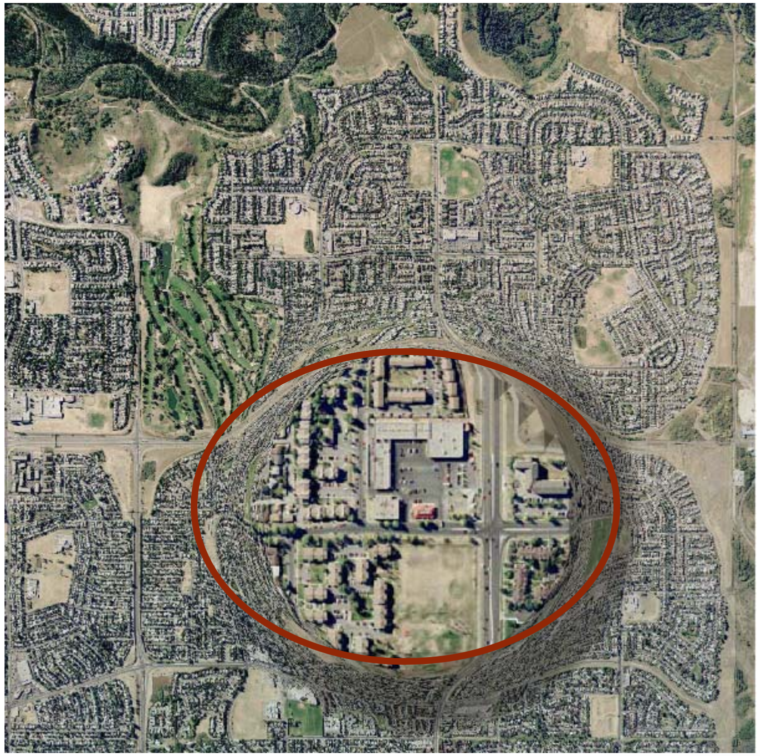
$$T(X)=\begin{cases} G_i(X), & X\in U_i,\ i=1\ldots n \\ \sum_{i=1}^{n} w_i(X)G_i(X_i), & X\notin \bigcup_{i=1}^{n} U_i \end{cases}$$

$$w_i(X)=\left(1/d_i(X)^e\right)\Big/\left(\sum_{i=1}^{n} 1/d_i(X)^e\right)$$

- 其中$d_i(X)$表示X到U_i的距离。

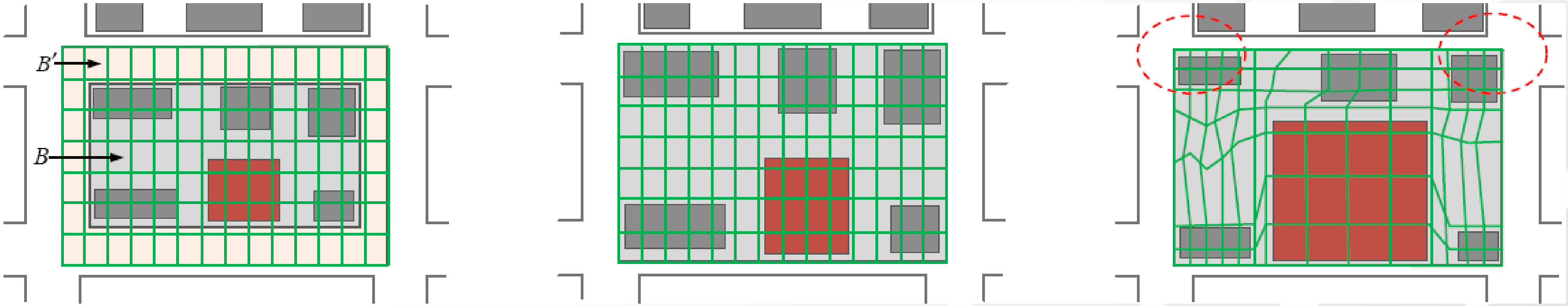
X Zhuang, KS Rhode, RS Razavi, DJ Hawkes, S Ourselin: A registration-based propagation framework for automatic whole heart segmentation of cardiac MRI. IEEE Transactions on Medical Imaging, 29 (9), 1612-1625

局部仿射：地图可视化

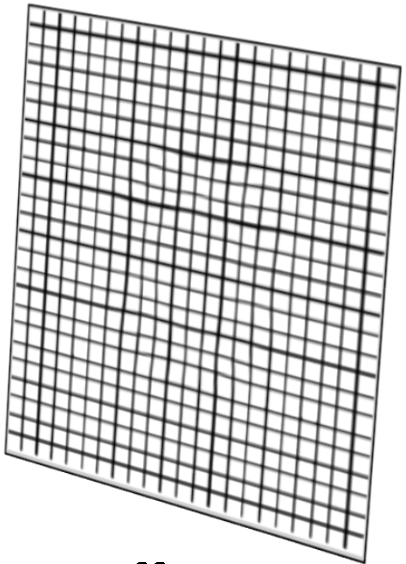


局部仿射：地图可视化

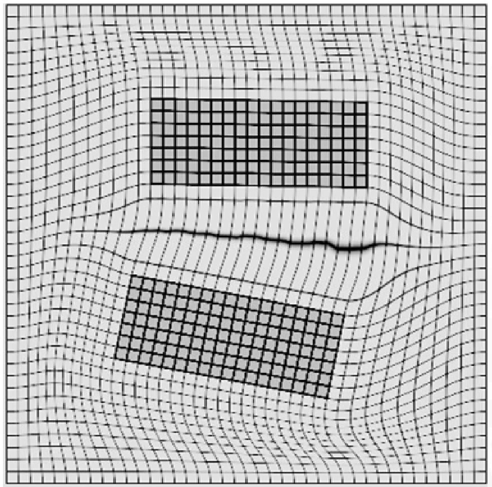
- 可能做法：
  - 放大整个街区，鱼眼放大
- 基于网格，有针对性的放大
  - 空间利用率更高，没有变形



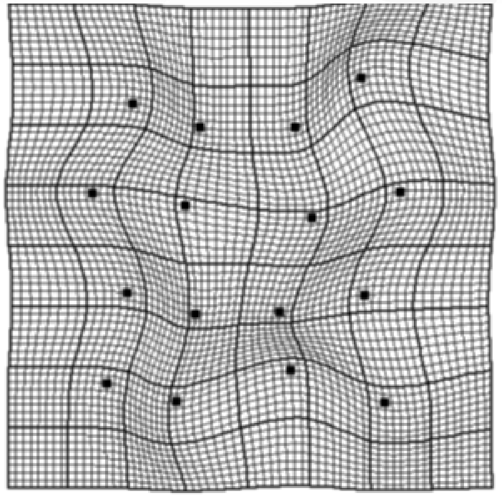
总结



affine

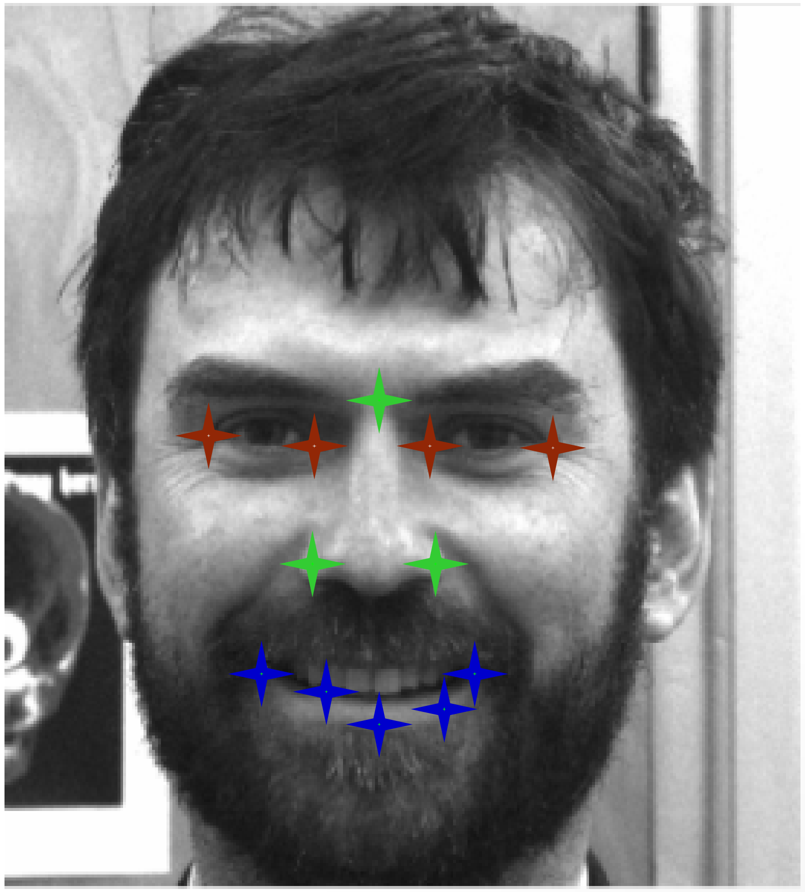


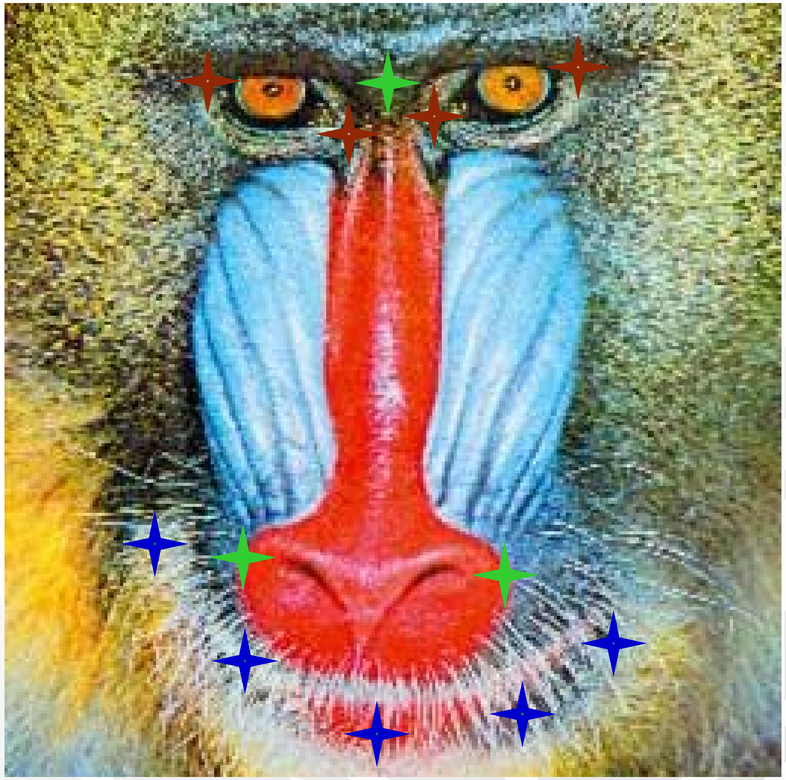
locally affine



deformable

数据处理与变换





[Images courtesy of David J. Hawkes]

1 设置对应点或区域，比如左右眼睛各2个点，鼻子3个点，嘴巴5个点，共12个对应点的局部变换（线性）；

2* 利用局部仿射变换算法，计算出狒狒图像中每个像素$X=(i,j)$ 对应到院士图像的坐标 $T(X)=(x,y)$

$$T(X)=\begin{cases} G_i(X), & X\in U_i,\ i=1\ldots n \\ \sum_{i=1}^{n} w_i(X)G_i(X_i), & X\notin \bigcup_{i=1}^{n} U_i \end{cases} \qquad w_i(X)=\left(1/d_i(X)^e\right)\Big/\left(\sum_{i=1}^{n} 1/d_i(X)^e\right)$$

3 在院士图像上插值得出坐标(x,y)的灰度值$f(x,y)$

4 把狒狒图像中像素(i,j) 点的灰度值设置为 $f(x,y)$

5 利用2-4把狒狒图像中所有像素的灰度值重新设置灰度值，得到的图像即为形变图像

作业请用：Python，Matlab，C/C++等常见编程环境

1设计算法、交互和界面，实现人脸到狒狒的形变。作业的算法内容在课堂上讲解；参考课件（9 图像数据基本变换 5 空间变换.pdf）。

附件：狒狒和我的头像

2作业以小组形式，每个小组不能超过3个人。提交作业时只要一个代表提交就可以。记住：不要多人重复提交。

3 提交内容包括：（1）代码，如果有可执行文件同时提交可执行文件；代码要有非常清晰的注释。（2）数据（如果有用到）。（3）一份报告：在报告中清晰描述问题和数据，数据处理的各个步骤及中间结果，代码结构，开发环境，可执行文件使用手册等细节问题；要求在报告中说明每位同学的贡献和工作内容。

Thank You！