二维图像Morphing

金小刚

Email: jin@cad.zju.edu.cn

浙江大学CAD&CG国家重点实验室

紫金港校区蒙民伟楼512

《西游记》中的七十二变?

典型例子：美孚石油公司广告、抖音

二维图像morphing技术

- 图像自然渐变(Image Morphing)是指把一幅数字图像以一种自然流畅的、戏剧性的、超现实主义的方式变换到另一幅数字图像。
- 图象morphing在影视特技、教育和娱乐等方面都是一种非常有用的技术，它是一种达到特殊视觉效果的有效方法。
- 尽管图像morphing在二维图像空间处理问题，但可以让人产生神奇的三维形状改变的错觉。可以避免复杂的三维造型过程。

实现Morphing的传统技术

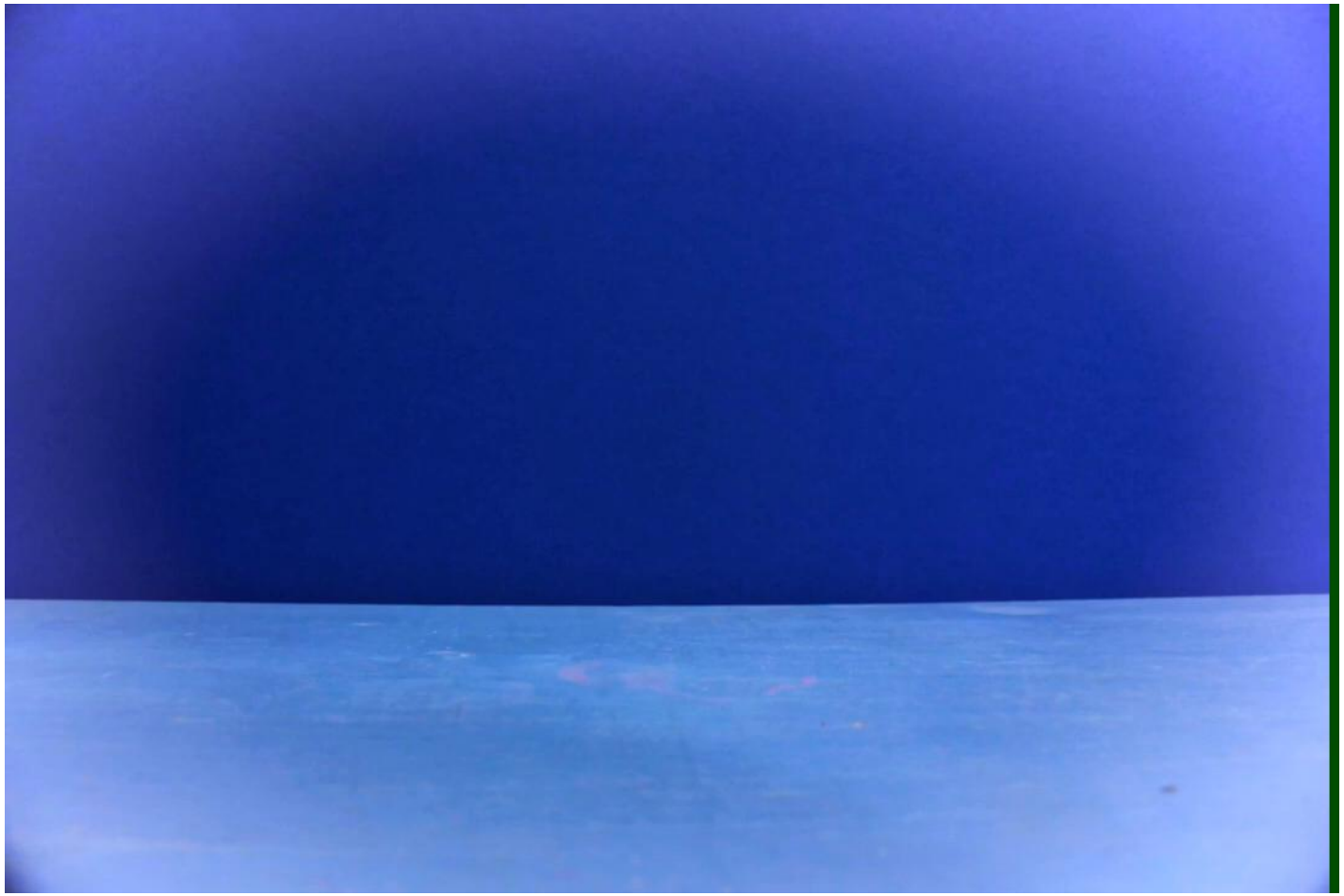
- 巧妙的剪辑
  - 如当一个人穿过几棵树或森林后变成另一个人。
- 停格运动动画法(stop-motion animation)
  - 通过对演员每化妆一次拍摄一帧的方法来达到人物逐步渐变的方法。其缺点是需要很多技巧和枯燥的工作。
- 二维粒子系统技术
  - 该技术与图象morphing类似，其主要思想是移动第一幅图象的象素块，使其逐渐解体，然后重建成第二幅图象。
- 交溶技术(cross-dissolve)
  - 或称为淡入淡出技术。即在一幅图象淡出(fade out)的同时淡入(fade in)另一幅图象。
  - 当两幅图象的几何未对齐时，该方法的效果较差。
  - 从图象处理的角度来看，交溶技术实际上是图象之间的线性插值。

实现Morphing的传统技术

stop-motion animation

二维粒子系统技术

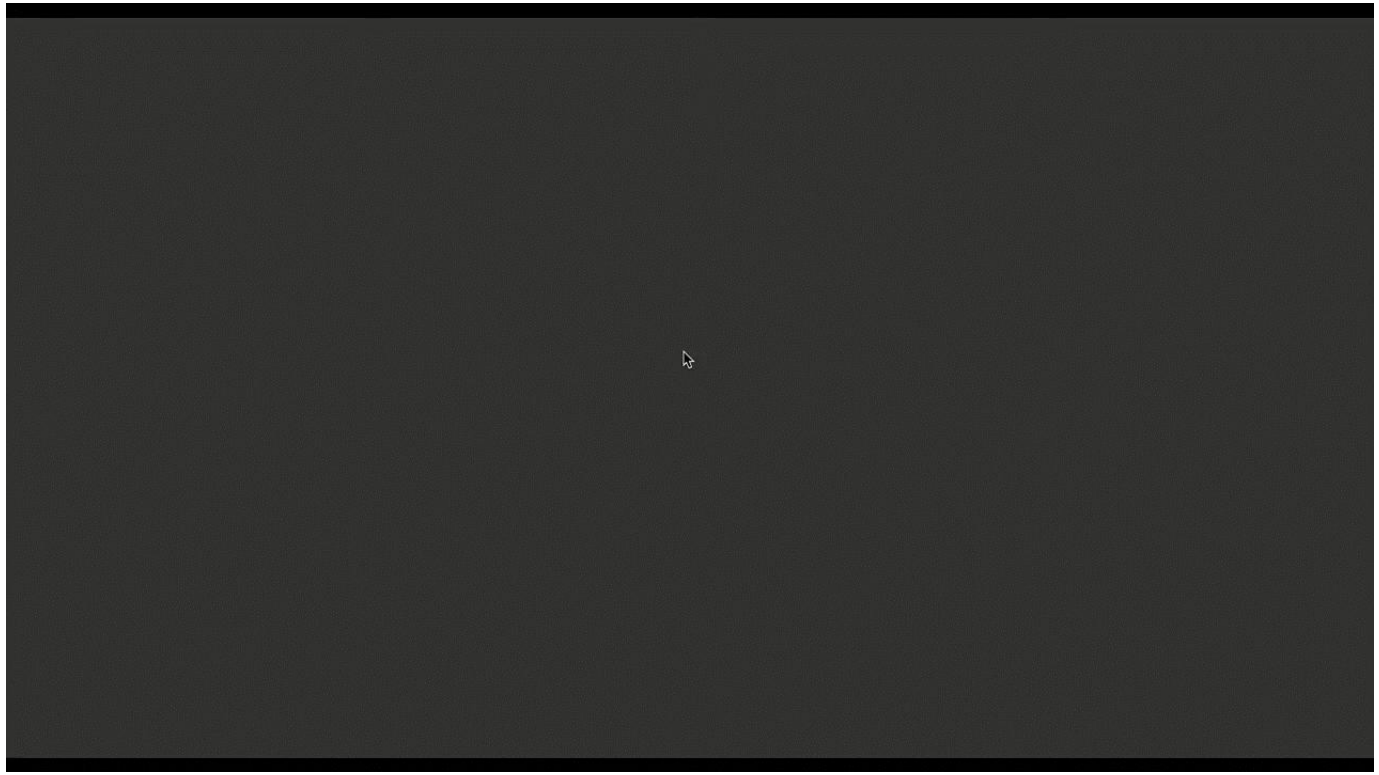
实现Morphing的传统技术



stop-motion animation

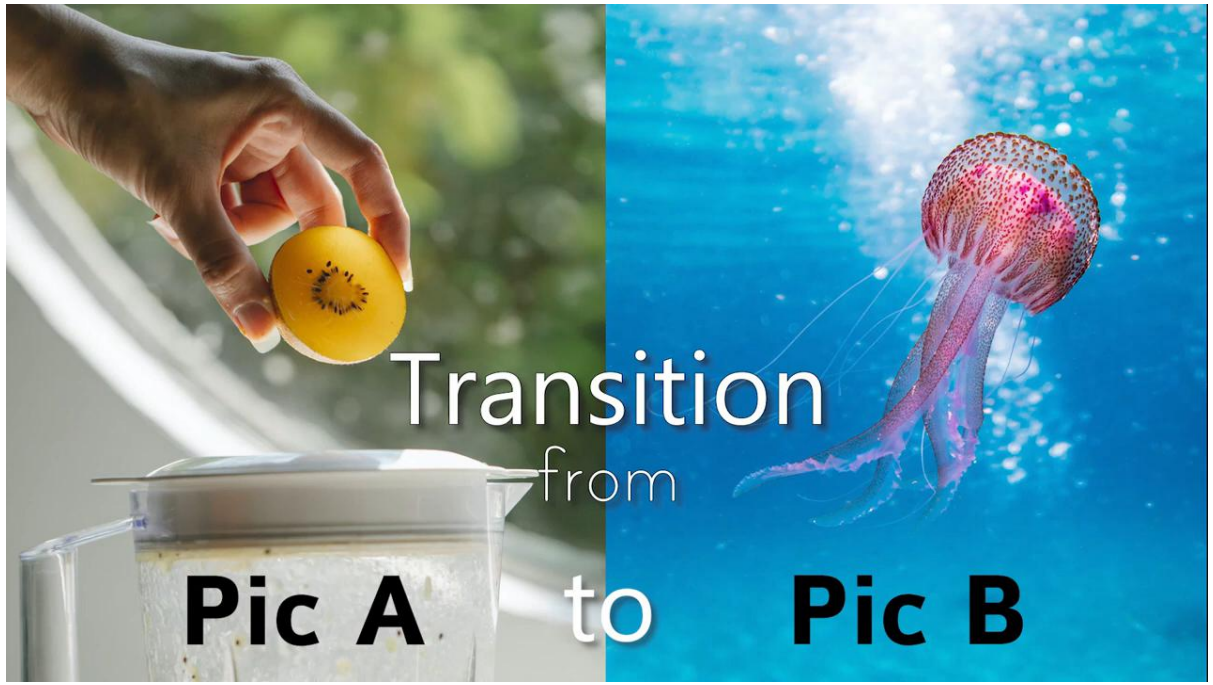
2020年8月23日 I took over **1000 photos** for this animation where I experiment with morphing! I hope you enjoy : D Let me know what you think of the saturation, I love vivid colors but am not sure if it's too much. Also, I rubbed the paint of my desk lol woops!

实现Morphing的传统技术



2D Image morphing particle system

实现Morphing的传统技术



Cross-dissolve

图像Morphing vs. 三维Morphing

- 与图象morphing相关的另一种方法是我们后面要讨论的三维morphing技术。
- 该方法的主要思想是先对一对三维的物体进行造型，然后插值对应的几何属性和非几何属性。
- 三维morphing的缺点是物体之间的对应关系很难建立，而且该方法对物体的几何表示也相当苛刻，比如要求多边形的个数相等，物体的拓扑结构相同等等。
- 当待morphing的两个物体造型简单、对应关系容易建立时，三维morphing技术是一种比较好的方法。但在许多时候，由于待morphing物体的造型很复杂，比如两个动物之间的morphing，两个人之间的morphing等等，三维morphing技术就显得比较困难。
- 在这种情况下，若虚拟摄象机的位置不变，用图象处理的技术反而会更简单。

图像morphing的过程

- 我们用warping表示单幅图象的扭曲变形。为了实现两幅二维图象I_S和I_D的morphing过程，动画师首先通过简单的几何元建立图象特征之间的对应关系。
- 几何元可以为网格节点、线段、曲线、点等，然后由这些特征对应关系计算出morphing所需的几何变换，几何变换定义了两幅图象上点之间的几何对应关系。
- 满射C_0: $I_S \rightarrow I_D$把第一幅图象的几何形状映射为第二幅图象的几何形状，满射C_1: $I_D \rightarrow I_S$把第二幅图象的几何形状映射为第一幅图象的几何形状。
- 需要两个映射的原因是图象点与点之间的对应关系不一定是一一对应。

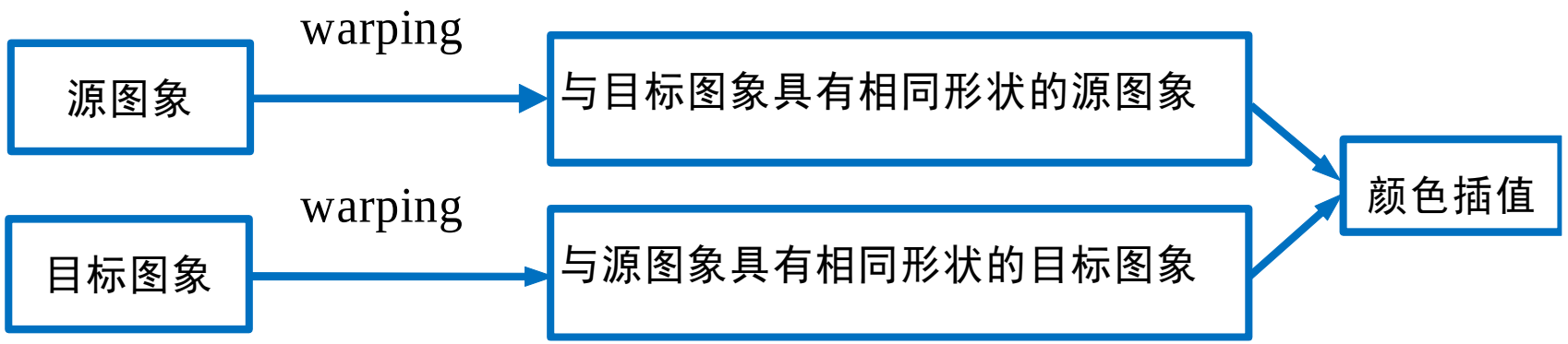
图像morphing的过程

图像morphing的过程

- 设P_0为源图象上的点，P_1为目标图象上的点，则源图象和目标图象的warping函数W_0和W_1分别定义为：

$$W_0(P_0,t)=(1-t)P_0+tC_0(P_0)$$
$$W_1(P_1,t)=(1-t)C_1(P_1)+tP_1$$
$$t\in[0,1]$$

- 当两幅图象变形对齐后，我们可采用简单的颜色插值得到中间帧图象。



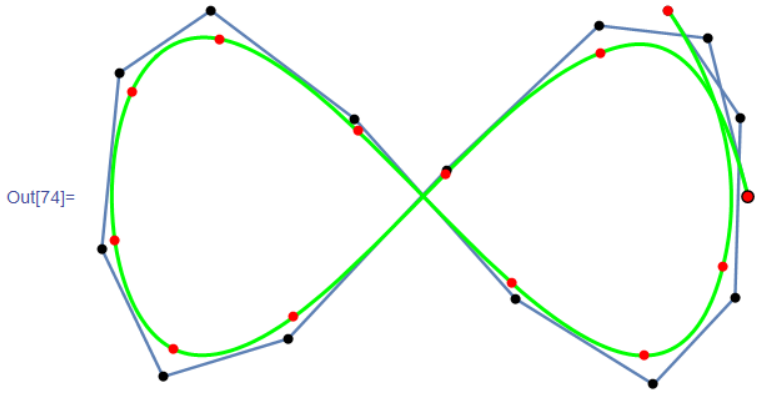
图像Morphing例子

《泰坦尼克号》中的图像Morphing效果

基于网格的图象morphing方法

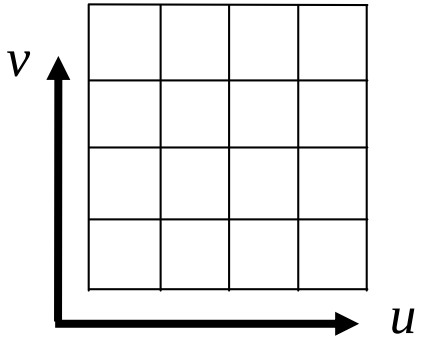
- 基于网格(Grid)的图象morphing是图象morphing中最早的方法。
- 这一方面的先驱是美国的Lucasfilms特技组Industrial Light Magic公司(ILM)的D. Smythe，在电影《Indiana Jones and the Last Crusade》（《夺宝奇兵3》）中(1989年出品)，他把该技术用于一个坏人的提前衰老死亡。
- 影片中的那个坏人以为找到了圣杯，用它喝酒，却发现找错了杯子。结果，他在数秒钟内急速变老，人变枯萎并可怕地死亡。该技术后来成功地用于多部电影中，如《Willow》、《The Abyss》等。
- 网格通常为双三次样条曲面，如Bezier样条曲面、Catmull-Rom样条曲面等。

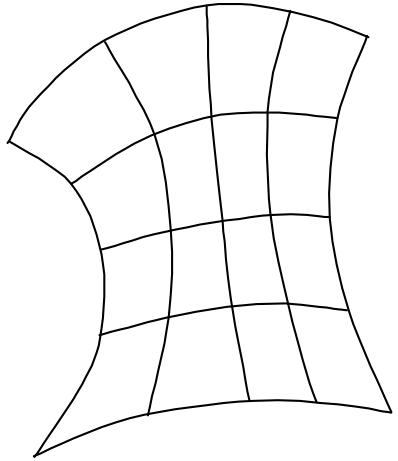
双三次样条曲面与Image Warping

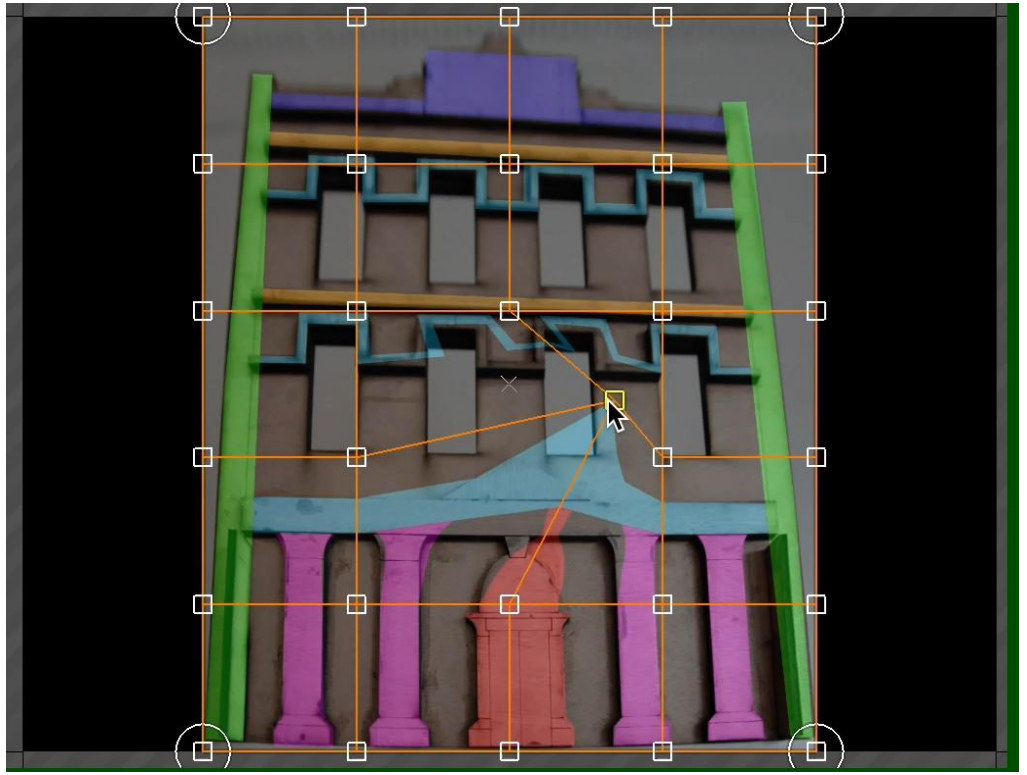


样条曲线：把一维参数空间$u \in [0,1]$映射为曲线

$(u,v) \in [0,1]\times[0,1]$

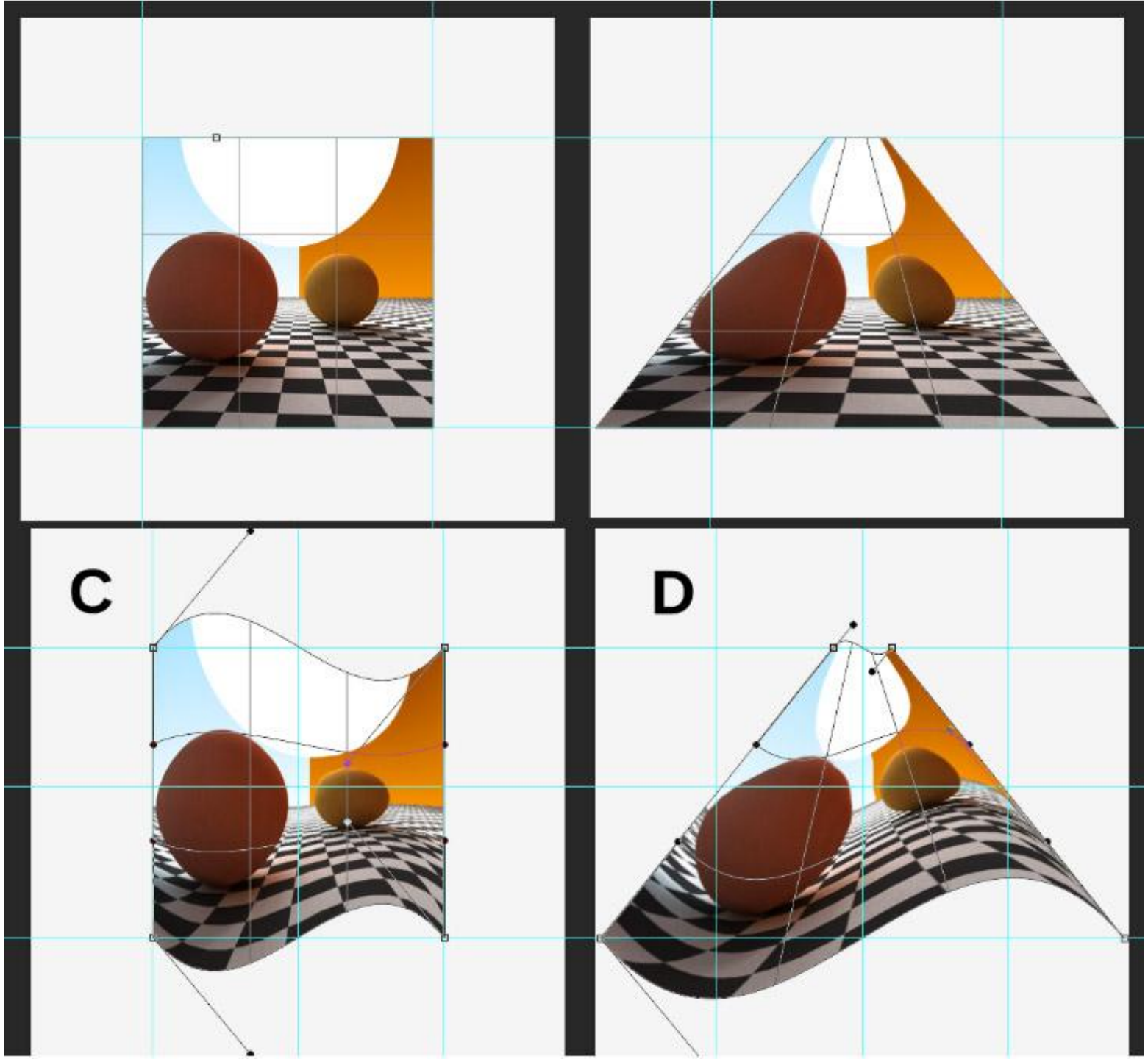


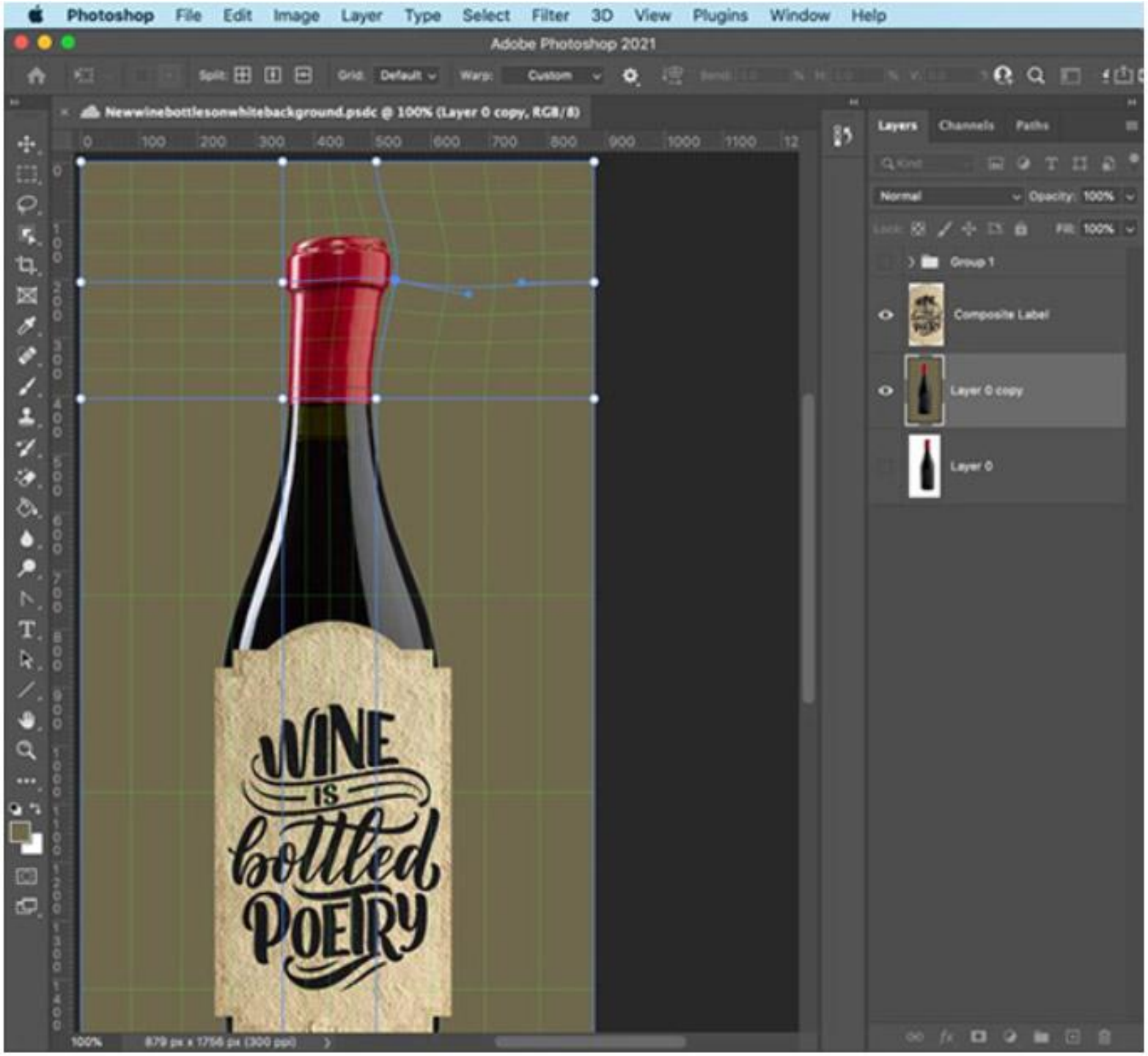




样条曲面：把二维参数空间$(u,v) \in [0,1]\times[0,1]$映射为曲面

$$\mathbf{p}(u,v)=\sum_{i=0}^{n}\sum_{j=0}^{n}B_{i,n}(u)B_{j,n}(v)\mathbf{P}_{ij},\quad u\in[0,1],v\in[0,1]$$


C
D

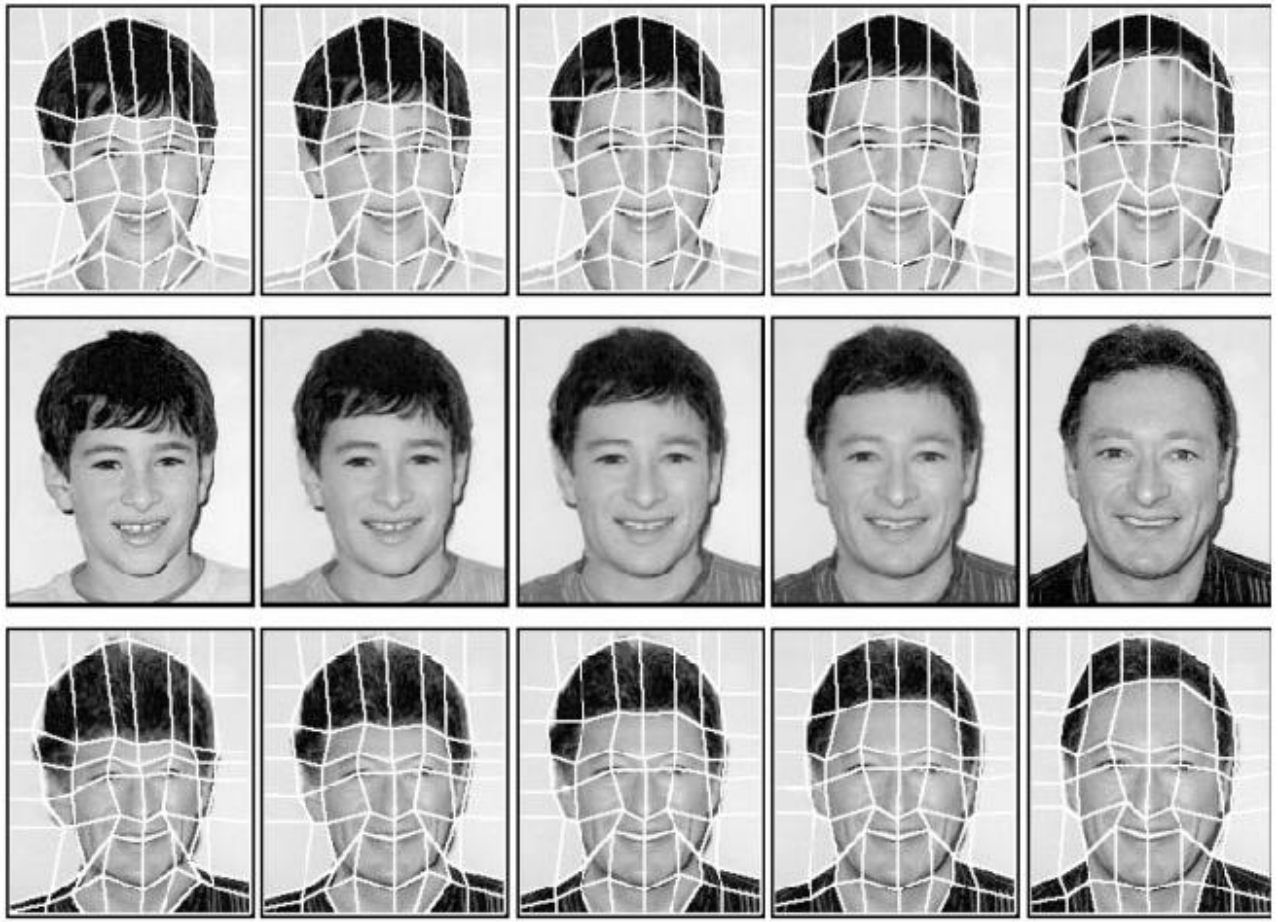


A custom Split Warp has been applied around the neck of the bottle. The Density for the visual Guides has been set to 4; there are four guidelines between each Split Warp.

Indiana Jones and the Last Crusade

基于网格的图象morphing方法

- 该图中排的五帧图象表示两张人脸的morphing过程，其中第一帧和第五帧分别为源图象I_S和目标图象I_D。
- 在源图象中，我们放置曲面网格M_s (上排所示)， M_S指定了控制顶点的坐标。在目标图象中，我们放置网格M_D(下排所示)， M_D指定了M_S在目标图象的对应点。
- 曲面M_S和M_D用来定义把源图象的所有点映射到目标图象的空间变换，M_S和M_D的拓扑关系相同。
- 为了简单起见，网格的边界通常与图象的边界重合，而且冻结不动。



Warp specification

应用时，推荐使用相同的Grid分辨率

Specify corresponding *spline control points, interpolate to a complete warping function*

基于网格的图象morphing方法

- 中间帧图象可以下面的四个步骤来生成：

1. 线性插值网格M_S和M_D，得到网格M。
2. 应用由网格M_S和M定义的变换，使源图象I_s扭曲变形到I_0。

   假设M_S对应的曲面为$P^S(u,v)=\sum\sum P_{ij}^S B_i(u)B_j(v)$，$M$对应的曲面为$P(u,v)=\sum\sum P_{ij}B_i(u)B_j(v)$，则图象$I_0$的计算过程为：对于$I_0$上的每一像素$\tilde{P}$，由$\sum\sum P_{ij}B_i(u)B_j(v)=\tilde{P}$计算出满足该公式的参数$(\tilde{u},\tilde{v})$，则$I_0$上$\tilde{P}$点的颜色值为图象$I_s$上点$P^S(\tilde{u},\tilde{v})$的颜色值(或双线性插值)。
3. 应用由网格M_D和M定义的变换，使目标图象I_D变形到I_1。
4. 对图象I_0和I_1进行线性插值，得到中间帧图象。

基于网格的图象morphing方法

- 给定 $f(u,v)=\sum\sum P_{ij}B_i(u)B_j(v)-\tilde{P}=0$, 如何根据P的值计算(u, v)的值?
- 可用**梯度下降法**
- 梯度下降法是求解无约束最优化问题的一种最常用方法，是一种迭代算法，每一步需要求解目标函数的梯度向量。
- 某一函数沿着某点处的方向导数可以以最快速度到达极值，该方向导数我们定义为该函数的梯度：$\nabla=\frac{df(\theta)}{d\theta}$

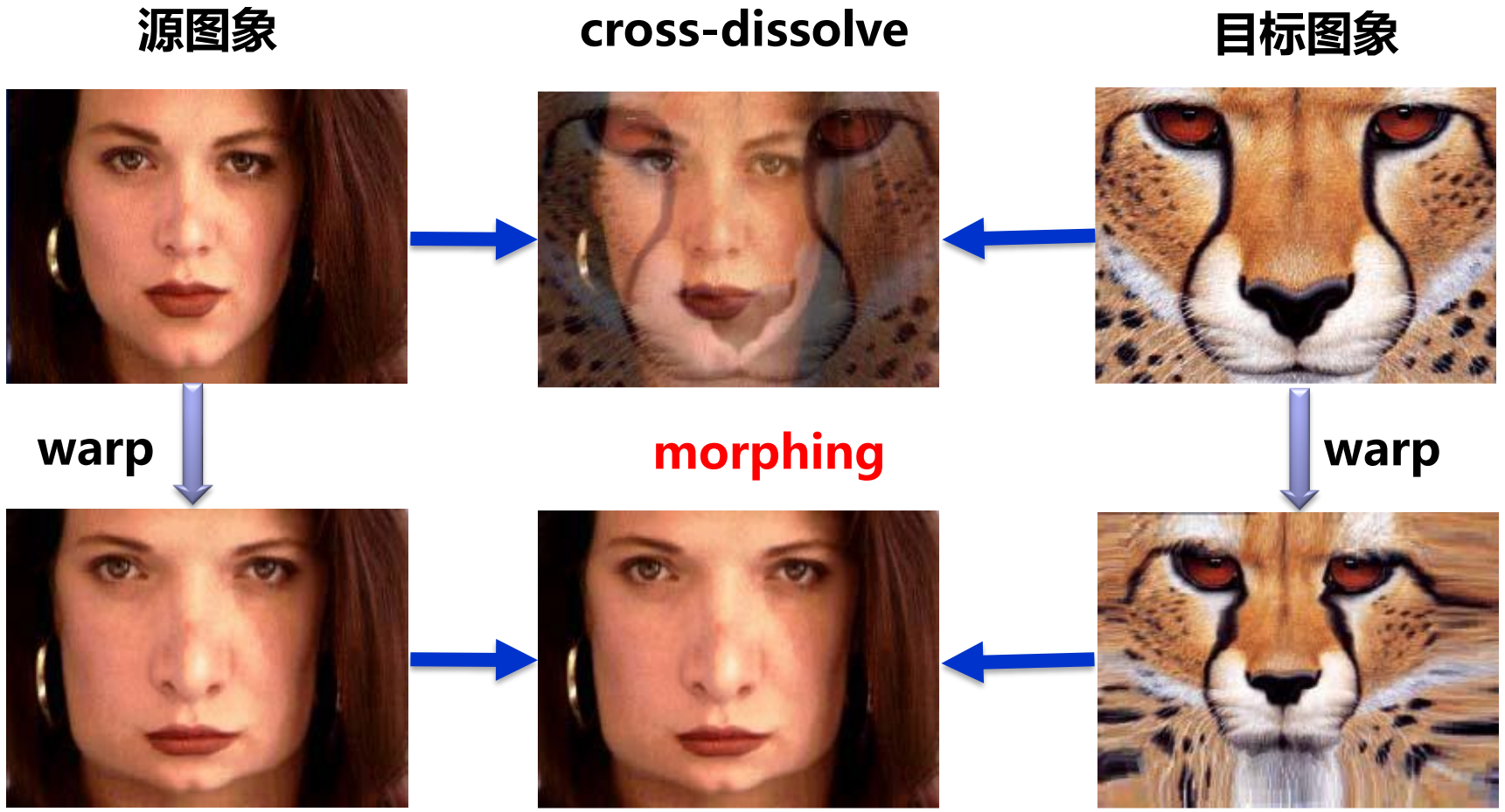
其中θ是自变量，$f(\theta)$是关于θ的函数，∇表示梯度。 所要研究的梯度下降式子可以写为：

$$\theta_{n+1}=\theta_n-\eta\nabla f(\theta_n)$$

其中η是步长，θ_{n+1}是由θ_n 按照上述式子更新后的值。

在这里，初值可设为$(u=0.5, v=0.5)$

基于网格的图象morphing方法



图象morphing中的过渡控制

- 在前面讨论的morphing过程中，我们采用的是均匀过渡，即中间帧图象各部分之间的过渡速度相同。如果允许用一非线性函数来决定图象扭曲和颜色插值的速度，则可以得到许多有戏剧性的视觉效果。

- 在下图中，左上角和右下角的图象分别为源图象和目标图象，定义扭曲变形函数的特征置于图象的上面。上排和下排图象分别显示了以**均匀速度**扭曲源图象和目标图象，即源图象和目标图象上的点均以均匀的速度从初始位置运动到它们的最终位置。对这两排图象进行颜色线性插值，我们得到中排的中间帧图象。可以看出，在颜色插值之前，上排和下排图象的几何形状保持对齐。

均匀速度控制

非均匀过渡函数

- 把非均匀过渡函数应用于相同的源图象和目标图象，我们得到下图所示的效果。
- 在该例子中，定义的过渡函数加速了源图象中鼻子部分的变形速度。
- 人脸的其余部分在前半部分morphing过程中保持不变，从中间的那一帧开始，人脸开始线性变形到最终位置。同样的控制函数应用于下排图象序列。
- 因此，通过采用**非均匀控制函数**，我们可以取得许多奇特的morphing效果。

非均匀控制函数

Face Image Morphing

基于线对的图象morphing方法

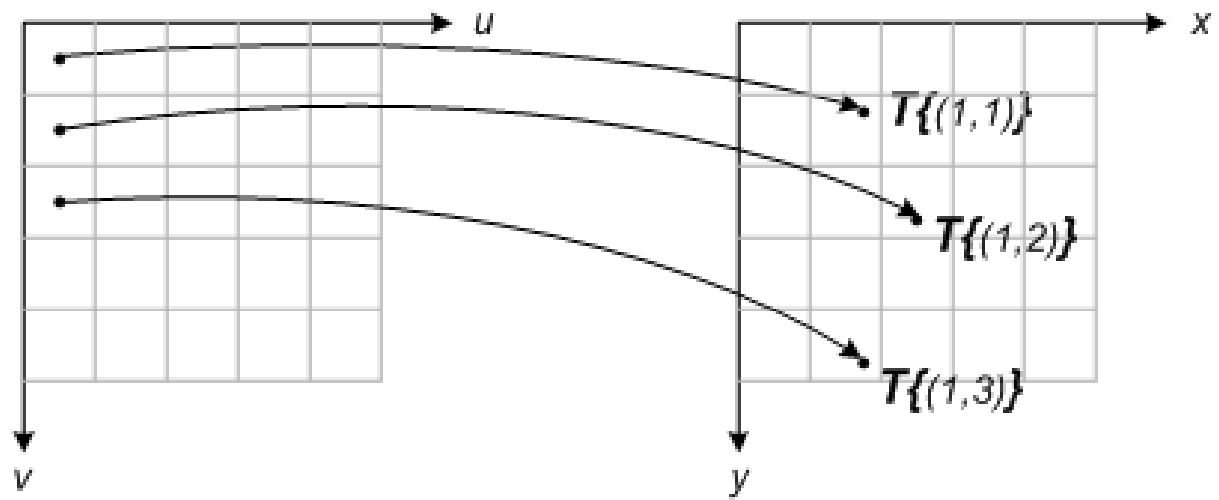
- 基于网格(grid)的特征指定方法使用起来并不很方便，1992年，Pacific Data Images公司的Beier和Neely提出了一种基于线对的特征指定方法(Pacific Data Images (PDI) was an American computer animation and visual effects production company based in Redwood City, California, that was bought by DreamWorks SKG in 2000. It was renamed PDI/DreamWorks and was owned by **DreamWorks** (梦工厂) Animation)

- 该方法允许动画师用**线对**对morphing进行直观的控制，通过交互地指定图象的特征，可以方便地达到动画师预期的视觉效果。

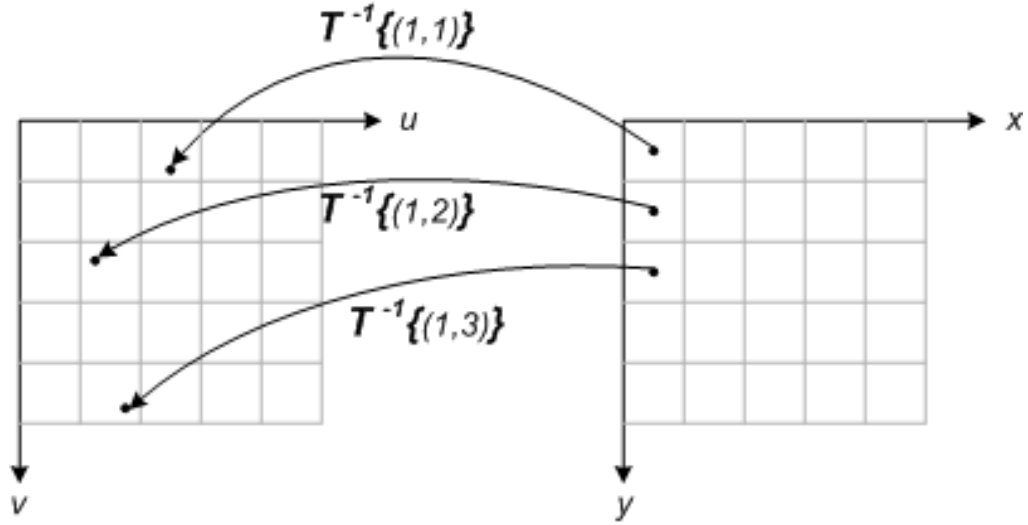
Ref: Beier T, Neely S. Feature-based image metamorphosis. Computer Graphics, 1992, 26(2): 35~42 (2026年3月18日，被引用次数：1863)

单幅图象的变形映射

- 在讨论两幅图象的morphing之前，先考虑单幅图象的warping。
- 单幅图象的warping映射有两种方法：
  - 正向映射。正向映射为逐个扫描源图象素，然后把它拷入目标图象的适当位置。
  - 逆向映射。其思想是逐个扫描目标图象的象素，根据其位置采样源图象。
- 逆向映射有一个很好的优点：目标图象中的每一个象素都可以在源图象中找到对应值。而对于正向映射，目标象素的某些象素可能没有被置上，这些空洞必须通过插值来填上。
- 基于线对的方法采用逆向映射。因而要解决的问题是：对于目标图象的每一个象素，求它在源图象中所要采样的位置。


u
x
T{(1,1)}
T{(1,2)}
T{(1,3)}
v
y
Input space
Output space

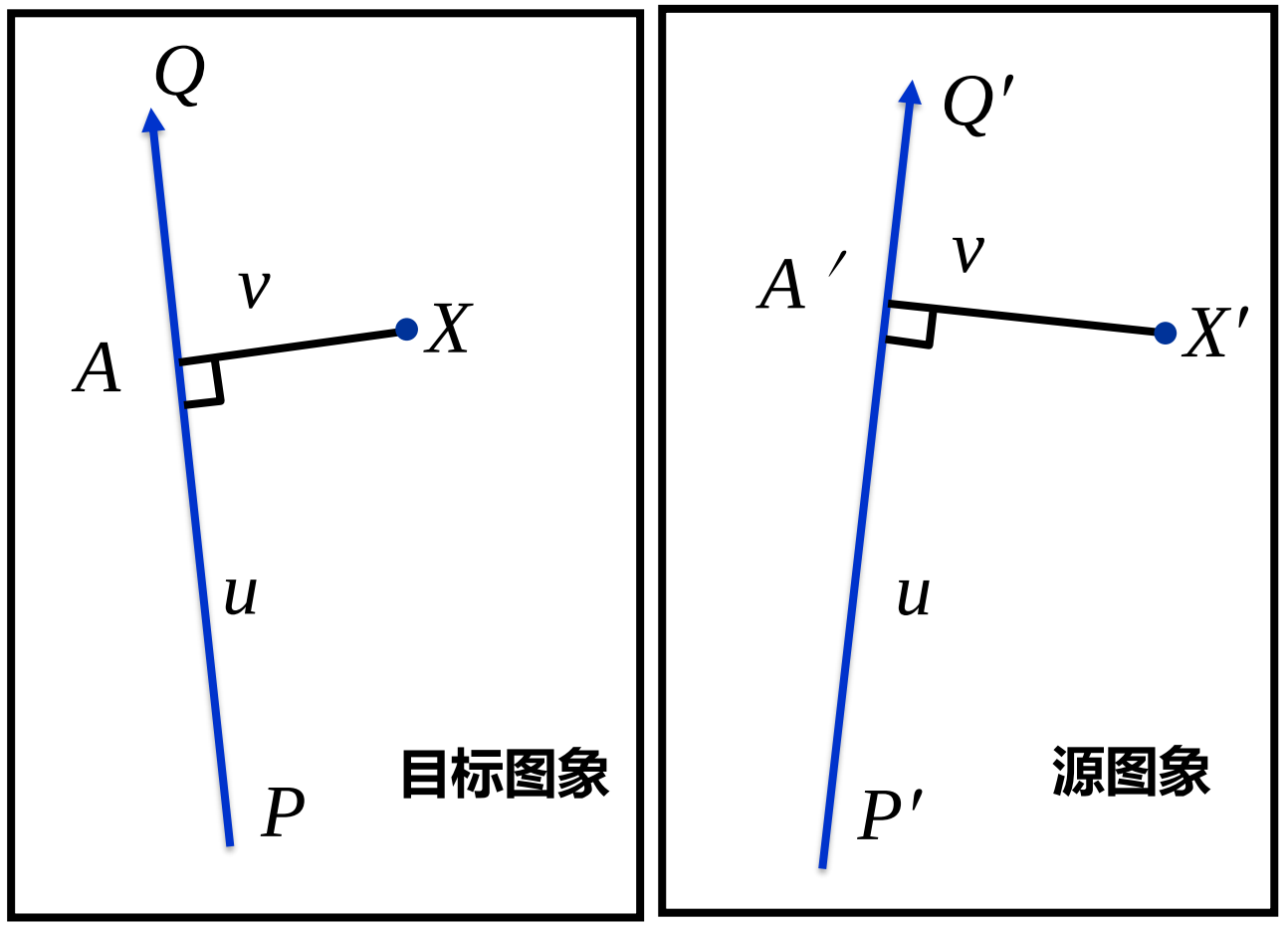
正向映射


T -1{(1,1)}
u
x
T -1{(1,2)}
T -1{(1,3)}
v
y
Input space
Output space

逆向映射

由一对直线确定的变换

- 两幅图象之间的变换可以用一对直线来指定。
- 设I_S为源图象，I_D为目标图象。若先在源图象中定义一条有向直线段，再在目标图象中定义一条有向线段，那么这一对直线段定义了一个从一幅图象到另一幅图象的映射，该映射把一条直线映为另一条直线。
- 设从$I_S \rightarrow I_D$的映射为 f: $X' \rightarrow X$，A为X 在PQ上的投影，u为PA与PQ的比，v为XA的长度，如右图所示。



单对线对

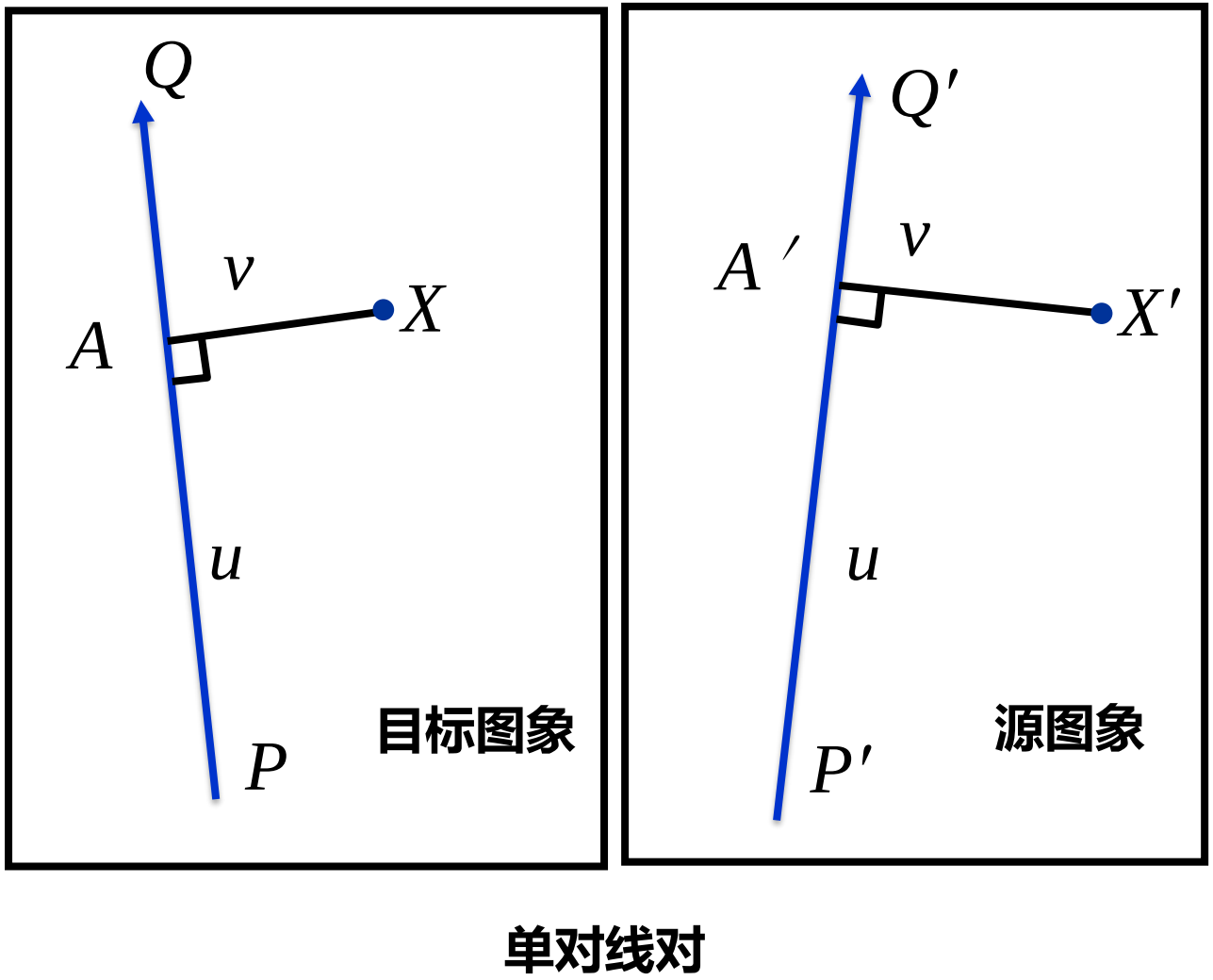
由一对直线确定的变换

$$u = \frac{PA}{\|PQ\|} = (X-P)\bullet\frac{(Q-P)}{\|Q-P\|}\times\frac{1}{\|Q-P\|} = \frac{(X-P)\bullet(Q-P)}{\|Q-P\|^2}$$

$$v = AX = (X-P)\bullet\frac{Perpendicular(Q-P)}{\|Q-P\|}$$

$$X' = P' + u(Q'-P') + v\frac{Perpendicular(Q'-P')}{\|Q'-P'\|}$$

其中函数Perpendicular()返回一长度与输入矢量相等且与输入矢量垂直的矢量。



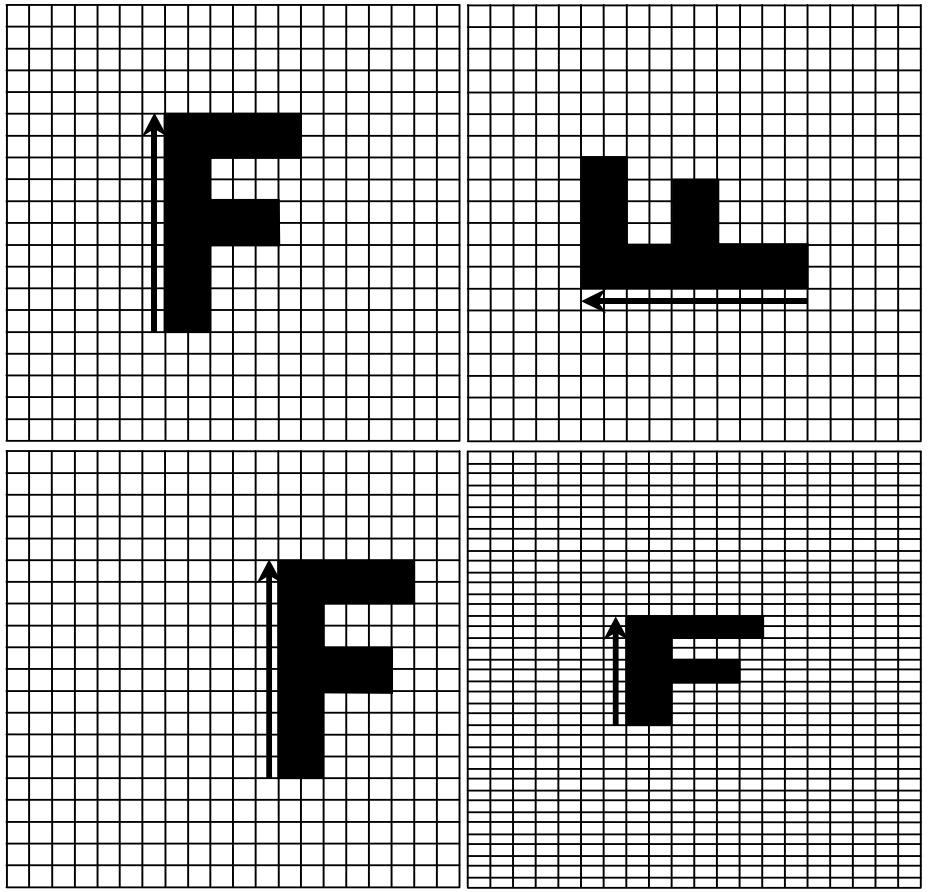
单对线对

由一对直线确定的变换

- 若只有一对直线段，则图象变换的过程可描述如下：对于任一$X \in I_D$，计算u, v, X'的值，然后把源图象在X'处的颜色值赋给目标图象的X象素，即:

$$I_D(X) \leftarrow I_S(X')$$

- 一对直线之间的变换实际上是一个由旋转、平移和比例变换复合成的变换。

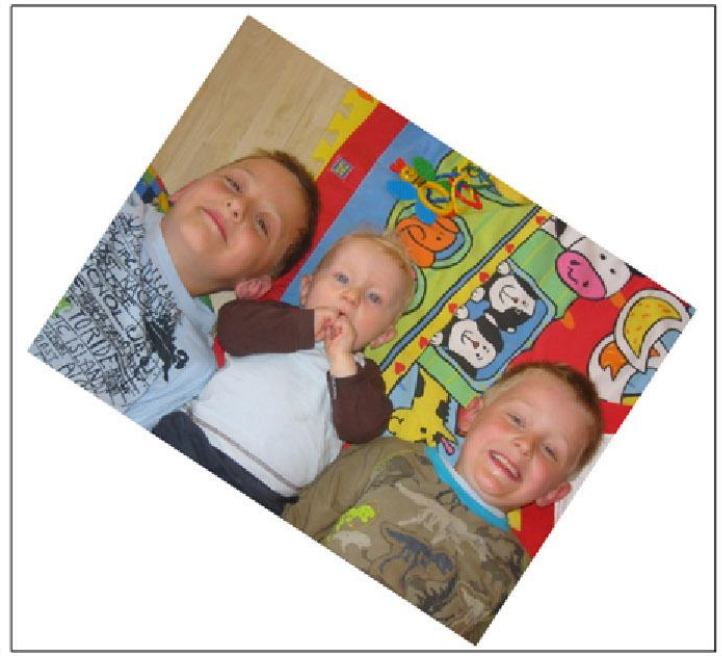


左上角的为原图象，把该图中的特征线旋转、平移和比例缩放后，我们分别得到右上角、左下角和右下角的图象。

(a) Input

(b) Translate

(c) Scale



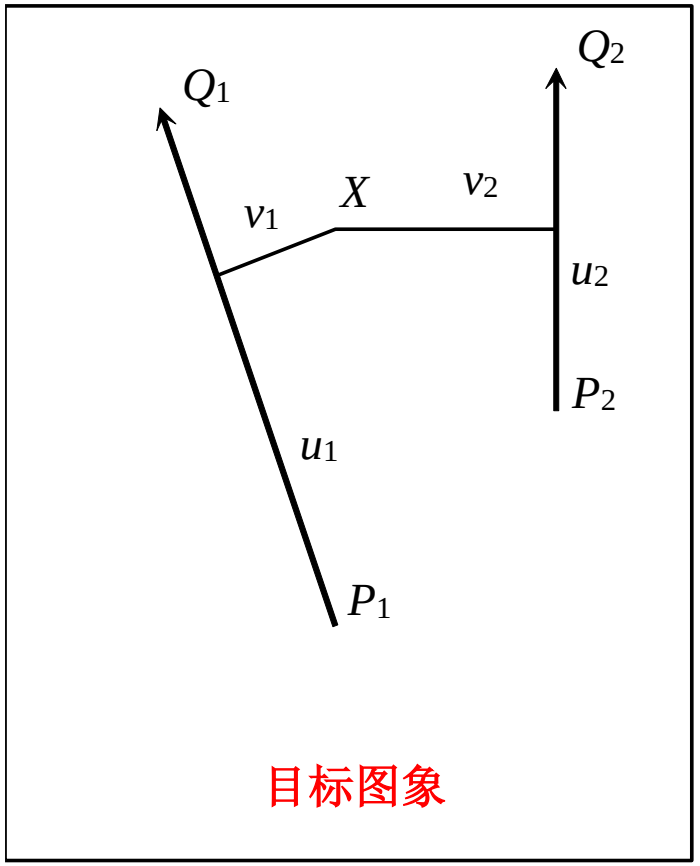
(d) Rotate

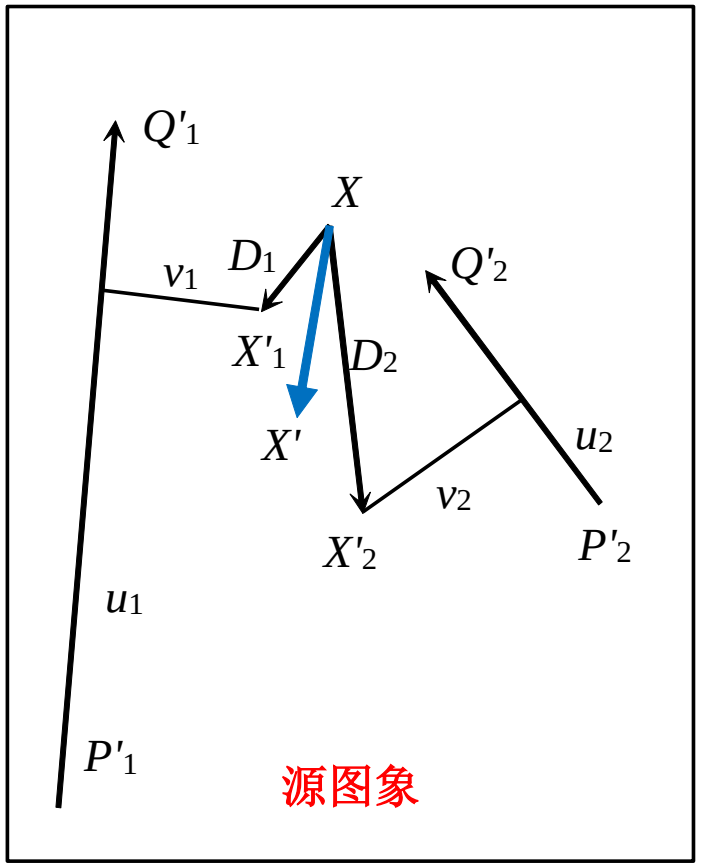
多对直线进行变换

- 由多对直线所指定的变换要比由一对直线所指定的变换复杂得多。
- 设$X \in I_D$，对于每一对直线对i，都有一个与它对应的点X_i'，如下图所示。设$D_i = X_i' - X$，显然D_i为由第i对直线变换所引起的象素位置的偏移量。
- 对所有的偏移量进行加权平均，得：

$$D = \sum_{i=1}^{n} w_i D_i \Big/ \sum_{i=1}^{n} w_i$$

- X'可取为 $X' = X + D$





参数的含义

- 若只有一对直线，则

$$X' = X + D_1 = X + (X_1' - X) = X_1'$$

与前面的结果相符。

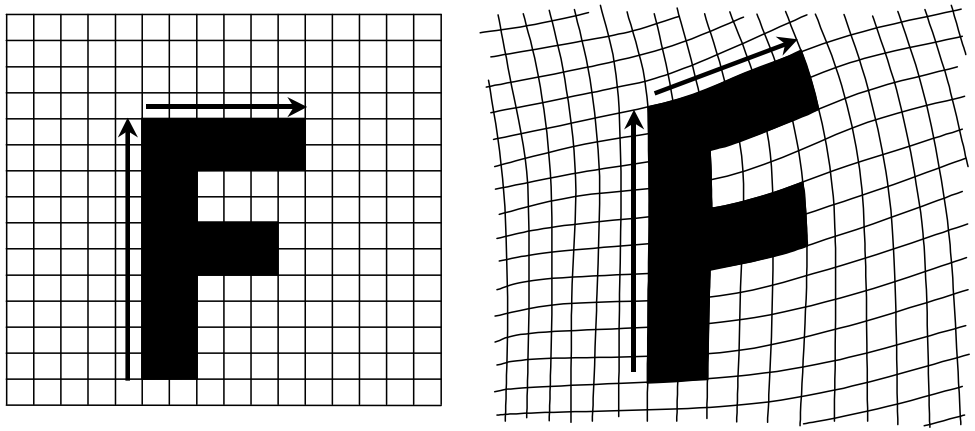
- 权因子取为：

$$w_i = \left[\frac{length_i^{\,p}}{a + dist_i} \right]^b$$

- 其中$length_i$为直线P_iQ_i的长度，$dist_i$为X到直线P_iQ_i的距离，a、b和p为调整线对相对影响效果的参数。
- 当$0<u_i<1$时，$dist_i$的值取为$abs(v_i)$；当$u_i<0$时，$dist_i$的值取为X到P_i的距离；当$u_i>1$时，$dist_i$的值取为X到Q_i的距离。
- 若a接近于0，则当$dist_i \to 0$时，权因子接近∞，直线对上的象素可精确映射。当a取较大值时，映射将变得光滑，但控制精度变差。
- 参数b决定了随着距离$dist_i$的变大不同直线相对强度的减弱程度。若b等于0，则每个象素受所有直线同等的影响。b的取值一般为[0.5, 2]。
- 参数p决定了直线的长度对映射的影响。若p =0，则长短直线的地位等效；若p =1，则长直线比短直线有更大的权因子；p的取值范围一般为[0,1]。

多对直线的变形映射算法

- 采用多对线对时，得到的变换就不再象一对线对那么简单。
- 在下图中，左边的是原图，通过旋转“F”上面的线段，整幅图象变形为右图所示的图象。



for（目标图象 DI 中的每一象素 X）

$Dsum = 0.0$；

$weightsum = 0.0$；

for（每条线段 P_iQ_i）

根据 P_iQ_i 计算 u,v；

根据 u,v 和 $P_i'Q_i'$ 计算 X_i'；

计算由这一线段引起的偏移量 $D_i = X_i' - X$；

$dist$=从 X 到 P_iQ_i 的最短距离；

$weight = (length^p / (a + dist))^b$；

$Dsum += D_i * weight$；

$weightsum += weight$；

end for

$X' = X + Dsum / weightsum$；

$DI(X) = SI(X')$；

end for

两幅图象之间的morphing

- 设这两幅图象为I_S，I_D，先在两幅图象中定义控制变形的对应直线对，通过插值得到中间图象I的直线。
- 插值方法有两种：
  - 对直线端点进行线性插值。该方法的缺点是对于旋转的直线对可能得到缩短的插值直线。
  - 对直线的中点、朝向和长度进行插值，通常这一方法效果较好。
- 由$I_S \rightarrow I$的直线对变换可得到一幅变形图I_1；由$I_D \rightarrow I$的直线对变换可得到I_2。对交溶参数作动画，把I_1和I_2进行交溶处理得到中间的变形图象I。

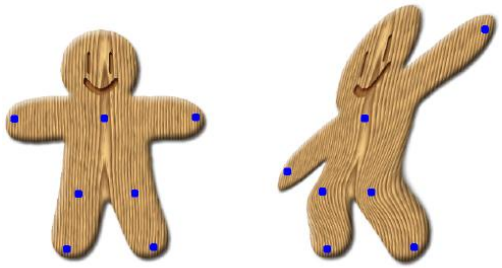
- 基于线对特征法的优点是直观，缺点是有可能生成一些意料之外的图象。

Michael Jackson’ s MTV “Black OR white”

一些其它工作

- 点对 ⟶ 散乱数据插值 (Radial Basis Function)

  Ruprecht, Detlef, and Heinrich Müller. "Image warping with scattered data interpolation." *IEEE Computer Graphics and Applications* 15.2 (1995): 37-43.



- Moving least squares ⟶ Affine, similarity and rigid transformations

  Schaefer, Scott, Travis McPhail, and Joe Warren. "Image deformation using moving least squares." *ACM Transactions on Graphics (TOG)*. Vol. 25. No. 3. ACM, 2006.

Beier'1992

Rigid MLS

视域morphing

- 通过插值给定图象的几何和颜色，图象morphing取得了令人惊呀的视觉效果。
- 但图象morphing不能保证得到的结果是自然的，部分原因在于图象morphing没有考虑视点的变化。
- 解决办法：视域morphing!

Ref: Seitz S M, Dyer C R. View Morphing. Computer Graphics, 1996, 30(3):21~30 (2026年3月18日，被引用次数：**1377**)

同时插值几何、颜色和视点!

不考虑视点变化可能出现的严重畸变现象

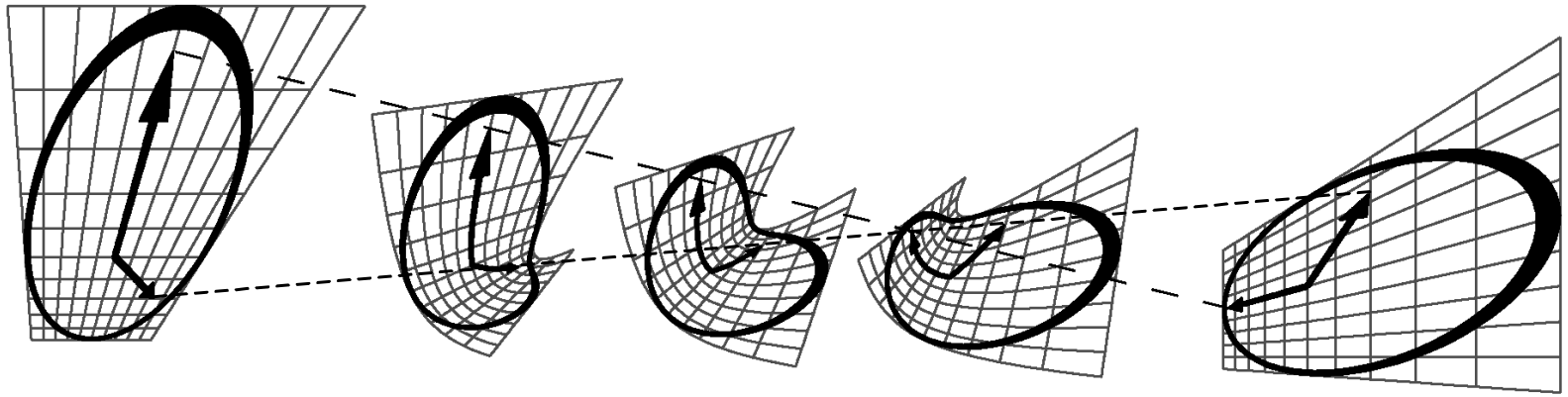
- 考虑一个平面形状的两个不同视域的插值，任何这样的两幅图象通过投影映射$H(x, y)$相关联:

$$H(x,y)=\left(\frac{ax+by+c}{gx+hy+i},\frac{dx+ey+f}{gx+hy+i}\right)$$

- 因为两个上述表达式的和通常是一有理二次式，不再是一投影映射，所以，对投影映射进行线性插值不能保证其结果仍然是投影映射。
- 因此，一般的图象morphing不能保证中间帧是同一物体在新视点的投影图象，而是一种会带来形状畸变的变换。

不考虑视点变化可能出现的严重畸变现象

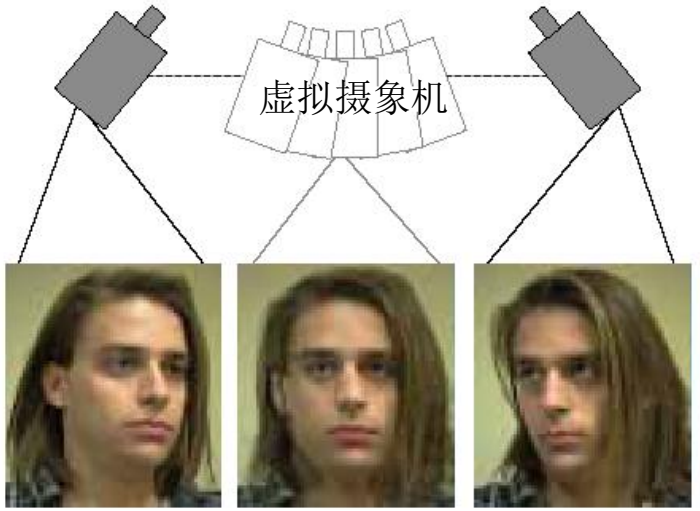
- 一个恼人的现象是**直线**在morphing后**变成曲线**，导致不自然的图象过渡。
- 下图是一个时钟在两个不同透视视域的morphing过程，中间形状通过线性插值时钟在不同视域的同一点得到，可以发现时钟在中间帧扭曲变形。



形状扭曲的morphing过程，虚线为同一特征的线性路径

视域morphing

- 我们自然会问，给定某一物体在不同视点的投影图象，能否找出一种与以前不同的图象morphing方法，使得能够生成该物体在新视点的投影图象？
- 1996年，Seitz等提出了一种称为视域morphing的新方法，通过采用现有的图象morphing技术为中间步骤，该方法能模拟给定图象在相应三维空间的视点变化。
- 与图象morphing相比，该方法同时插值几何、颜色和摄像机，因而能产生类似三维的视觉效果。



源图象　视域morphing　目标图象

对从两个不同视点摄取的同一物体的图象进行视域morphing可以产生移动虚拟摄象机的效果

计算视域morphing需要的信息

- 同一三维物体或场景在两个不同视点的投影图象I_0和I_1
- 两个视点的投影矩阵$\prod_0$和$\prod_1$
  - 在视域morphing中，我们并不需要预先知道物体的三维几何信息，但需要投影矩阵的信息。投影矩阵可以由多种途径获得，例如，由预先知道的一些图象点的三维位置信息计算求得(摄像机定标)。
- 两幅图象象素之间的对应关系
  - 象素之间的对应关系则由图象morphing技术来获得。如果象素的对应关系正确，则得到的morphing图象透视关系正确。
  - 实验发现，近似的象素对应关系也能生成视觉上令人信服的视域morphing效果。

一些定义和表示

- 我们采用欧氏空间的齐次坐标来表示点的坐标。
- 设 $\mathbf{Q}=(X,Y,Z,1)^T$ 表示场景上的点， $\mathbf{P}=(x,y,1)^T$ 表示图象上的点，它们与标量的乘积分别用$\tilde{\mathbf{Q}}$和$\tilde{\mathbf{P}}$来表示。很显然，齐次坐标 $\tilde{\mathbf{Q}}$ 和 $\mathbf{Q}$、$\tilde{\mathbf{P}}$ 和 $\mathbf{P}$表示的是相同的点。
- 摄象机表示为3×4的齐次投影矩阵的形式： $\prod=[\mathbf{H}\,|\,-\mathbf{HC}]$

其中C表示摄象机中心的位置， 3×3矩阵H指定了象平面在世界坐标系中的位置和方向。透视投影的方程为：

$$\tilde{\mathbf{P}}=\prod\mathbf{Q}$$

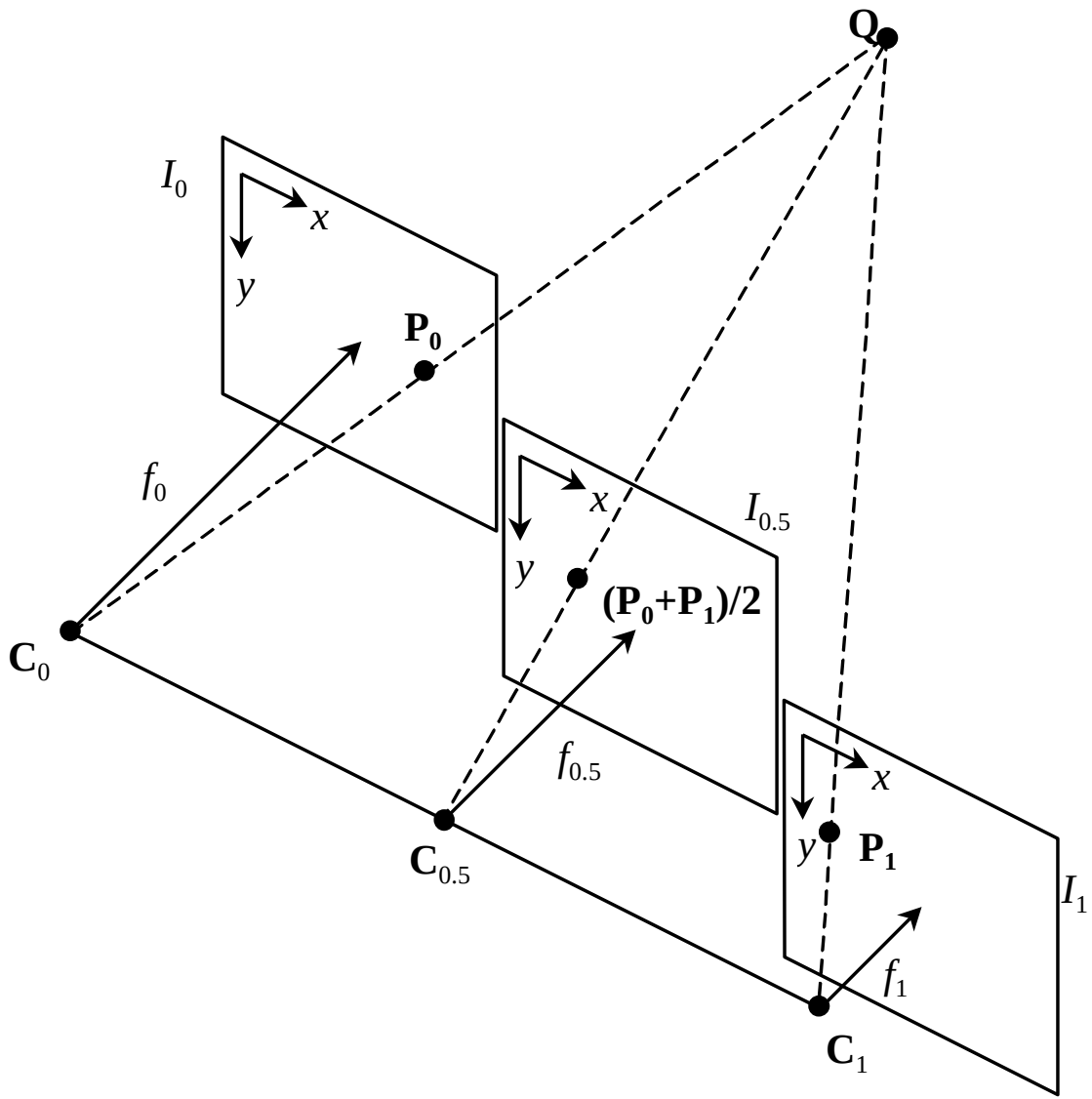
- 在后面的讨论中，我们用二元组<I, $\prod$ >表示“视域” 。视域包括一幅图象以及和它相关的投影矩阵。

平行摄象机的情形

- 先讨论平行摄象机的情形。
- 设摄象机从世界坐标系的原点(0,0,0)移到$(C_X, C_Y, 0)$，焦距f_0从变到f_1，投影图象从I_0变到I_1，如下图所示，则它们对应的投影矩阵分别为：

$$\Pi_0 = \begin{bmatrix} f_0 & 0 & 0 & 0 \\ 0 & f_0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad \Pi_1 = \begin{bmatrix} f_1 & 0 & 0 & -f_1 C_X \\ 0 & f_1 & 0 & -f_1 C_Y \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

- 分别称这样的摄象机和视图为平行摄象机和平行视图。



平行视域的morphing。把象平面I_0和I_1上的相应象素进行线性插值，我们得到了同一场景在另一平行视域的投影图象$I_{0.5}$。

平行摄象机的情形

- 设景物点 $\mathbf{Q}=(X,Y,Z,1)^T$ 在象平面上的投影分别为$\mathbf{P}_0$和$\mathbf{P}_1$。对$\mathbf{P}_0$和$\mathbf{P}_1$进行线性插值，得到

$$(1-t)\mathbf{P}_0+t\mathbf{P}_1=(1-t)\frac{1}{Z}\prod_0\mathbf{Q}+t\frac{1}{Z}\prod_1\mathbf{Q}=\frac{1}{Z}\prod_t\mathbf{Q}$$

其中

$$\prod_t=(1-t)\prod_0+t\prod_1$$

- 从上式可以看出，图象插值生成了一个投影矩阵为$\prod_t$的新的视域。 $\prod_t$为$\prod_0$和$\prod_1$的线性插值，它表示了一个中心为$\mathbf{C}_t$、焦距为f_t的新的摄象机位置，其中:

$$\mathbf{C}_t=(tC_X,tC_Y,0),\quad f_t=(1-t)f_0+tf_1$$

平行摄象机的情形

- 结论：对于平行摄象机，插值图象与使摄象机的中心沿直线$\mathbf{C}_0\mathbf{C}_1$移动并同时连续推拉镜头(Zoom)的效果是相同的。

- 由于上述图象插值方法得到了同一物体在新视点的投影图象，因而透视关系正确。

摄象机不平行的情形

- 从透视变换的知识可知，如果两个视域共享相同的摄象机中心，则这两个视域可通过一平面投影矩阵相关联。
- 设I和I’分别为投影矩阵 $\prod = [\mathbf{H} \mid -\mathbf{HC}]$ 和 $\prod' = [\mathbf{H}' \mid -\mathbf{H}'\mathbf{C}]$ 的投影图象。对于场景中的点Q,设它在I中的投影为**P**，在I’中的投影为**P**’，则有：

$$\tilde{\mathbf{P}}' = \prod' \mathbf{Q} = \mathbf{H}'\mathbf{H}^{-1} \prod \mathbf{Q} = \mathbf{H}'\mathbf{H}^{-1}\tilde{\mathbf{P}}$$

其中 $\mathbf{H}'\mathbf{H}^{-1}$ 为一 3×3的投影矩阵，它把图象平面I重投影为I’ 。

重投影

- 更一般地，任何可逆的3×3矩阵表示一平面投影矩阵，它把点一一映射为点，把线一一映射为线。

- 重投影是一种非常有用的操作，它允许照片拍好或者场景绘制好以后改变视线的方向。

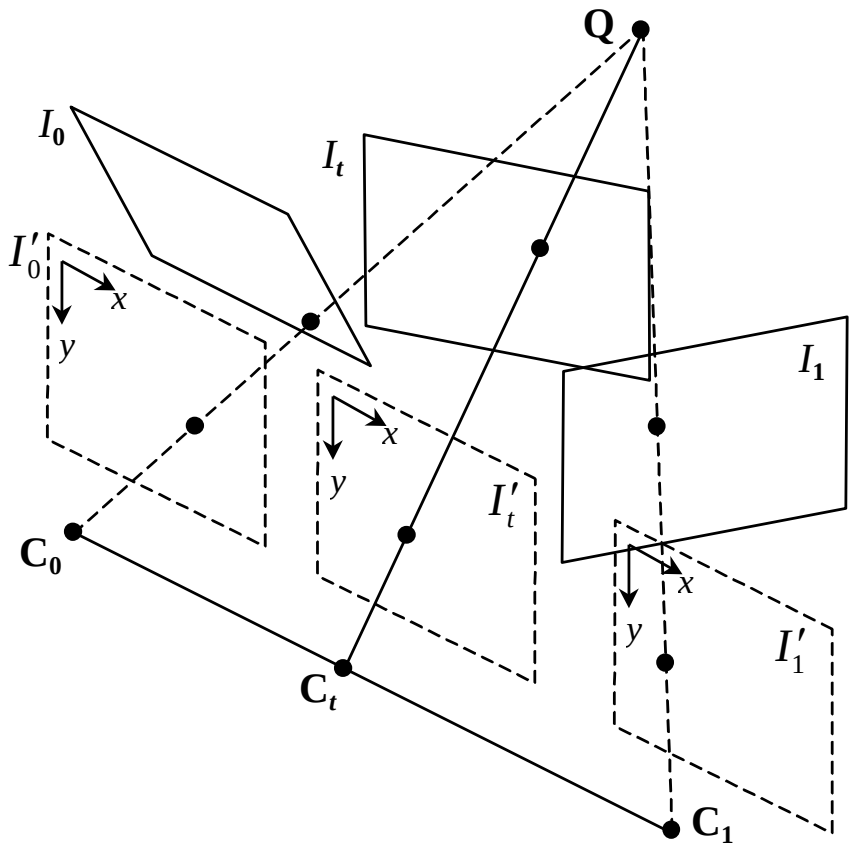
- 重投影即可以用扫描线方法，也可以用纹理映射方法来实现。

非平行透视视域→平行透视视域

- 有了重投影，计算能保持透视关系的非平行透视视域的morphing问题可以转化为平行透视视域的morphing问题。
- 设I_0和I_1分别为投影矩阵 $\prod_0 = [\mathbf{H}_0 \mid -\mathbf{H}_0\mathbf{C}_0]$ 和 $\prod_1 = [\mathbf{H}_1 \mid -\mathbf{H}_1\mathbf{C}_1]$ 的投影图象。
- 为了方便起见，我们选择一种特殊的世界坐标系，使得$\mathbf{C}_0$和$\mathbf{C}_1$位于世界坐标系的x轴上，不妨设 $\mathbf{C}_0 = [X_0, 0, 0]^T$，$\mathbf{C}_1 = [X_1, 0, 0]^T$
- 其余两个坐标轴的选取应以尽量减少图象重投影带来的失真为原则，一种简单可行的方案是取y轴为两个图象平面法向的叉积。

视域morphing过程图示

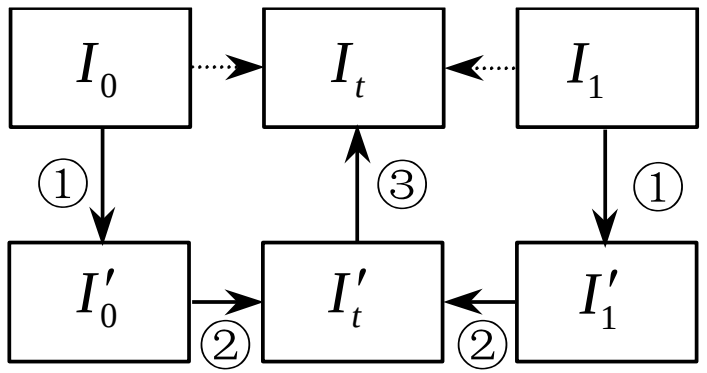
- 在直线上的透视图象可以看成是图象重投影和图象插值两个过程的组合，如下图所示。



视域morphing的三个步骤

- 给定中间帧的投影矩阵 $\Pi_t = [\mathbf{H}_t \mid -\mathbf{H}_t\mathbf{C}_t]$，对应于投影矩阵$\prod_t$的投影图象$I_t$可用下面的三步算法来生成：
  1. 前置变形(Prewarp)。对图象I_0应用投影变换$\mathbf{H}_0^{-1}$，对图象I_1应用投影变换$\mathbf{H}_1^{-1}$，生成前置变形后的图象I_0’和I_1’。
  2. Morphing。采用平行摄象机的方法，线性插值图象I_0’和I_1’相应点的位置和颜色，得到图象I_t’。
  3. 后置变形(Postwarp)。对图象I_t’应用变换$\mathbf{H}_t$，生成图象I_t。

三步变换序列



视域morphing的三个步骤

- 前置变形在不改变摄象机中心的情况下，使图象I_0’和I_1’对应的象平面平行；
- Morphing过程使摄象机的中心移至$\mathbf{C}_t$；
- 后置变形使象平面变换至新的位置和方向。

- 可以发现，前置变形后的图象I_0’和I_1’对应的投影矩阵分别为$\prod_0$’=[**I** | -$\mathbf{C}_0$]和$\prod_1$’=[**I** | -$\mathbf{C}_1$]，其中I表示3×3的单位矩阵。这种特殊的投影矩阵使得两图象上的对应点位于同一条扫描线上，因此，图象I_t’的计算可以以逐条扫描线的次序进行，其中只需进行单条扫描线的重采样。

模糊现象的处理

- 在前置变形、后置变形和morphing过程中，需要对图象进行多次重采样，这有可能会使中间帧图象引起较明显的模糊现象：
  - 为了减轻这种现象，我们可以对输入图象进行过采样；
  - 或把对每幅图象进行的两次变形变换合并为一次。

- 后一种方法与现有的采用逆向映射的morphing技术相一致，其缺陷是丧失了扫描线方法的优点。

折叠(fold)和空洞(hole)现象

- 对于给定的两幅参考图象，前面的讨论假设一幅图象上的每一点在另一幅都可见。实际上，当视点改变时，投影图象上的点经常会发生可见性变化，从而出现折叠(fold)和空洞(hole)现象。
  - 当图象I_0（或I_1）中的一可见面在I_t中被遮挡时，将出现折叠现象。此时，I_0中的多个象素映射到I_t中的同一点，引起二义性。
  - 当图象I_0（或I_1）中的一个被遮挡的面突然变成可见时，将出现空洞现象。此时，I_t中的部分区域在I_0中没有对应点。

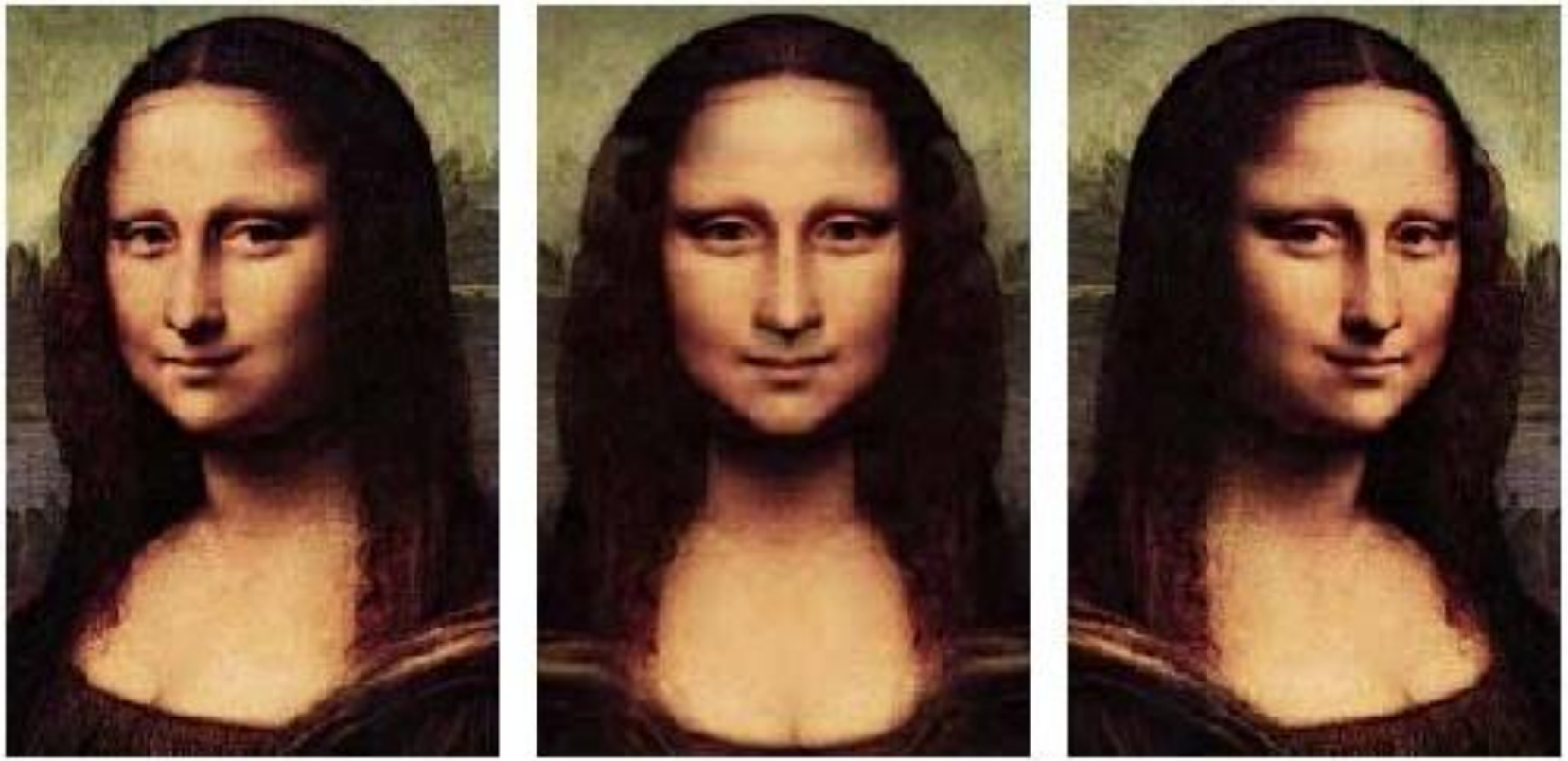
折叠(fold)和空洞(hole)现象的解决方法

- 如果深度信息已知，折叠现象可用Z缓冲消隐算法来消除。
- 如果没有三维形状信息，可以采用一种称为“disparity”（视差）的方法。
- 对于平行的视图，对应点$\mathbf{P}_0$和$\mathbf{P}_1$的disparity定义为它们x轴坐标的差。
- 由投影矩阵可知，在平行视图中，一个点的disparity与该点的深度成反比，故若以1/ disparity代替深度值，我们可直接应用Z缓冲算法。
- 由于在视域morphing中，前置变形使图象对应的象平面平行，所以“disparity”方法非常有效。
- 与折叠不同，空洞不能通过只用图象的信息来消除。常用的空洞添充方法包括背景颜色法、相邻象素插值法等，其中相邻象素插值法是一种流行的方法。

视域morphing应用于人脸的一些结果

这两个例子均有逼真的三维旋转效果。

蒙娜丽莎图的视域morphing



DEMO


View Morphing
Steven M. Seitz and Charles R. Dyer
Department of Computer Sciences
University of Wisconsin --- Madison
http://www.cs.wisc.edu/~dyer/vision.html

总结

- 图象morphing是一种达到视觉特效的有效方法。
- 尽管只在二维图像空间处理问题，但可以让人产生神奇的三维形状改变的错觉。可以避免复杂的三维造型过程。

- **局限性：**缺乏三维几何信息，无法象其它三维物体一样进行几何变换、改变材质属性、改变光照、阴影等，摄象机的动画会受到很大的限制。

Morphing——
The defining visual effect of the late 80s early 90s!

11minutes

The End