基于StyleGan的肖像Morphing和二维多边形形状渐变

金小刚

Email: jin@cad.zju.edu.cn

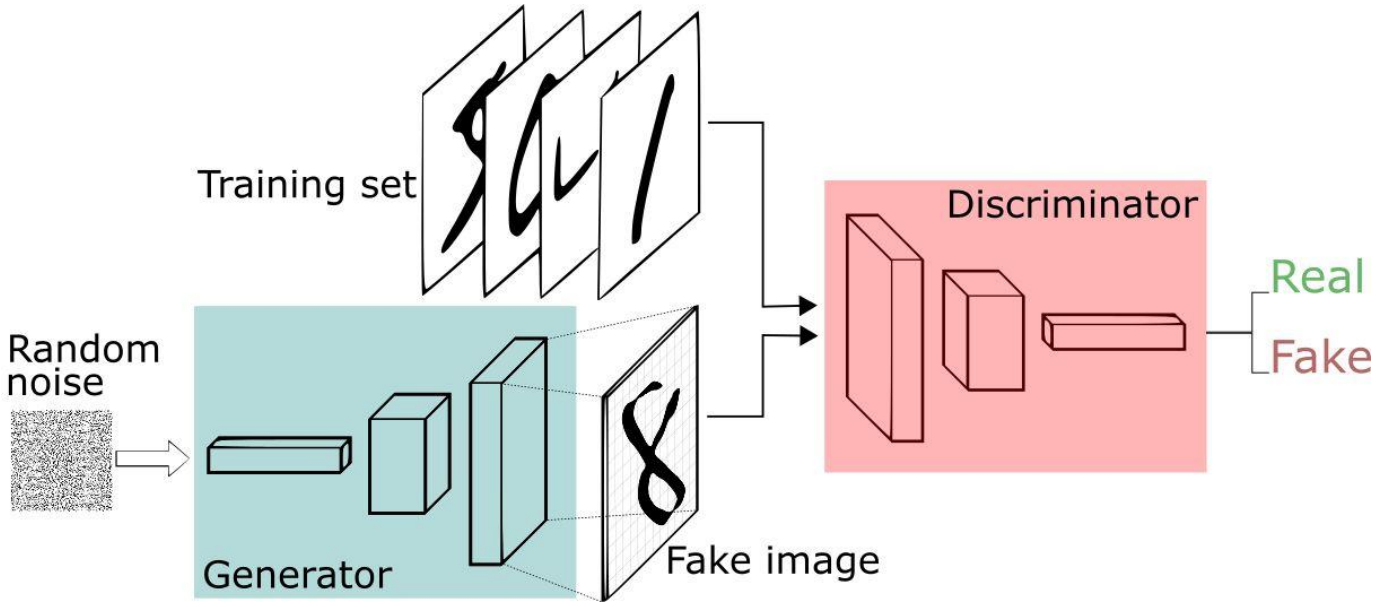
浙江大学CAD&CG国家重点实验室

紫金港校区蒙民伟楼512

@Copyright Xiaogang Jin

基于StyleGan的肖像Morphing

- **2014年，Goodfellow提出了GAN(Generative Adversarial Network)**



- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. Generative adversarial networks. *Communications of the ACM*, 2020, *63*(11), pp.139-144. (The original version of this paper is entitled “Generative Adversarial Networks” and was published in *Advances in Neural Information Processing Systems 27* (NIPS 2014).) (截止2025年3月10日，google scholar引用**78,950**次)

Generative Adversarial Networks

- 2019 年 3 月，ACM 宣布将图灵奖授予深度学习三巨头 Geoffrey Hinton、Yoshua Bengio 和 **Yann LeCun**，以表彰三人「在深度神经网络概念和工程上的突破」。
- **Yann LeCun**评价GAN为 “adversarial training is the coolest thing since sliced bread”。

2014

2015

2016

2017

An illustration of progress in GAN capabilities over the course of approximately three years following the introduction of GANs.

Generative Adversarial Networks

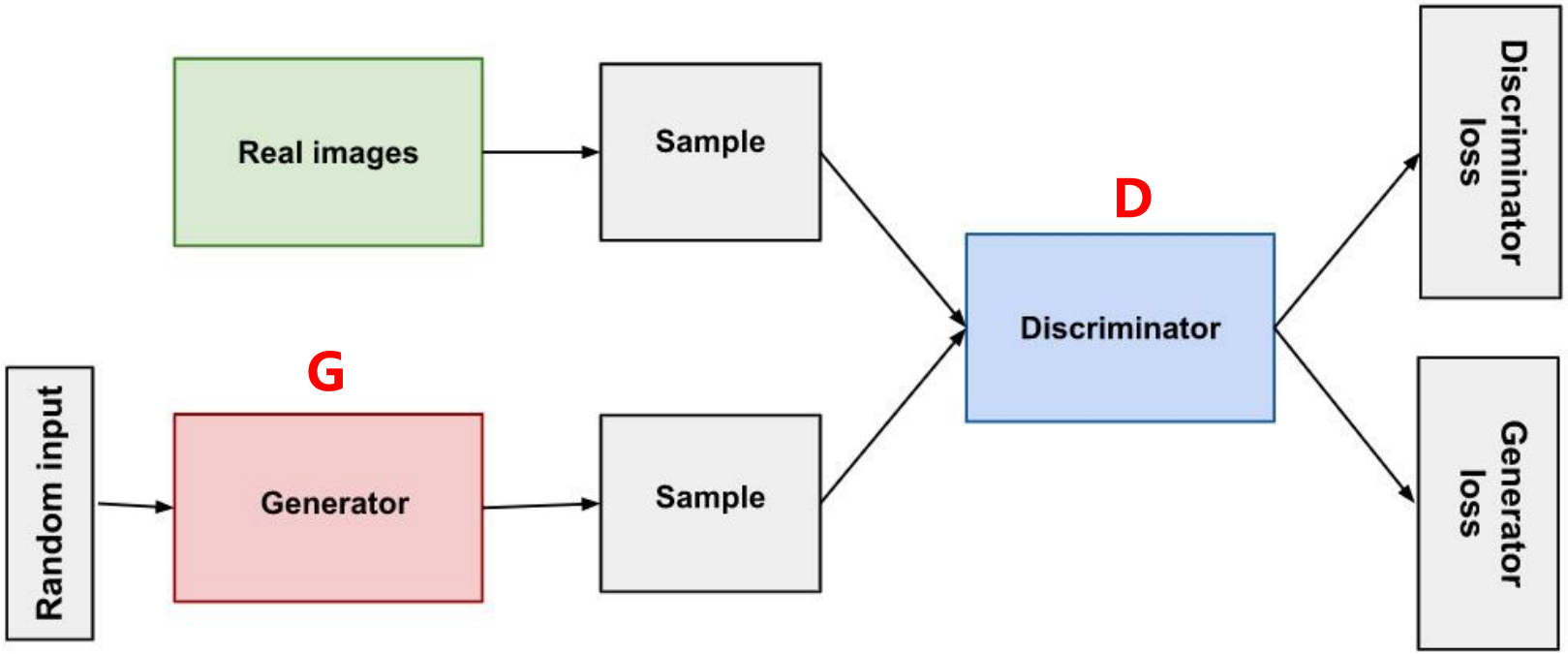
- 以下是使用 GANs在MNIST 和 CELEBA-HQ (https://github.com/tkarras/progressive_growing_of_gans) 两个数据集上训练后获得的结果示例 。

GAN基本思想

- GAN全称是**生成对抗网络**，顾名思义是一种生成模型。它是一种无监督学习方法，通过让**两个神经网络相互博弈**的方式进行学习。
- 生成式对抗网络由**生成器(Generator, G)**和**判别器(Discriminator, D)**构成。生成对抗网络的核心目的是训练生成器。生成器的目的是生成与真实样本尽可能相似的“假样本”，判别器的目的是尽可能区分出给定样本是真实样本还是生成的“假样本”。二者目的相悖，在不断博弈的过程中相互提高，最终在判别器判别能力足够可靠的前提下仍无法区分给定样本是真实样本还是生成样本，从而我们说生成器能够生成**“以假乱真”**的样本。

GAN基本思想

- 生成对抗网络 (GAN) 包含两个部分：
  - ✓ **生成器G**学习生成合理的数据。生成的实例会成为判别器的负性训练示例。
  - ✓ **判别器D**会学习区分生成器的虚假数据与真实数据。判别器通过生成可观测的结果来惩罚生成器。



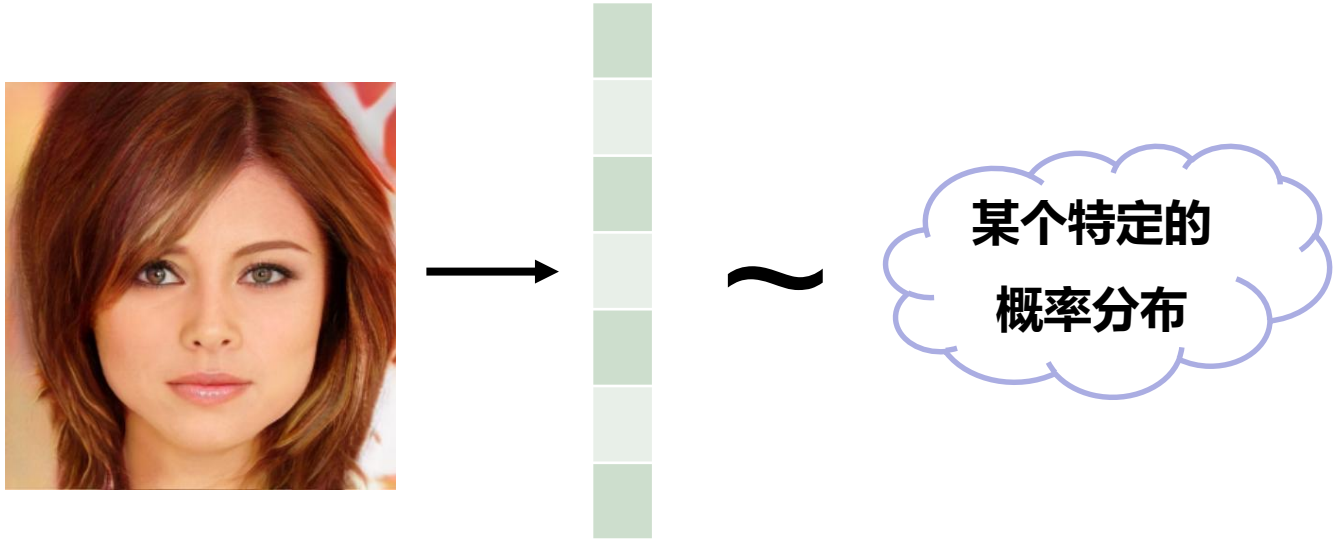
GAN 结构概览

- 在开始训练时，生成器会生成明显的虚假数据，而判别器会快速学会辨别数据是虚假的：

- 随着训练的进行，生成器越来越接近能够分辨判别器的输出：

Generative Adversarial Networks for Faces

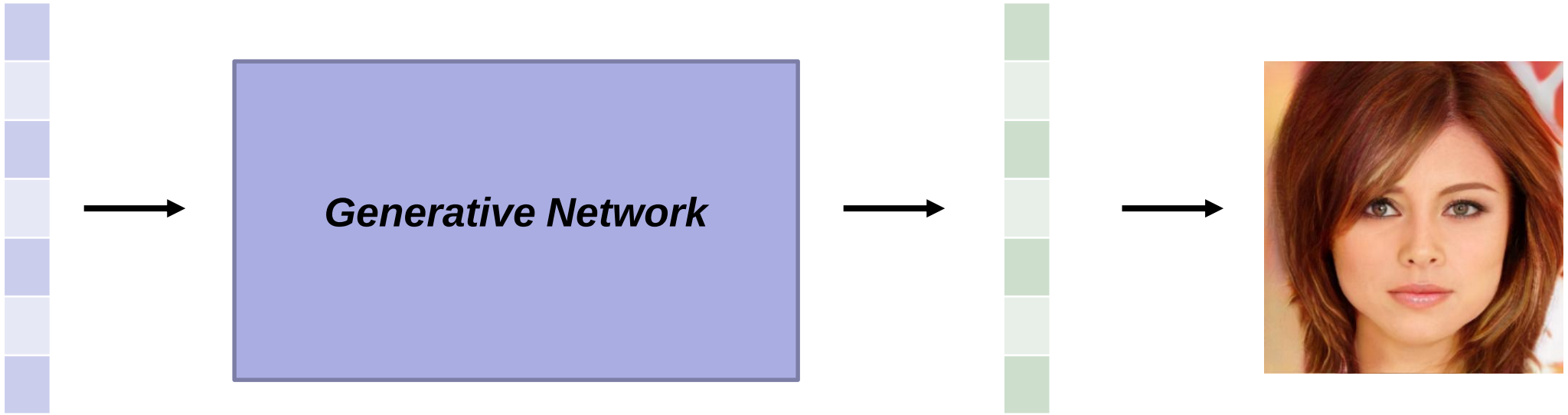
- **目标：**针对特定的某个复杂概率分布，生成随机变量。
- 例如，我们希望能够生成分辨率为$n \times n$的人脸图像，每张图像可以视为一个$N = n \times n \times 3$维向量（r、g、b三个分量）。
- 但是，不是每个N维随机向量都能表示一张人脸。能表示合理人脸的N维向量符合 N **维向量空间上某个特定的概率分布**。



Generative Adversarial Networks

- **How to generate complex random variables?**
- 向量空间上的“人脸概率分布”非常复杂，无法进行显式地描述。
- 可以将符合人脸概率分布的复杂随机变量表示为一个**复杂的函数**应用于**简单的随机变量**的结果。
  - **complex random face variables = function(simple random variables)**
- 通过一个神经网络对这个复杂的函数进行建模，神经网络将一个简单的随机变量作为输入，并输出遵循“人脸概率分布”的随机变量。这个简单的**随机变量**被称为**隐码(latent code)**，隐码所在的空间称为**隐空间(latent space)**。

Generative Adversarial Networks

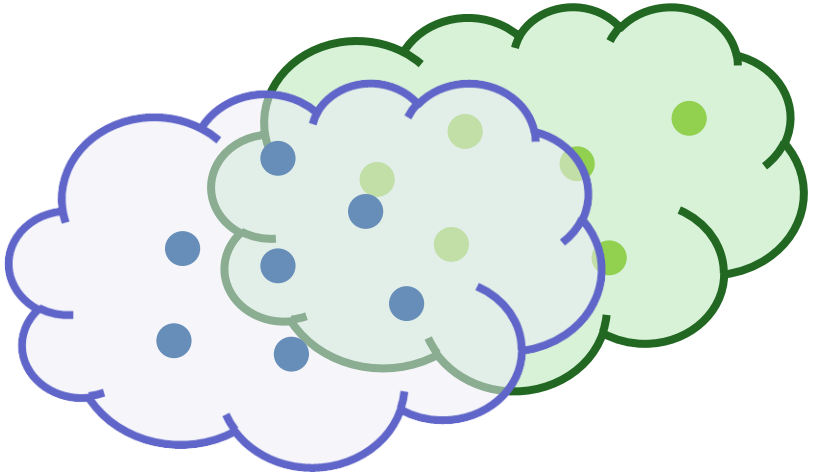


从某个非常简单的分布中采样的随机变量

符合人脸分布的随机变量

Generative Adversarial Networks

- **How to train the Generative Network?**
- 将**生成的分布与目标分布进行比较**，来训练生成网络。
- 但是，我们不知道如何显式地表达真正世界中的“人脸概率分布”，因为这个分布过于复杂。

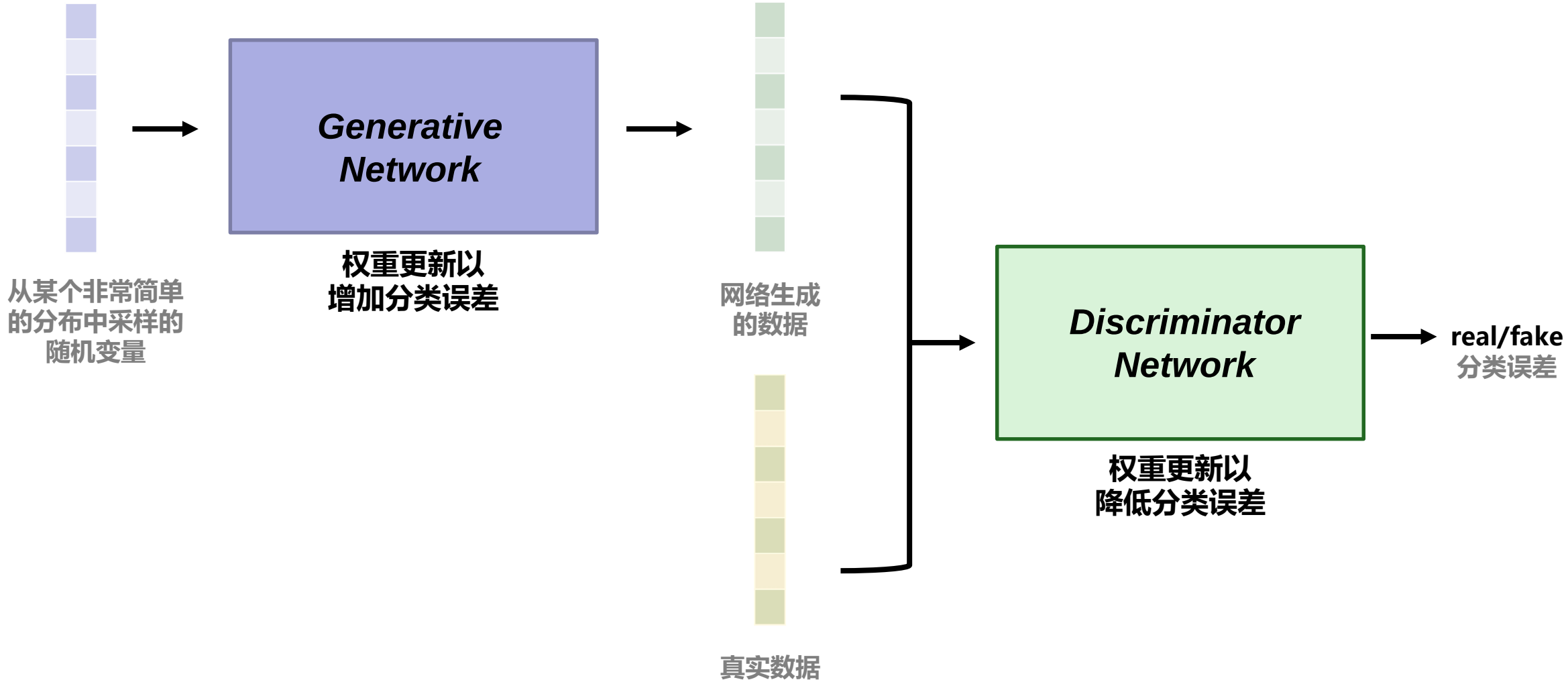


怎么比较生成的概率分布和真实的“人脸概率分布”？

Generative Adversarial Networks

- **Adversarial Training !**
- 用**间接比较**的方法训练生成模型。
- 引入一个**鉴别器**，用于执行真实样本和生成样本之间的区分任务。
- **鉴别器：**对真实和生成的数据进行分类，目标是**最小化**真实数据和生成数据之间的分类误差。
- **生成器：**生成尽可能真实的数据，用来欺骗鉴别器，目标是**最大化**真实数据和生成数据之间的分类误差。
- 在生成对抗网络中，生成器和鉴别器两个网络都试图击败对方，他们之间的竞争使这两个网络在各自的目标上“进步”。

Generative Adversarial Networks

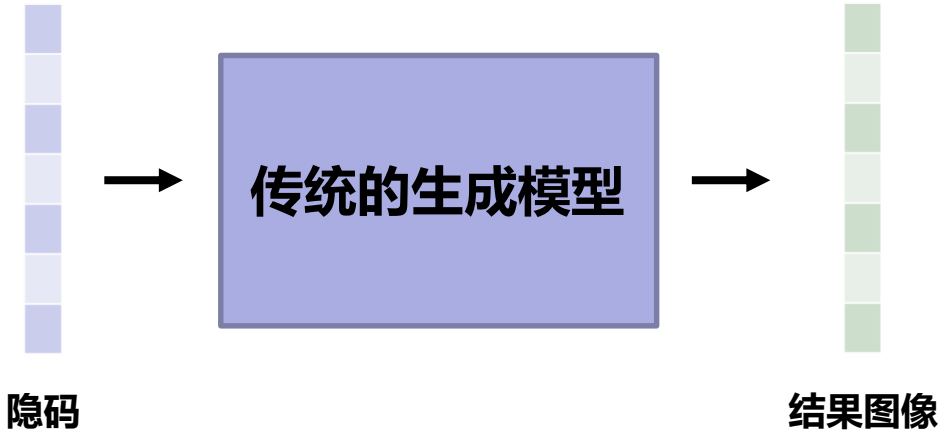


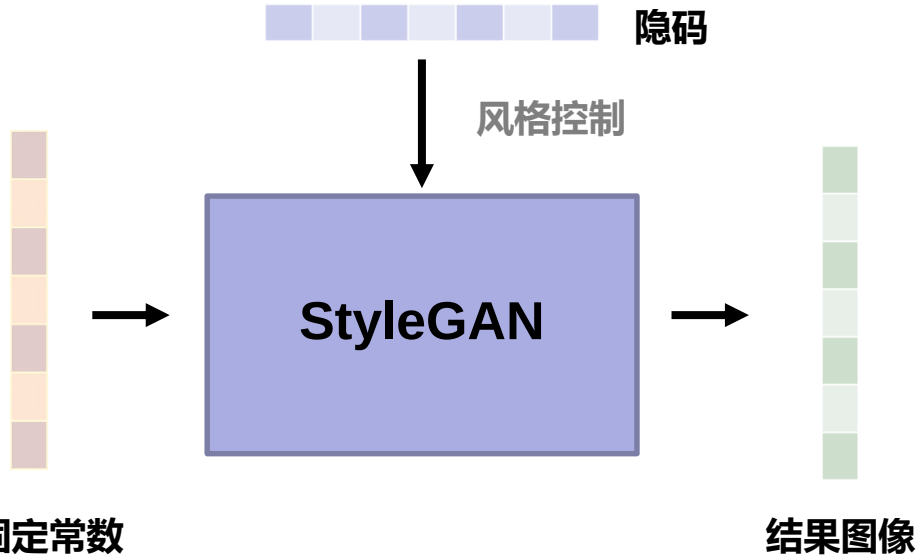
StyleGAN

- Tero Karras, Samuli Laine, and Timo Aila. 2019. **A style-based generator architecture for generative adversarial networks.** In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). 4401–4410.
- 一种引入了风格迁移的生成对抗网络变体。
- **StyleGAN**的两大优点决定了其非常优越的**图像编辑能力**:
  - 生成图像的质量非常高。
  - 隐空间性质优良，有良好的语义**解耦性质**，将不同level的语义属性进行**分离**，能生成更好的**插值结果**。

StyleGAN: 基于风格的生成器

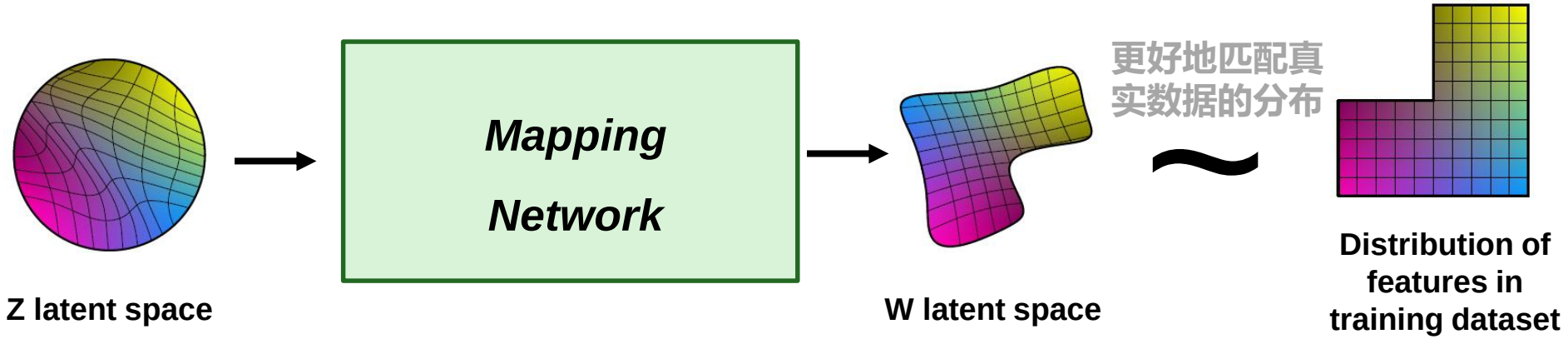
- **传统生成模型**的生成器以隐码作为输入，经过卷积等操作后，输出结果图像。
- **StyleGAN**以一个固定常数作为输入，而隐码用于控制生成器中每一层的**“风格”**。





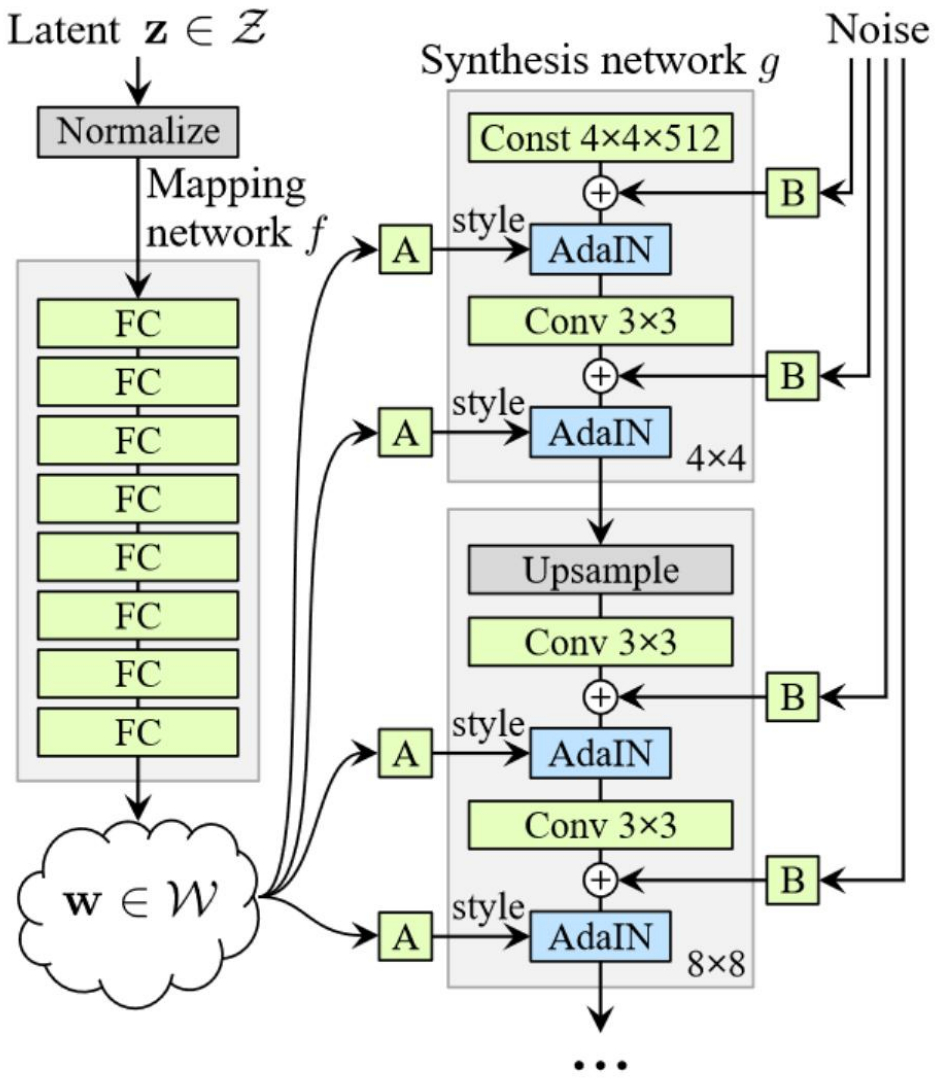
StyleGAN的隐空间

- StyleGAN有两种主要的隐空间：Z隐空间与W隐空间
- **Z隐空间：**一个符合标准正态分布的隐空间。
- **W隐空间：**从Z隐空间中随机采样的z隐码被输入映射网络(mapping network)， 得到的w隐码分布于W隐空间中。 W隐空间的分布是经过映射网络学习得到的，因此与Z隐空间相比，可以更好地匹配真实数据的分布，并且有更好的**解纠缠**（Disentanglement）性质。

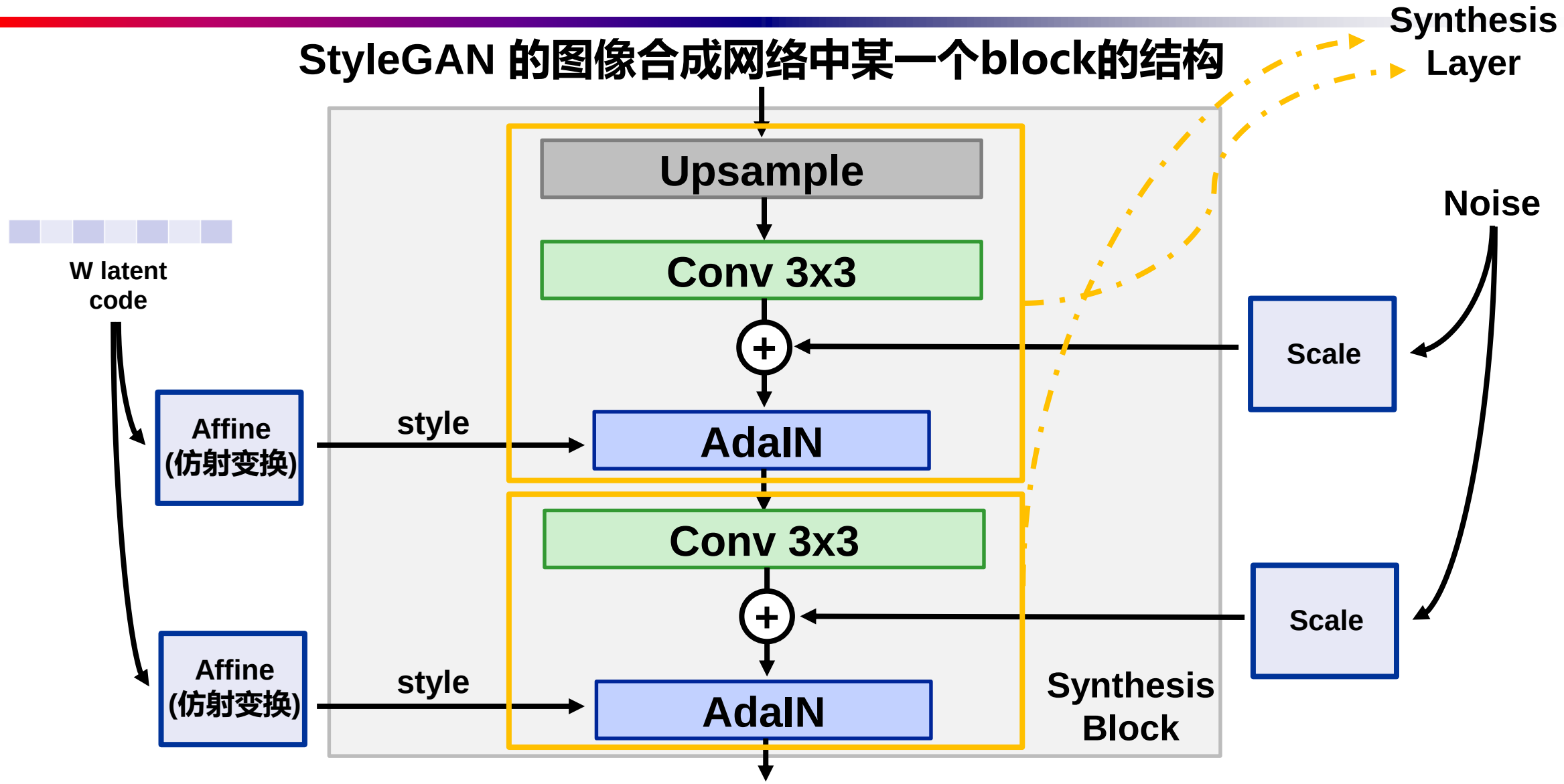


StyleGAN生成器中的风格控制

- 那么，StyleGAN是如何**生成图像的**?
- **StyleGAN利用隐码进行风格控制，隐码通过控制图像的风格决定最终的生成结果。**
- 从Z隐空间中随机采样的z隐码，通过映射网络后转化为w隐码。
- w隐码通过仿射变换后再输入每一层图像合成网络进行风格控制。 这种仿射变换是在图像合成网络的训练过程中学习的。



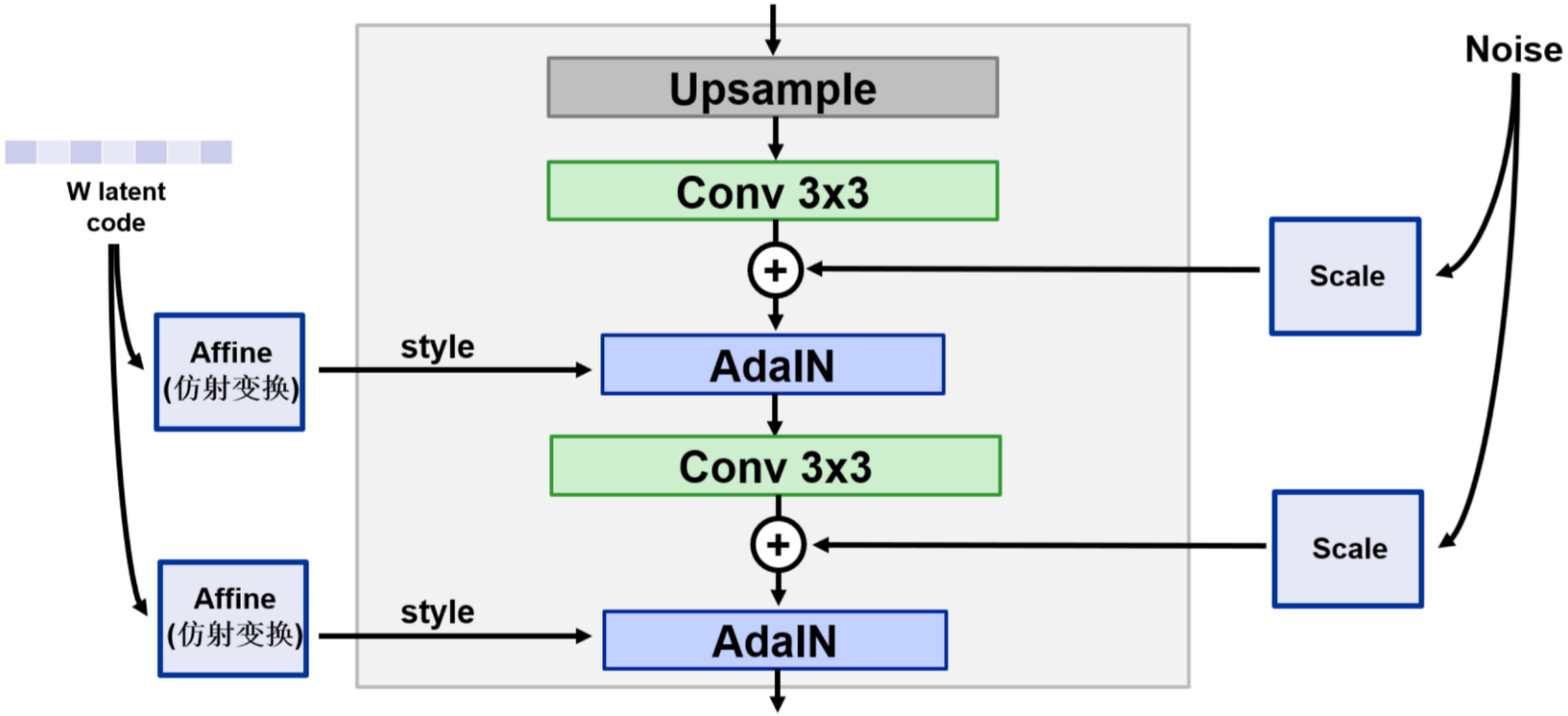
StyleGAN生成器中的风格控制



StyleGAN生成器中的风格控制

- w隐码经过仿射变换(Affine)以后生成风格(style)， 在AdaIN模块（实现风格迁移和风格混合）中，输入的特征图 x_i在归一化后，用style进行缩放和偏置，从而实现对特征图的风格控制。$Styles = (y_s, y_b) = Affine(w)$

$$AdaIN(x_i, Styles) = y_{s,i}\frac{x_i - \mu(x_i)}{\sigma(x_i)} + y_{b,i}$$



StyleGAN生成器中的风格控制

- 而在**StyleGAN**的图像合成网络中，输入不同**Synthesis Layer**的style控制不同粒度的语义特征，即不同粒度的风格。
- **Coarse styles**：控制人脸姿态、人脸几何形状等。**大约对应网络的1-4层。**
- **Middle styles**：发型、部分人脸形状等。**大约对应网络的5-8层。**
- **Fine styles**：肤色、发色、光影、背景等。**大约对应网络剩下的层数。**
- 因此，通过**混合不同粒度**的style，能产生不同层级的**风格混合**效果。

StyleGAN生成器中的风格混合

- **Coarse styles：** 控制人脸姿态、人脸几何形状等
- 下图为将**Source A**的**coarse styles**替换为**Source B**之后的结果

StyleGAN生成器中的风格控制

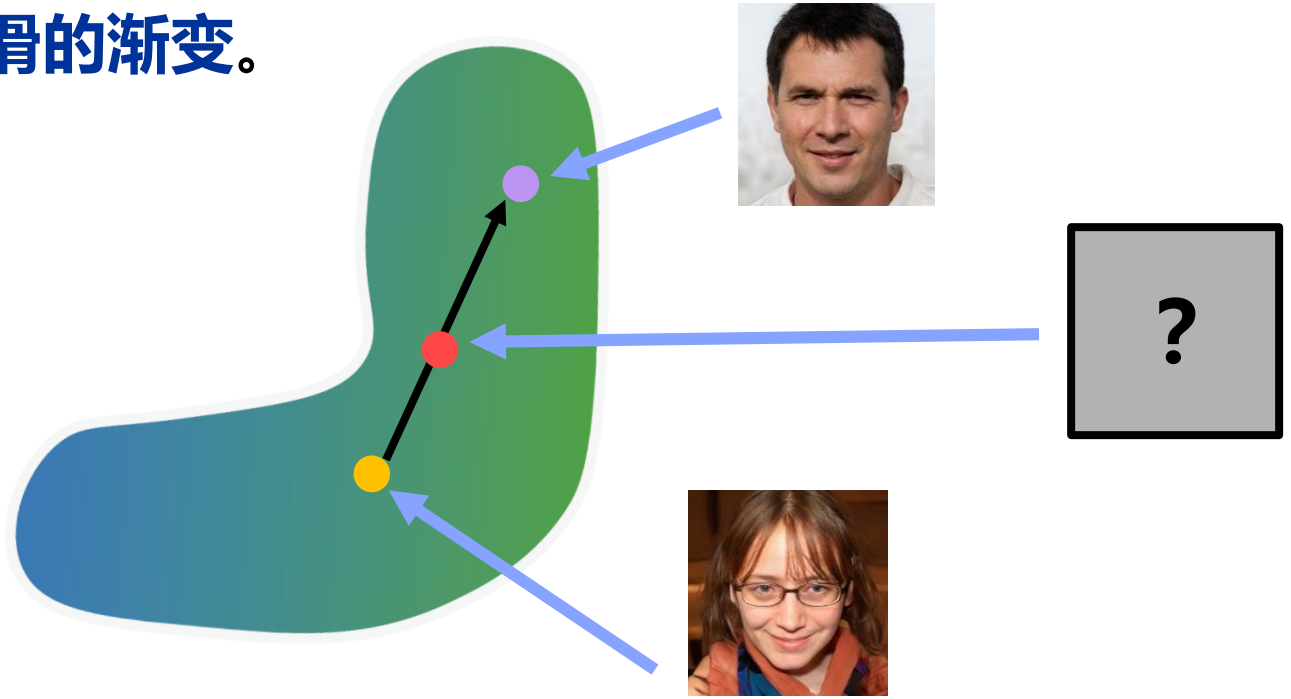
- **Middle styles：**发型、部分人脸形状等
- 下图为将输入**Source A**的**middle styles**替换为输入**Source B**之后的结果

StyleGAN生成器中的风格控制

- **Fine styles：**肤色、发色、光影、背景等
- 下图为将**Source A**的**fine styles**替换为**Source B**之后的结果

StyleGAN中的图像Morphing

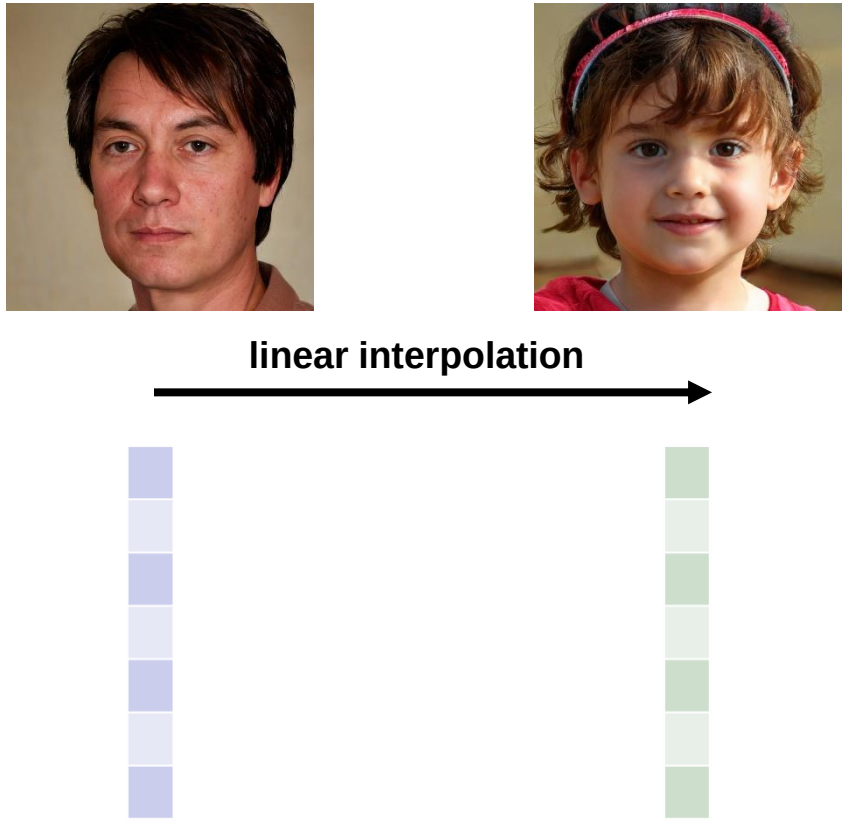
- StyleGAN倾向于学习一个**平滑**的隐空间，即**隐空间的接近区域中采样的隐码能生成相似的图像**。这表明，在隐空间中的连续移动会产生一条**平滑变化的图像路径**，并且每个图像都接近目标域（真实的人脸）。
- 因此，对StyleGAN的两个隐码进行插值，可以在两个人脸图像之间产生非常**自然平滑的渐变**。



StyleGAN中的图像Morphing

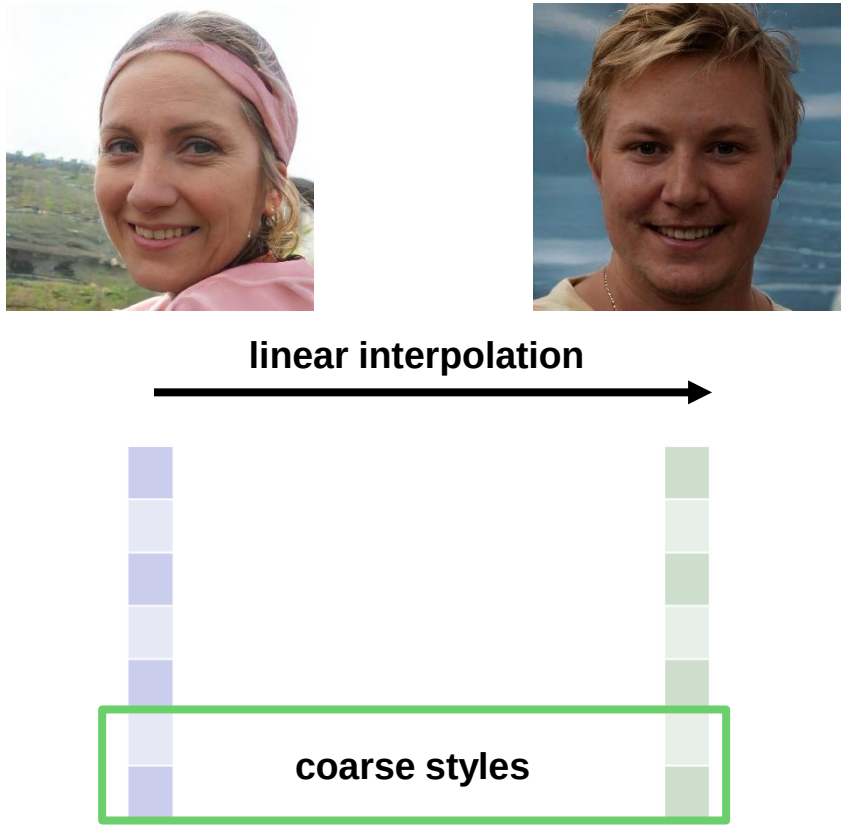
StyleGAN中的图像Morphing

- 直接对两个w隐码进行混合：



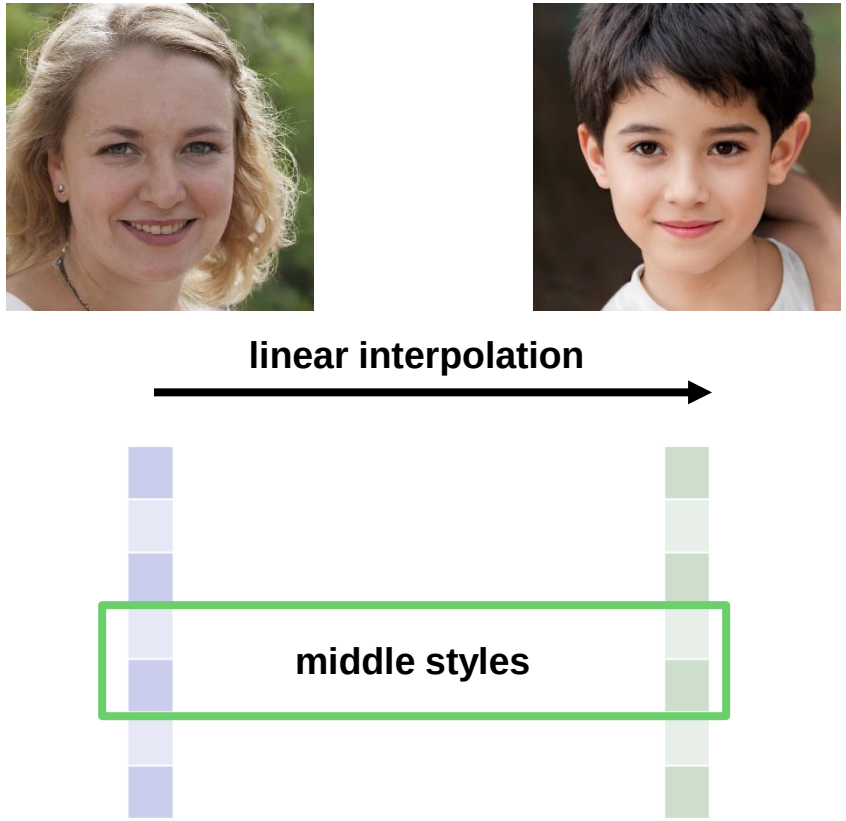
StyleGAN中的图像Morphing

- 对两个w隐码的coarse styles进行混合：



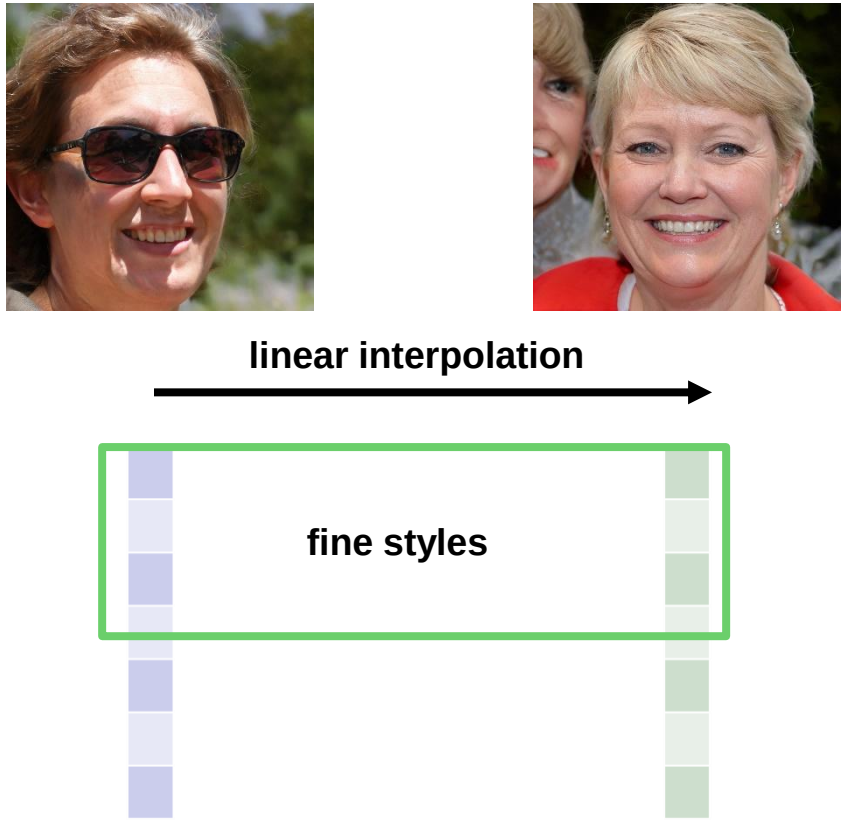
StyleGAN中的图像Morphing

- 对两个w隐码的middle styles进行混合：



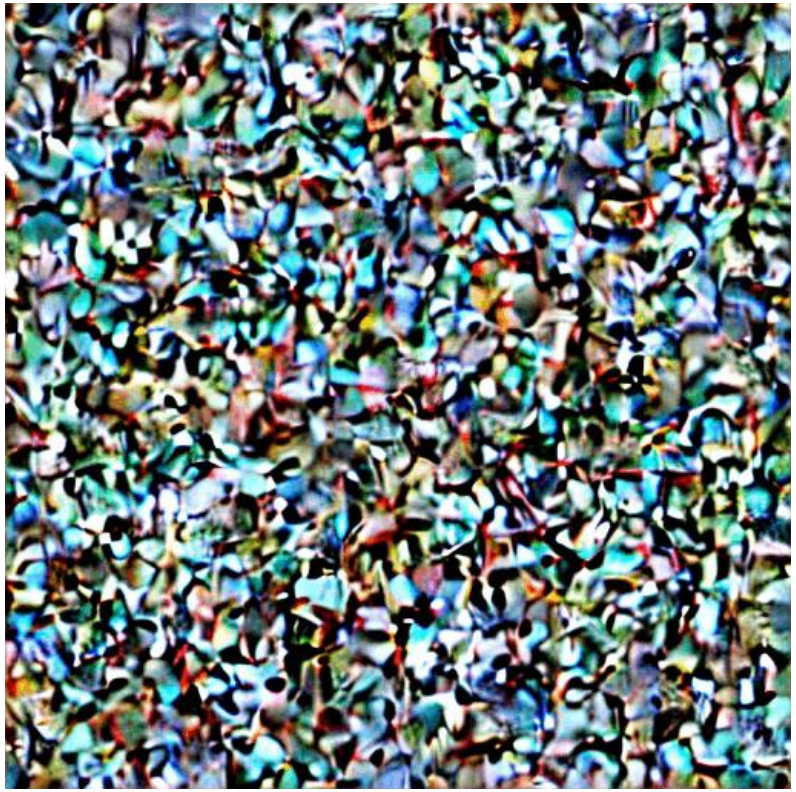
StyleGAN中的图像Morphing

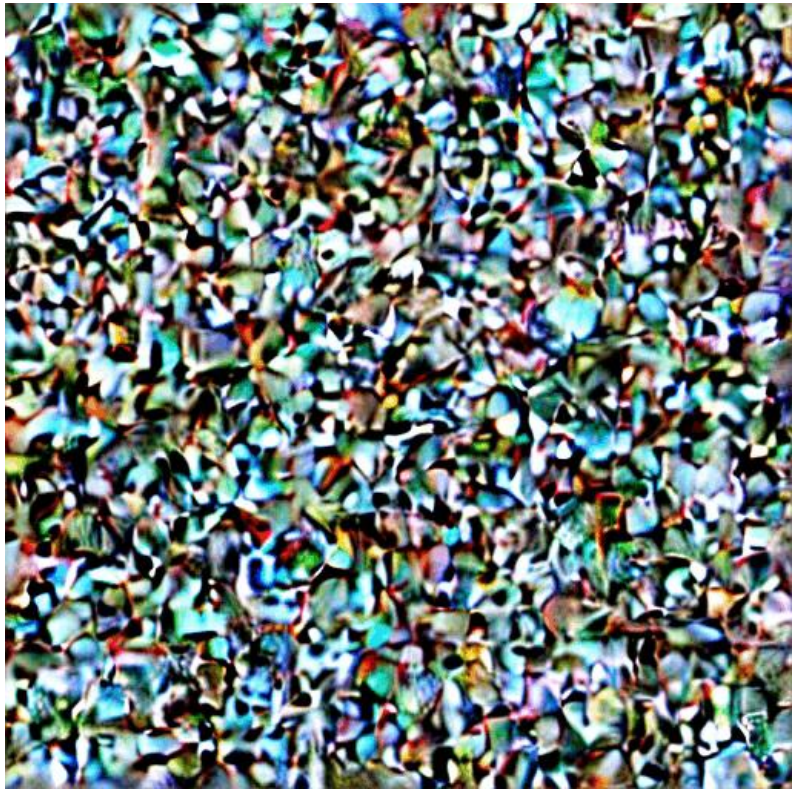
- 对两个w隐码的fine styles进行混合：



基于Diffusion模型的Morphing

基于Diffusion模型的Morphing





Stable Diffusion in Action

Diffusion Models

- 正向扩散(加噪声)
  - $x_0 \rightarrow x_1 \rightarrow \cdots x_T$
  - 将数据分布$x_0 \sim p(x)$,通过扩散过程转变为噪声 $x_T \sim \mathcal{N}(0, \sigma^2 I)$

$\boldsymbol{x_0}$ $\boldsymbol{x_1}$ $\boldsymbol{x_{T-1}}$ $\boldsymbol{x_T}$

- 逆向扩散(去噪)
  - $x_T \rightarrow x_{T-1} \rightarrow \cdots x_0$
- 从噪声分布$x_T \sim \mathcal{N}(0, \sigma^2 I)$中采样,反转扩散过程以生成数据$x_0 \sim p(x)$

使用ControlNet 控制 Stable Diffusion

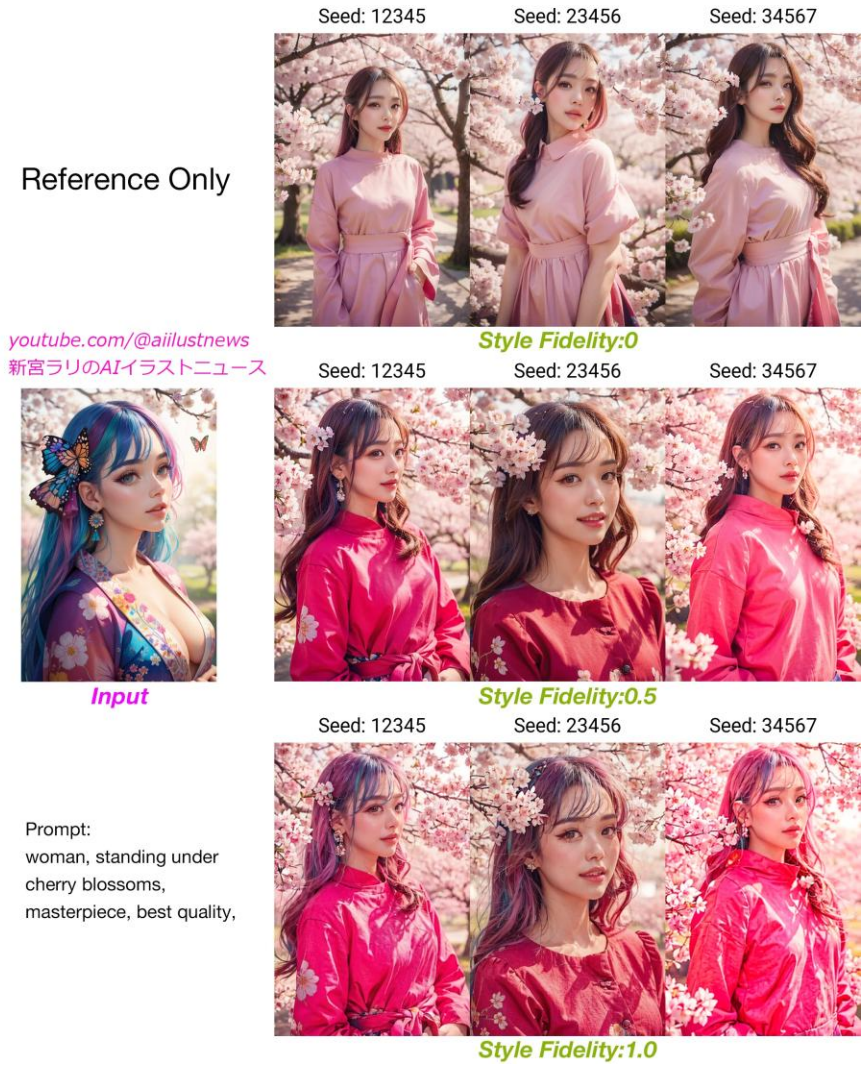
- ControlNet是一个用于控制AI图像生成的插件。它使用了一种称为“Conditional Generative Adversarial Networks”（条件生成对抗网络）的技术来生成图像。与传统的生成对抗网络不同，ControlNet允许用户对生成的图像进行精细的控制，如控制人物轮廓、控制人物姿态、图片生成线稿等等。这使得 ControlNet 在许多应用场景中非常有用，例如计算机视觉、艺术设计、虚拟现实等等。总之，ControlNet 可以帮助用户精准控制 AI 图像的生成，以获得更好的视觉效果。

Zhang L, Rao A, Agrawala M. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision 2023* (pp. 3836-3847). **https://github.com/lllyasviel/ControlNet**

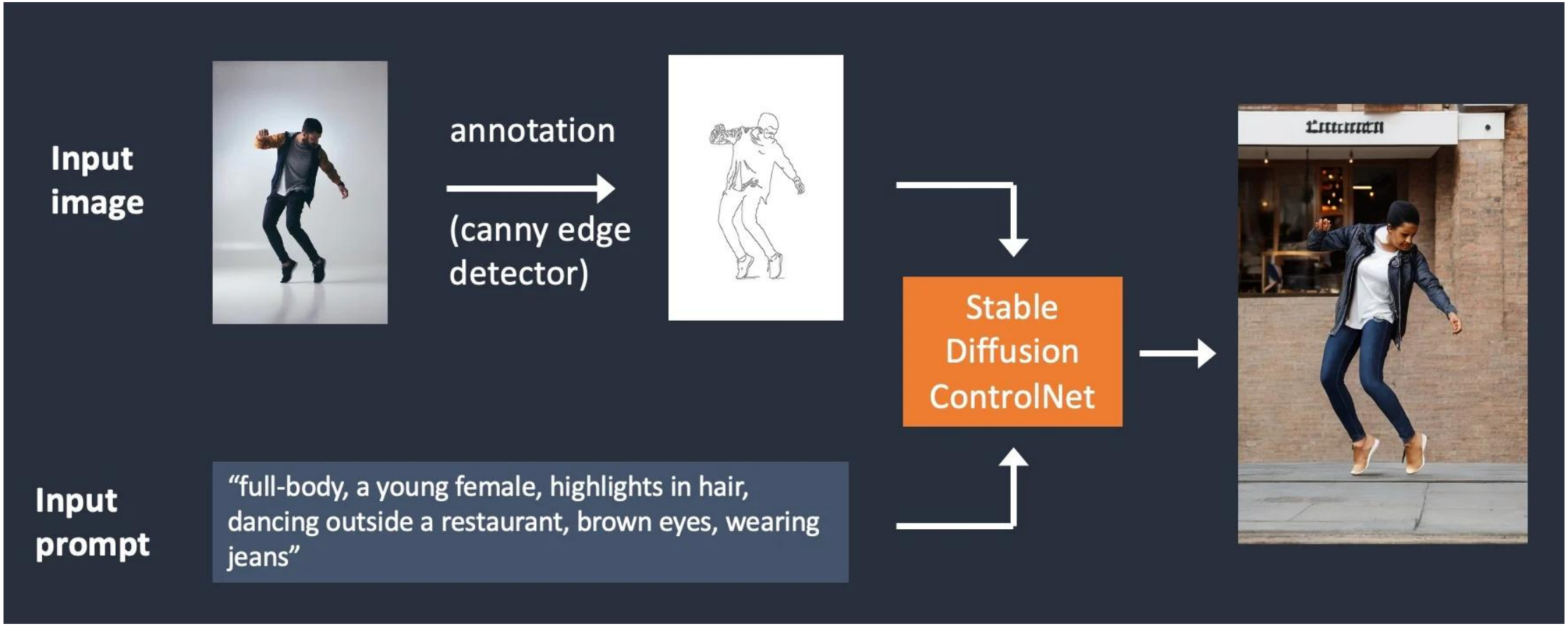
使用ControlNet 控制 Stable Diffusion

使用ControlNet 控制 Stable Diffusion

- Canny 边缘检测 (Canny edge detection)
- 法线贴图 (Normal map)
- OpenPose 姿态 + Prompt
- Scribble （涂鸦）
- Semantic Segmentation（语义分割绘制）
- ……



使用ControlNet 控制 Stable Diffusion



基于Diffusion模型的Morphing

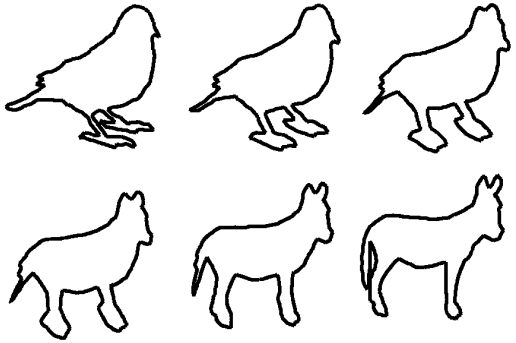
AIGC视频大模型

公司	产品/模型	发布时间	是否公开可用	功能特色
OpenAI	**Sora Turbo**	**2024/12/10**	ChatGPT Plus 和 Pro 用户	支持文本描述、图片或现有视频来生成视频
抖音	即梦	2024/5/9	全面开放	支持画布扩展、局部重绘
Luma AI	Dream Machine	2024/6/13	全面开放	支持**设置首尾帧**、制作无限循环视频
Runway	Gen-3 Alpha	2024/6/17	全面开放	支持文字嵌入，场景切换
生数科技	Vidu	2024/7/30	全面开放	支持比例切换，**首尾帧设置**
快手	**可灵1.5**	2024/9/19	全面开放	支持根据人声对口型，**首尾帧设置**
抖音	豆包 PixelDance/PixelWeed	2024/9/24	针对企业开放内测	支持多动作多主体交互
Pika Labs	Pika 1.5	2024/10/2	全面开放	支持爆炸、融化、粉碎或膨胀的Pika effects；支持局部修改和画面扩张
Meta	Movie Gen	2024/10/4	否	支持为视频匹配节奏和音乐

吾乃麒麟

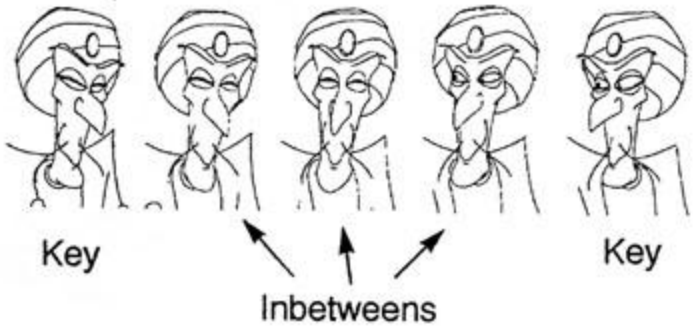
(1929 births)

二维多边形形状渐变(2D Shape Blending)



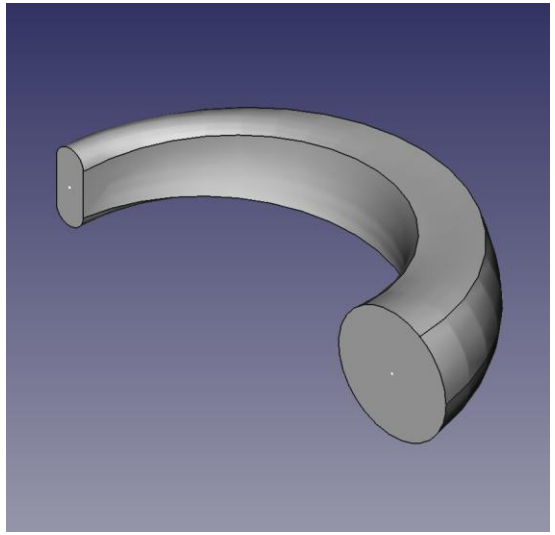
2D Shape Blending问题的引出

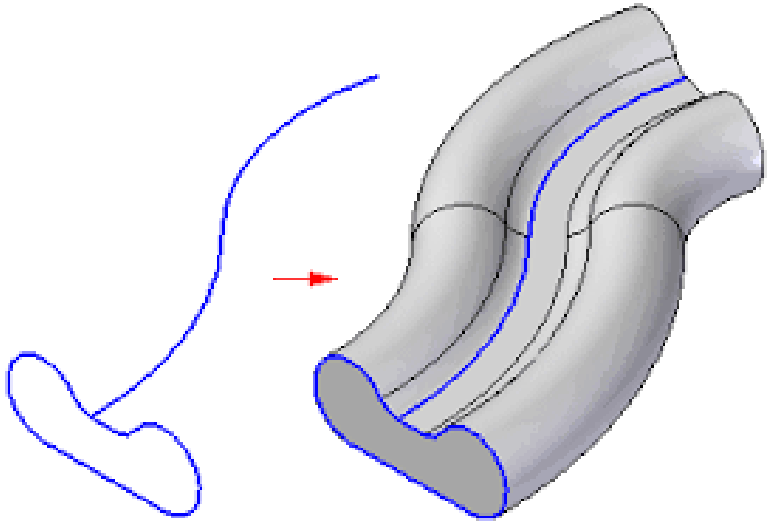
- 在二维角色动画(Character Animation)中，经常会碰到这样的问题：给定一个初始和最终的形状(shape)，我们称它们为**关键帧形状**，求从初始形状**光滑过渡**到最终形状的**中间形状**。
- 这个问题称为二维形状的自然渐变（Shape Blending或Shape Morphing）。
- 该问题实际上分为两个子问题: (1). **顶点的对应关系问题**; (2). **顶点的插值问题**。

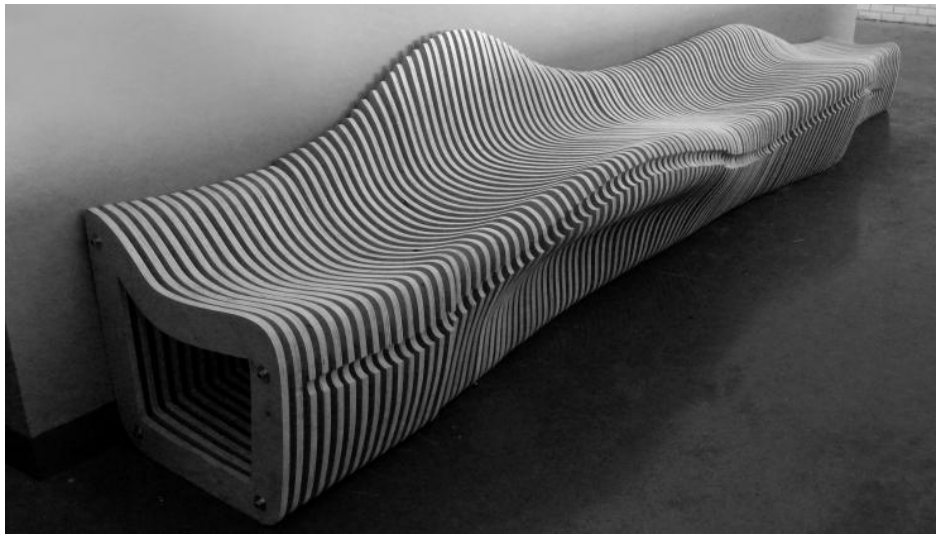


问题的引出

- 二维形状的自然渐变不仅在计算机动画，而且在模式识别、曲面重建和三维造型中也具有重要意义。



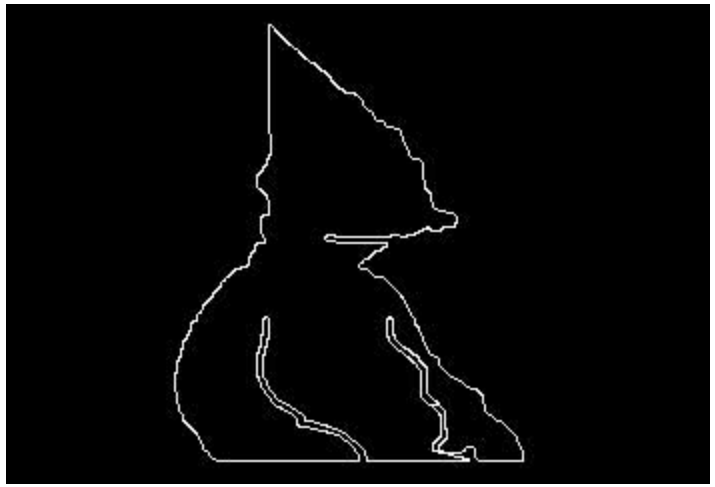




Sweeping曲面

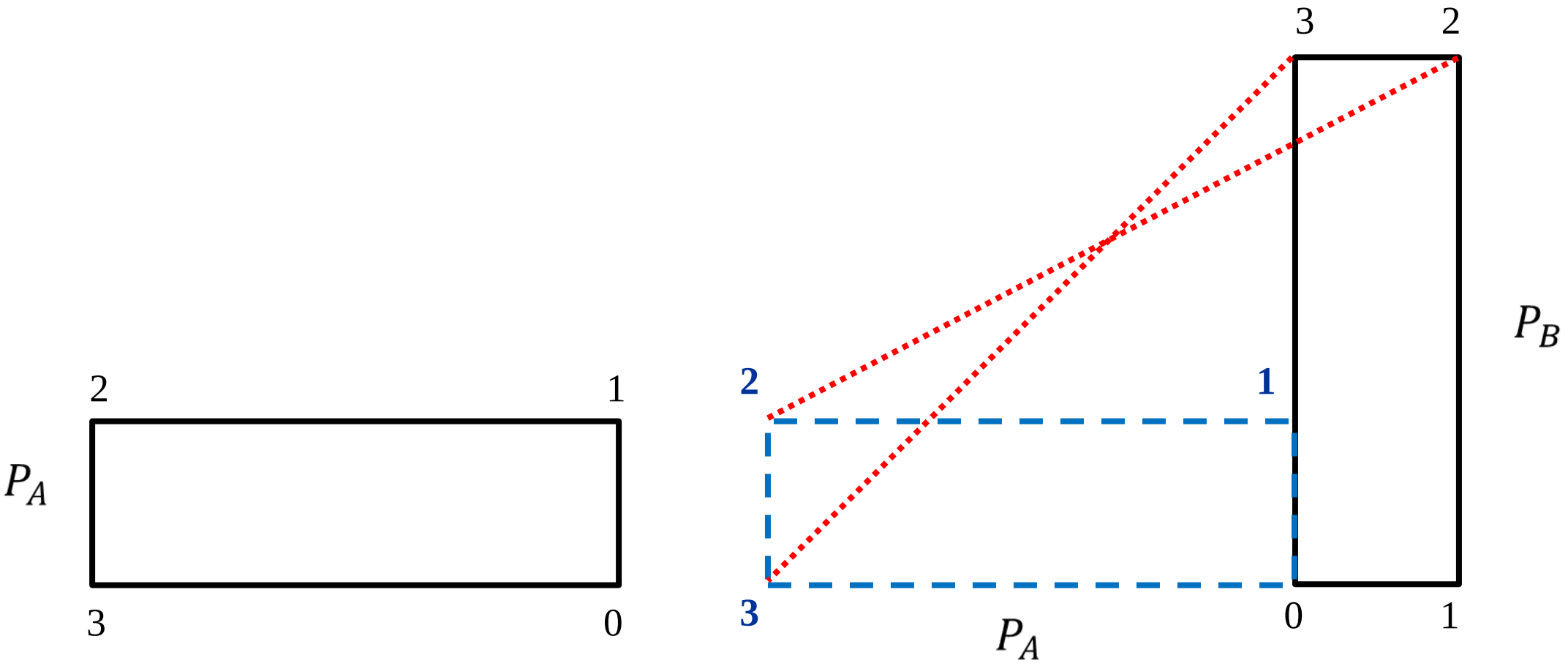
Skin曲面

问题的引出



轮廓

问题的引出



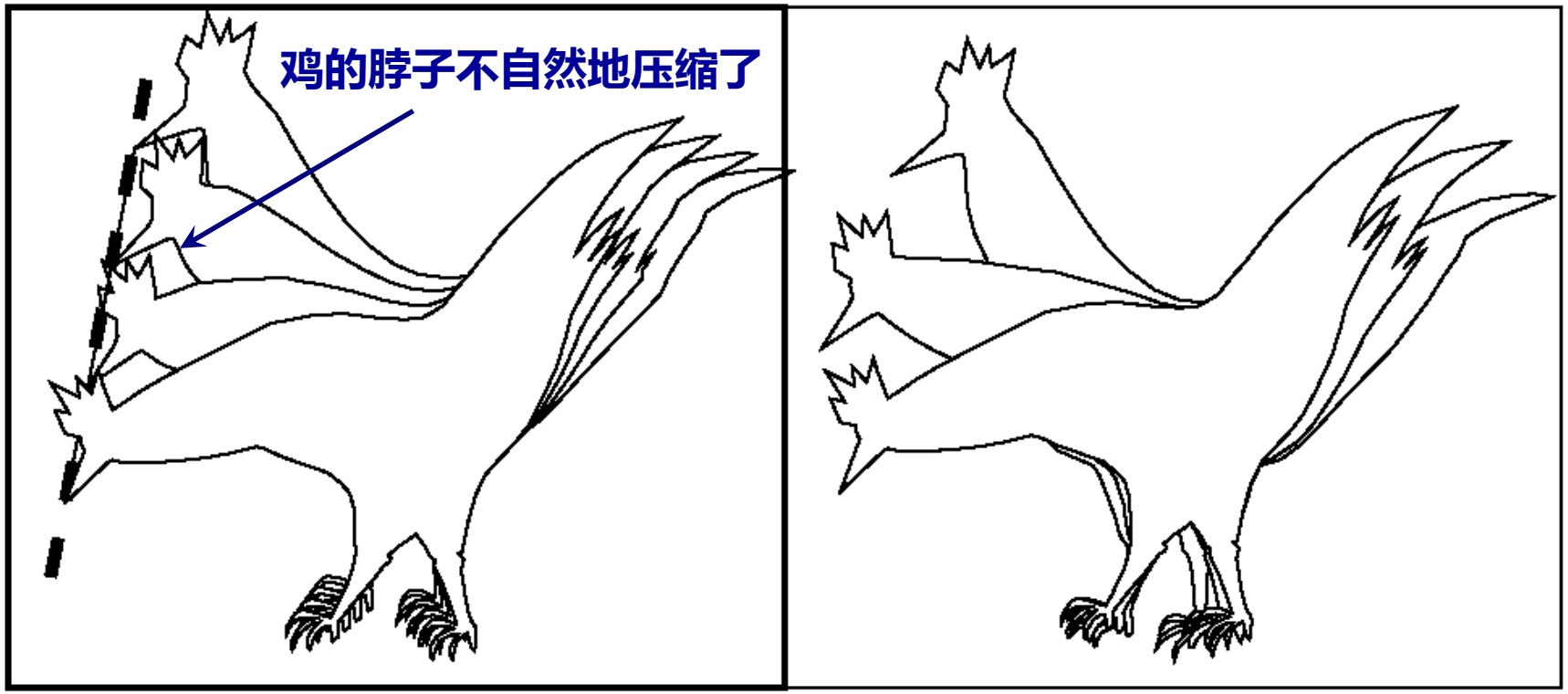
线性插值法

- 假设两个关键帧多边形的顶点为P_{A_i}和P_{B_i} (i=0, 1, 2, 3, …, n)，顶点的数目都为n个，则我们要解决的是：
  - 顶点的对应关系问题，即多边形P_A中的一个顶点与P_B中的哪个顶点对应?
  - 顶点的路径问题，即P_A以何种方式运动到P_B?
- 顶点路径问题一个简单的方法为采用**顶点线性插值**：

$$P_i = (1-t)P_{A_i} + tP_{B_i}, \qquad (i = 0,1,2,\cdots,n-1)$$

- 但线性插值会带来**收缩**(shrinkage)和**纽结**现象(Kink)，这在刚体旋转时表现得尤为明显。

线性插值法的缺点

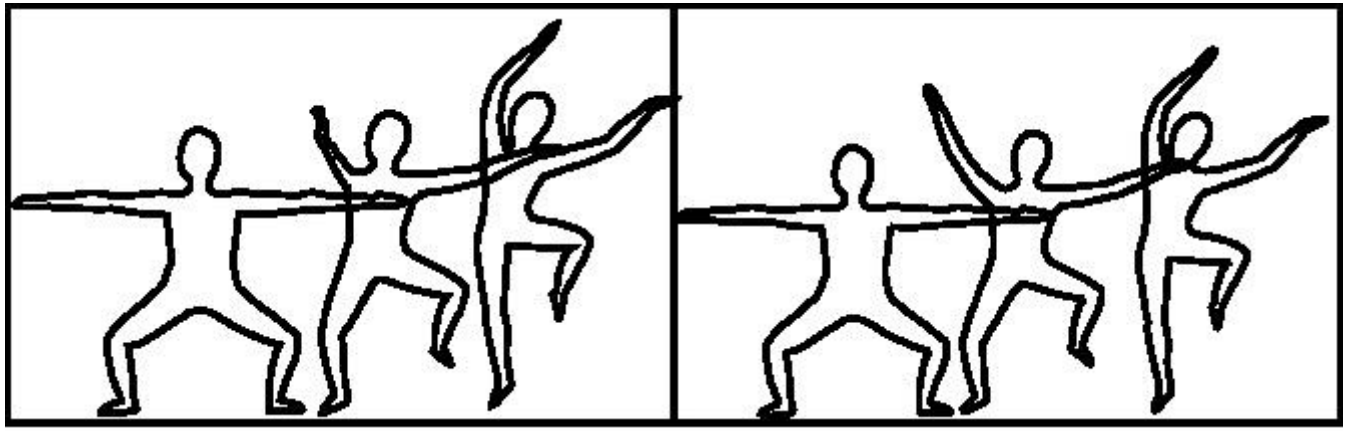


(a) 鸡的脖子压缩了　　(b) 更自然的结果

鸡的形状渐变

基于内在形状插值的多边形渐变方法

- Intrinsic Shape Interpolation的数学原理:乌龟几何+**Lagrange乘数法优化**

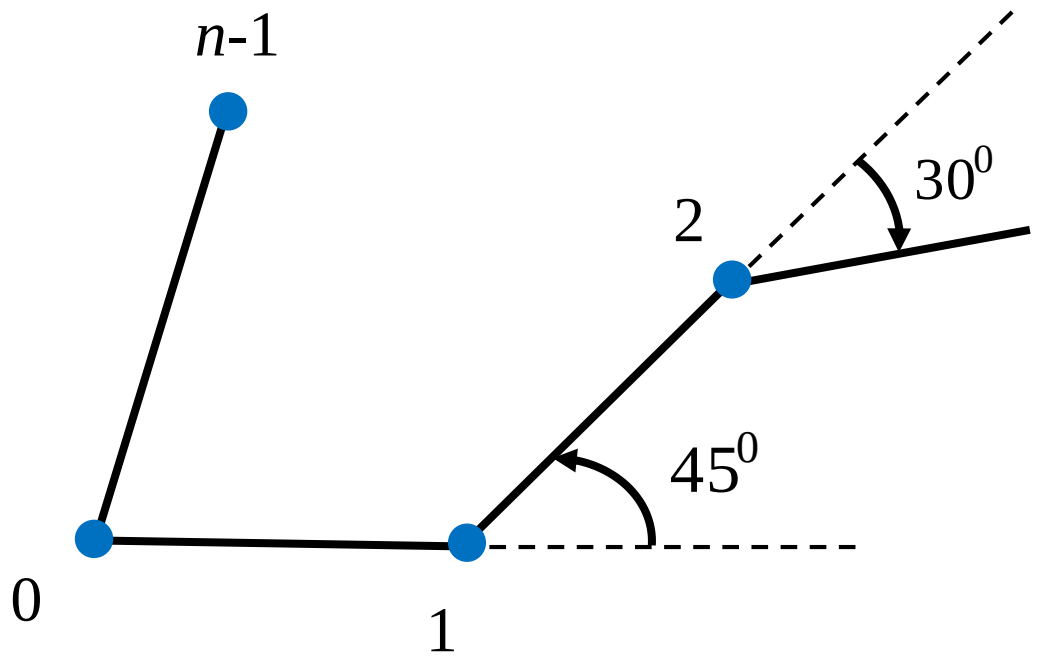


Withered arm **Normal arm**

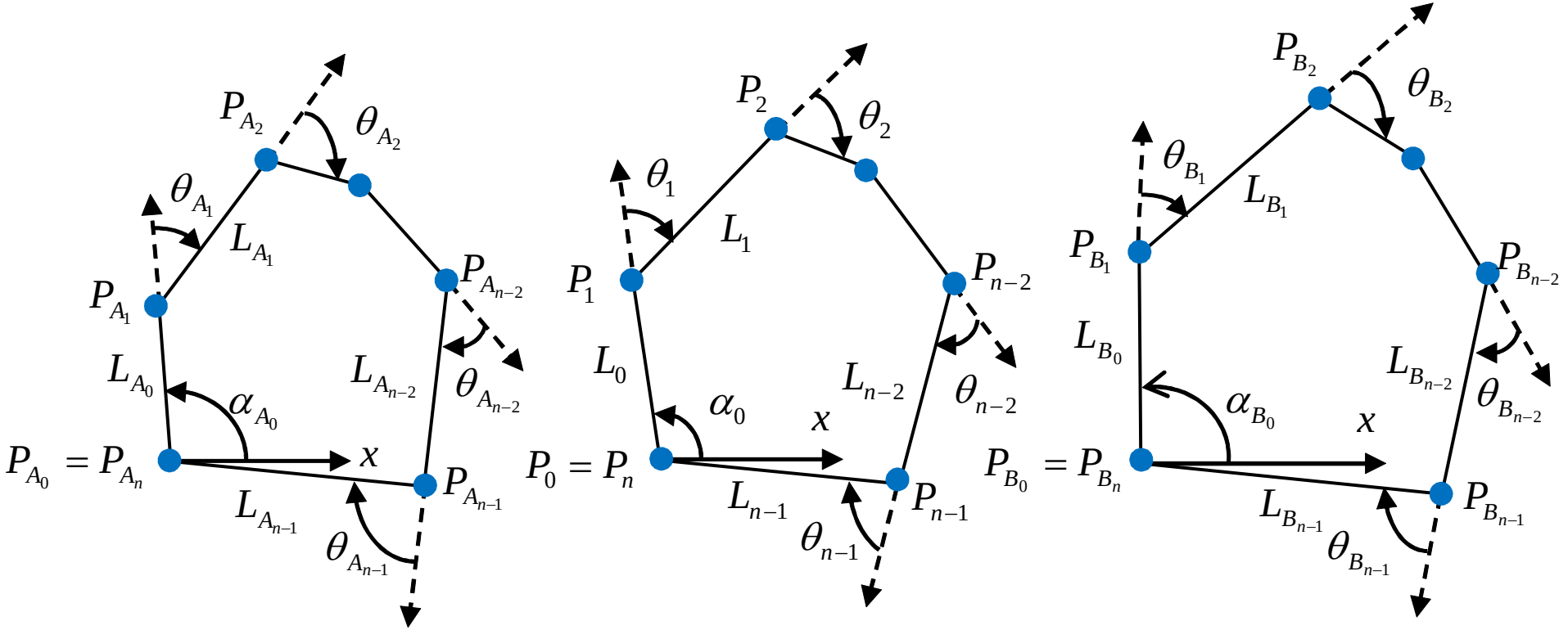
- 参考文献：Sederberg T W, Gao P S, Wang G J, Mu H. 2D shape blending: an intrinsic solution to the vertex path problem. Computer Graphics, 1993, 27(3):15~18

乌龟几何

- 在笛卡尔坐标系中，多边形是通过顶点的坐标显式给出的。
- 但多边形也可以通过乌龟几何来定义，即通过顶点处的边长和有向角来定义。例如，以某一点为起点，向东往前走10米，往左拐45^0，继续往前走6米，往右拐30^0，向前走5米，…，最后得到一多边形。
- 因而一个自然的想法是，能否对关键帧多边形的边长和顶点角进行插值来产生比线性插值更好的效果？
- 回答是肯定的，因为这种插值具有更好的几何意义。



多边形的内在定义



- 设m=n-1，逆时针方向的角为正的，P_0为形状的平移锚点(anchor point)，我们的目标是计算中间多边形的顶点P_i (i=1, 2, …, m)，$0 \le t \le 1$。
- 通过计算多边形P_A和P_B的边长和有向顶点角，得到多边形的内在定义$\{\alpha_0, L_0, (\theta_i, L_i)_{i=1}^{m}\}$:

$$\alpha_{A_0} = \theta(x, P_{A_0}, P_{A_1}) \qquad \theta_{A_i} = \theta(P_{A_{i-1}}, P_{A_i}, P_{A_{i+1}}) \qquad L_{A_i} = |P_{A_{i+1}} - P_{A_i}|$$

$$\alpha_{B_0} = \theta(x, P_{B_0}, P_{B_1}) \qquad \theta_{B_i} = \theta(P_{B_{i-1}}, P_{B_i}, P_{B_{i+1}}) \qquad L_{B_i} = |P_{B_{i+1}} - P_{B_i}|$$

其中：$\theta(P_{A_{i-1}}, P_{A_i}, P_{A_{i+1}})$表示边$P_{A_{i-1}}P_{A_i}$和边$P_{A_i}P_{A_{i+1}}$之间的**有向夹角**

中间多边形生成

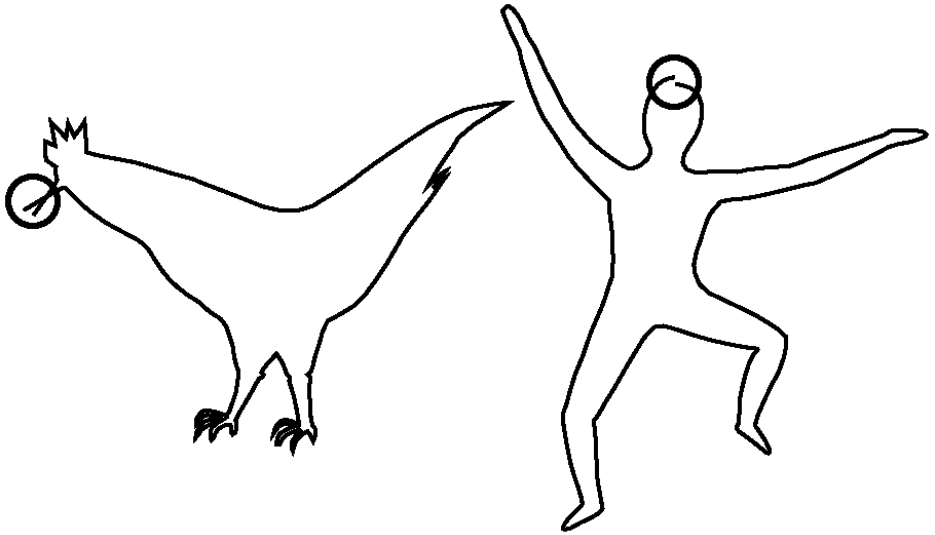
- 那么中间多边形可由插值相应的边长和顶点角得到：

$$\alpha_0 = (1-t)\alpha_{A_0} + t\alpha_{B_0}$$

$$\theta_i = (1-t)\theta_{A_i} + t\theta_{B_i} \qquad (i = 1,2,\cdots,m)$$

$$L_i = (1-t)L_{A_i} + tL_{B_i} \qquad (i = 1,2,\cdots,m)$$

- 遗憾的是，这样得到的多边形通常是不封闭的。但实验发现，得到多边形的起始点和终止点非常接近。



不封闭的多边形

中间多边形生成

- 一个解决方法为保持插值的顶点角不变，适当调整插值得到的边长(Edge **Tweaking**)：

$$L_i = (1-t)L_{A_i} + tL_{B_i} + S_i,\ \ (i = 0,1,2,\cdots,m)$$

- 如果关键多边形的某一条对应的边具有相同的长度L，那么我们可认为中间多边形的边长也为L，因此可认为$|S_i| \propto |L_{A_i} - L_{B_i}|$。为了防止除以零，我们定义：

$$L_{AB_i} = \max\left\{\left|L_{A_i} - L_{B_i}\right|, L_{tol}\right\}, \qquad (i = 0,1,2,\cdots,m)$$

其中：$L_{tol} = 0.0001 \times (\max_{i\in[0,m]}\left|L_{A_i} - L_{B_i}\right|)$

中间多边形生成

- 我们的目标是为了求得 $S_0, S_1, \cdots, S_m$ ，使目标函数：

$$f(S_0, S_1, \cdots, S_m) = \sum_{i=0}^{m} \frac{S_i^2}{L_{AB_i}^2}$$

最小，并且 $S_0, S_1, \cdots, S_m$ 应满足约束条件(强迫多边形封闭):

$$\varphi_1(S_0, S_1, \cdots, S_m) = \sum_{i=0}^{m} \left[(1-t)L_{A_i} + tL_{B_i} + S_i\right]\cos\alpha_i = 0$$

$$\varphi_2(S_0, S_1, \cdots, S_m) = \sum_{i=0}^{m} \left[(1-t)L_{A_i} + tL_{B_i} + S_i\right]\sin\alpha_i = 0$$

其中α_i是由矢量P_iP_{i+1}和x轴构成的有向角：$\alpha_i = \alpha_{i-1} + \theta_i, (i = 1, 2, \cdots, m)$

中间多边形生成

- 这是个具有约束条件的极值问题，可用Lagrange乘数法来求解。引进Lagrange函数:

$$\Phi(\lambda_1,\lambda_2,S_1,S_2,\cdots,S_m)=f+\lambda_1\varphi_1+\lambda_2\varphi_2$$

其中$\boldsymbol{\lambda}_1$，$\boldsymbol{\lambda}_2$为**Lagrange**乘数。对$\boldsymbol{\Phi}$求偏导得:

$$\frac{\partial\Phi}{\partial S_i}=\frac{2S_i}{L^2_{AB_i}}+\lambda_1\cos\alpha_i+\lambda_2\sin\alpha_i=0 \qquad (i=0,1,\cdots,m)$$

$$\sum_{i=0}^{m}[(1-t)L_{A_i}+tL_{B_i}+S_i]\cos\alpha_i=0$$

$$\sum_{i=0}^{m}[(1-t)L_{A_i}+tL_{B_i}+S_i]\sin\alpha_i=0$$

中间多边形生成

- 由第一式求得S_i代入后二式得：

$$\begin{cases} E\lambda_1 + F\lambda_2 = U \\ F\lambda_1 + G\lambda_2 = V \end{cases}$$

其中：

$$E = \sum_{i=0}^{m} L_{AB_i}^2 \cos^2 \alpha_i, F = \sum_{i=0}^{m} L_{AB_i}^2 \sin \alpha_i \cos \alpha_i, G = \sum_{i=0}^{m} L_{AB_i}^2 \sin^2 \alpha_i$$

$$U = 2\left\{ \sum_{i=0}^{m} \left[(1-t)L_{A_i} + tL_{B_i} \right] \cos \alpha_i \right\}$$

$$V = 2\left\{ \sum_{i=0}^{m} \left[(1-t)L_{A_i} + tL_{B_i} \right] \sin \alpha_i \right\}$$

中间多边形生成

- 如果 $EG - F^2 \neq 0$ ，则

$$\lambda_1 = \begin{vmatrix} U & F \\ V & G \end{vmatrix} \Big/ \begin{vmatrix} E & F \\ F & G \end{vmatrix}$$

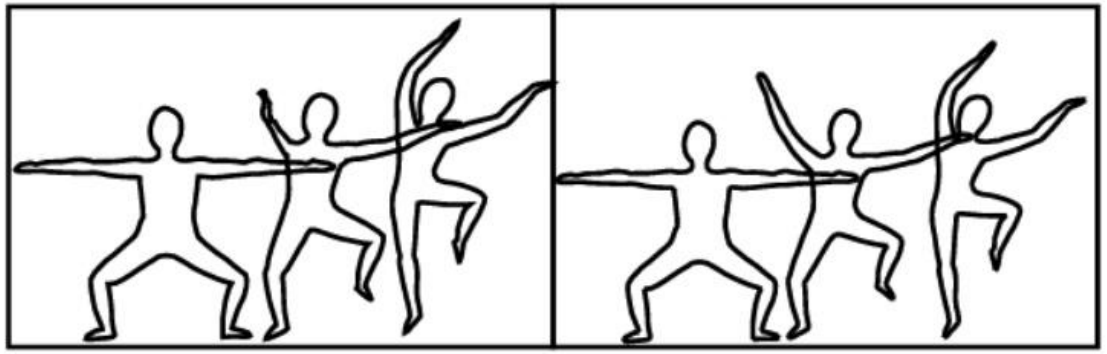
$$\lambda_2 = \begin{vmatrix} E & U \\ F & V \end{vmatrix} \Big/ \begin{vmatrix} E & F \\ F & G \end{vmatrix}$$

$$S_i = -\frac{1}{2} L_{AB_i}^2 (\lambda_1 \cos\alpha_i + \lambda_2 \sin\alpha_i),\ (i = 0, 1, 2, \cdots, m)$$

因而可得到中间多边形顶点（x_i, y_i）的坐标：

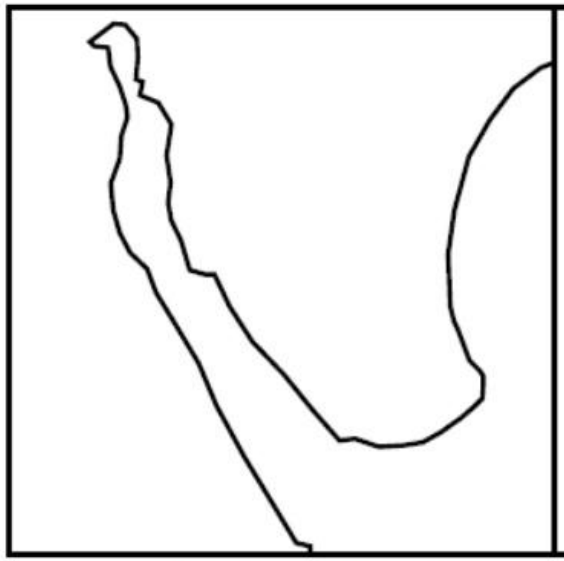
$$\begin{cases} x_i = x_{i-1} + L_{i-1} \cos\alpha_{i-1} \\ y_i = y_{i-1} + L_{i-1} \sin\alpha_{i-1} \end{cases}$$

与线性插值的结果对比

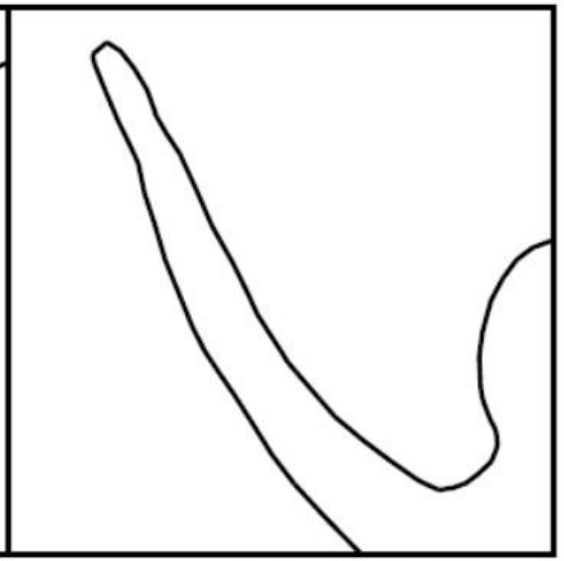


a. Withered Arm

b. Normal Arm



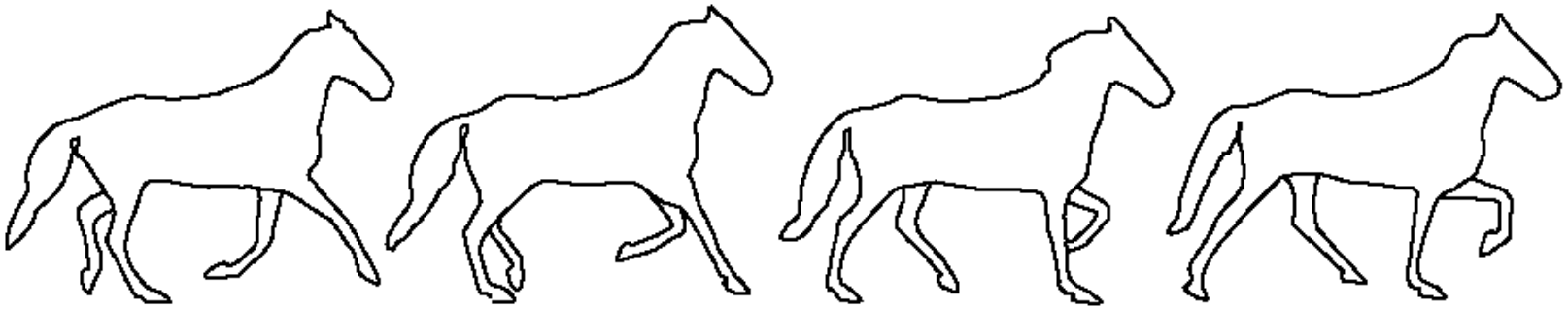
Linear vertex paths



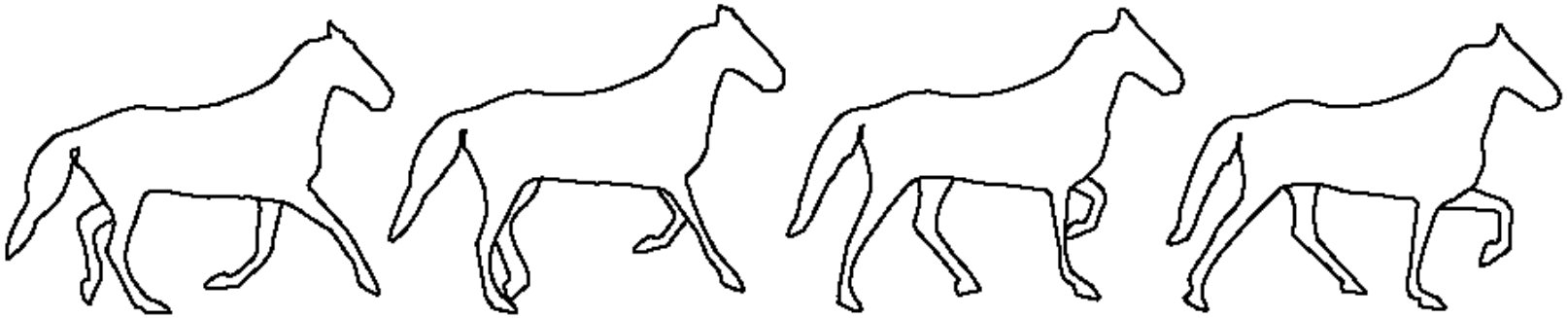
Intrinsic blend

结果

- 实验表明，对于角色动画中的许多应用，内在插值的形状渐变效果良好。



数字化得到的形状(形状取自经典摄影图书《**Animals in Motion**》)



给定的 **内在插值得到的** **内在插值得到的** **给定的**

算法实现

输入： 两个关键帧多边形的顶点

- 计算两个关键帧多边形的内在表示；
- 解线性方程组，计算λ_1, λ_2；
- 计算S_i；
- 中间多边形顶点的坐标；

输出： 中间帧多边形的顶点

DEMO

内在形状插值法的优缺点

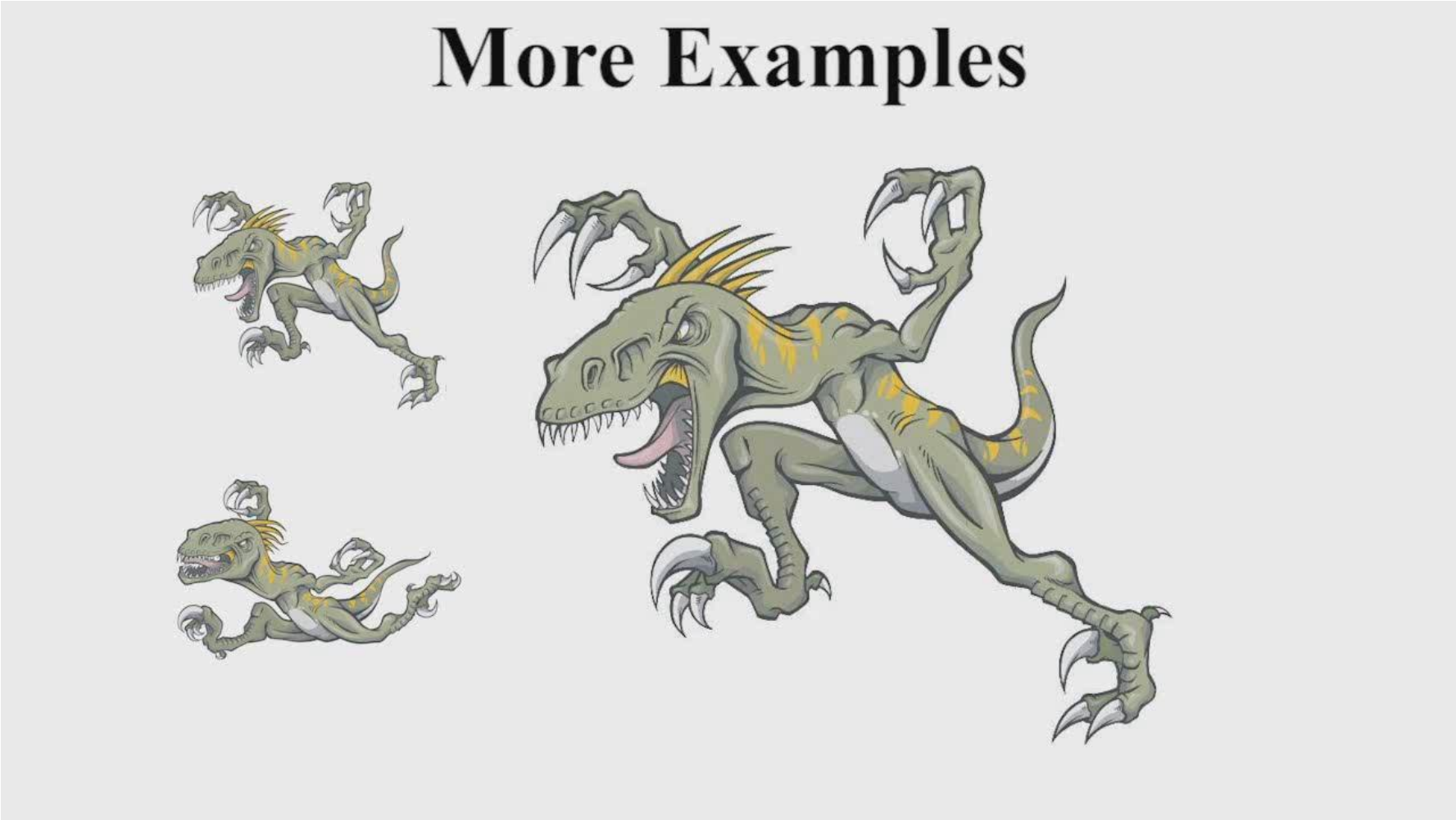
- **优点：**
  - 简单；
  - 能在一定程度上避免形状插值中出现的收缩和纽结现象，这在二维角色动画中尤其有用。
- **缺点：**
  - 顶点对应关系需要用户交互给出；
  - 结果依赖于顶点对应关系的好坏；
  - 当源和目标多边形中包含短边时，由于计算短边的方向不稳定，中间帧多边形有可能产生较严重的畸变。

基于模糊数学的二维多边形形状渐变

- 1996年，Zhang提出了一种基于模糊数学的二维多边形形状渐变方法。
- 它采用基于模糊数学的方法来建立源多边形和目标多边形**顶点的对应关系**以及**顶点之间的插值**。
- **优点：**
  - 能处理具有不同位置、方向、大小或形状的多边形。该方法不仅稳定，而且易于推广到曲线形状。

*** Zhang Yuefeng. A fuzzy approach to digital image warping. IEEE Computer Graphics and Applications, 1996, 16(6):34~41**

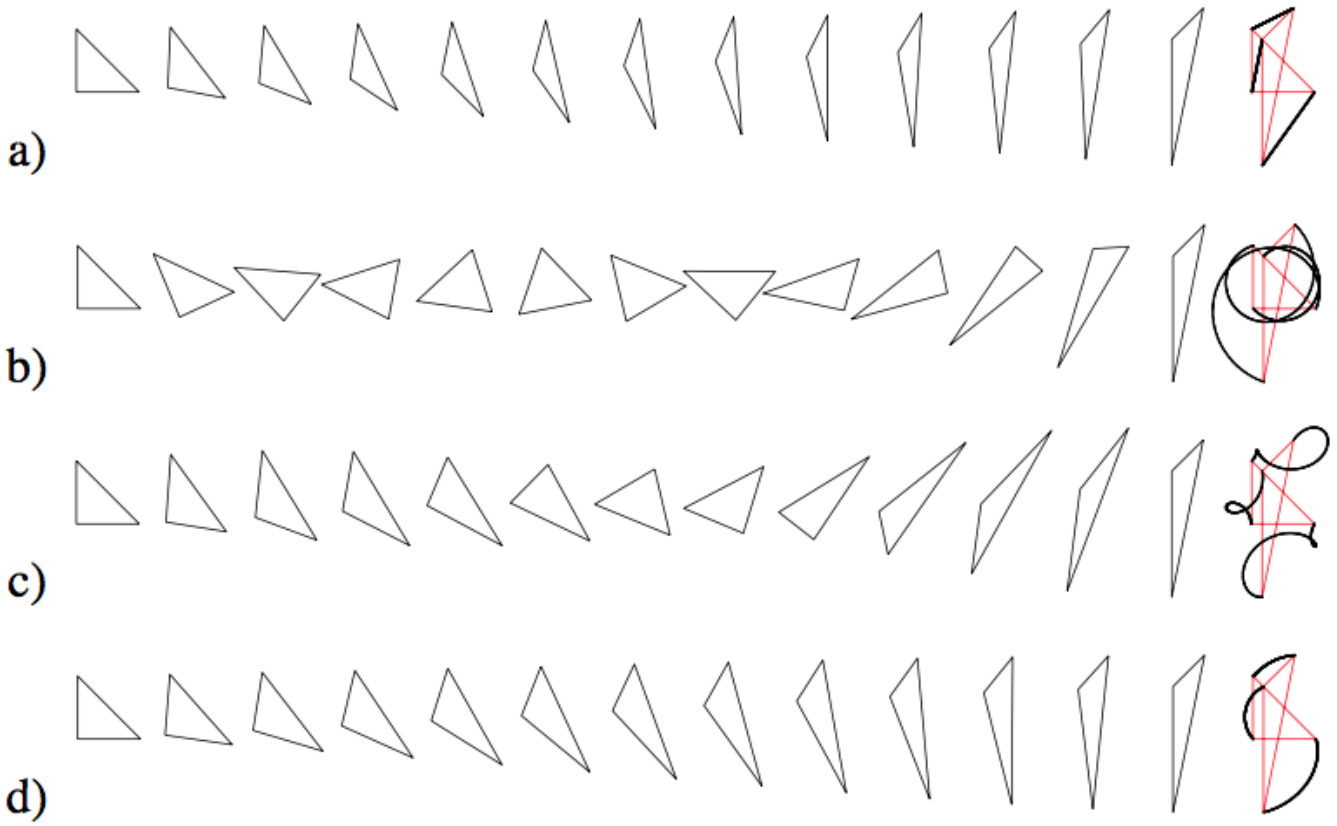
More 2D Shape Blending DEMO



More 2D Shape Blending DEMO

Morphing in After Effects

进一步读的论文



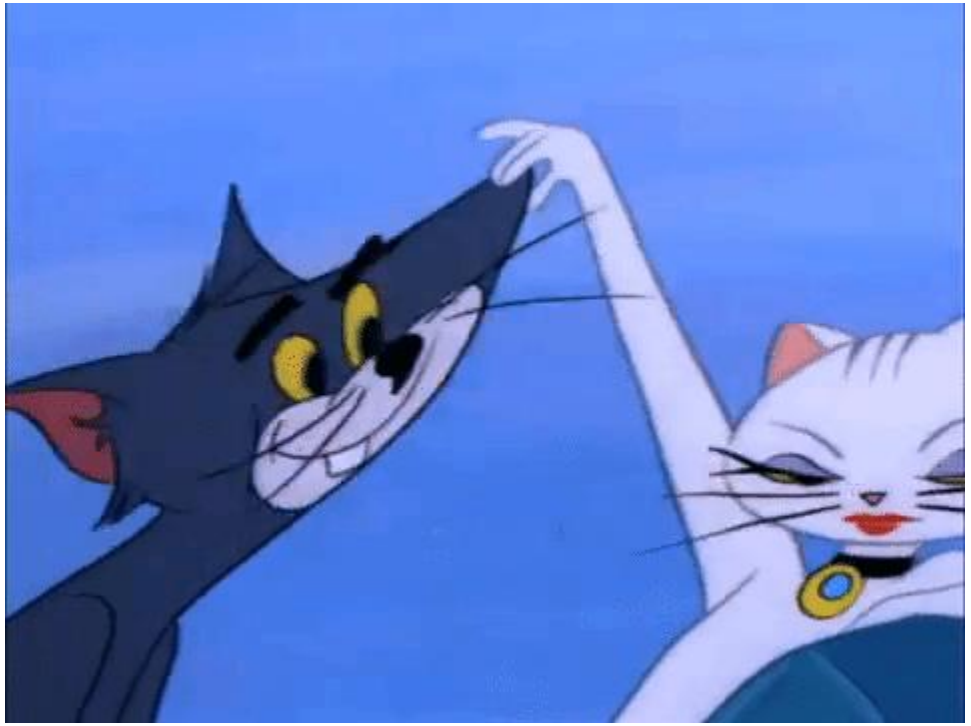
Transformations of a single triangle. (a) Linear vertex interpolation. (b-d) An affine map from the source to the target triangle is computed and factored into rotational and scale-shear parts. Intermediate triangles are constructed by linearly interpolating the angle(s) of rotation, the scaling factors, and the shear parameter. (b) corresponds to Equation 4; (c) shows the results of reducing the overall angle of (b) by subtracting 2PI from one of the angles; (d) correspondsto Equation 5 and represents the method of our choice. The last column in all rows shows plots of the vertex paths.

Alexa, M., Cohen-Or, D., & Levin, D. (2000, July). As-rigid-as-possible shape interpolation. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques* (pp. 157-164).

总结

- 基于StyleGan的肖像Morphing是社交媒体中常用的技术。

- 2D Shape Blending 是传统关键帧角动画设计中经常碰到的问题，可节约动画师设计中间帧的工作量。**局限性：**自动计算得到的结果不一定是动画师所需要的。



The End