

# 实时物理模拟之 Warped Stiffness 方法

Warped Stiffness(刚度弯曲)法是解决有限元方法模拟物体大变形下的一种方法，由 Muller 提出。本文从原理、实现方法上对 Warped Stiffness 方法进行分析，最后在 Window 上，利用 DirectX 渲染框架上实现了此算法。

各个章节目录安排如下：第一部分介绍了图形学中物理模拟的情况。第二部分介绍有限元在较大变形下失真的解决方法，即 Warped Stiffness 方法。第三部分介绍如何获取此变换矩阵的充分条件，第四部分介绍如何获得此旋转矩阵和具体的数值方法，第五部分简单介绍代码实现，第六部分是代码核心部分的讲解，第 7 部分总结了此方法的特点和更有优势的方法。

备注。本文是学习 Warped Stiffness 方法的一个总结，在受邀为投稿后，就将以前的学习笔记整理成知乎体的格式，如有不正确的地方还请各位指正。

## 目录

### 一、综述

1. 物理模拟概述
2. 模拟概述
3. 有限元方法的局限性

### 二、大变形的模拟方法

### 三、变换矩阵

1. 变换矩阵的形式
2. 验证变换的正确性
3. 简单四边形的变换矩阵
4. 三维空间的情况

### 四、极分解

1. SVD 分级的数学几何意义
2. 极分级的概念和唯一性
3. 极分级的数值方法
4. 数值实例

### 五、代码简单讲解

### 六、核心代码讲解

1. 程序模块
2. 核心模块讲解
- 3 执行效果

### 七、总结和后续扩展阅读

# 一、综述

## 1. 物理模拟概述

物理模拟的目标。图形学渲染解决单帧画面中光照看起来像不像的问题，而物理模拟解决多帧中物体运动像不像的问题。物理模拟是图形学的一个分支，就我所知，目前尚无明确分类，从模拟的类型分为刚体模拟和非刚体模拟。其中刚体模拟比较成熟，已经有开源软件，Bullet, PhysX, 还有其他...；非刚体模拟则难以概括，也无通用的开发包，但有以下几个热点，流体(需求大，从事人员多，范围也广，如多相流，浮力等)，布料(虚拟试衣)，头发、烟雾等热点。查看各种顶级会议论文和行业需求就知道当前方向。

国内情况(吐槽几句)。(1)目前国内行业需求少(国内 CG/游戏尚无大作，渲染尚未运用得好，还不能谈什么物理模拟，在动起来像不像的问题上，总可以找到替代方案糊弄用户，因此没有资金和人员投入来解决)；(2)从事人员屈指可数(从事图形学的人都少，从事物理模拟的人更少)，国内的研究者应该可以数的出来，查下知网就知道国内找不出此类文章(悄悄说，有的文章写的真水，不举例了)，国外 SIGGRAPH 上此类文章则较多，感觉国内与国外的知识差距有了断层...

学科方向。物理模拟基本是个交叉学科，跟随计算力学的步伐走，基本上是有限元方法、无网格法，物质点法，近期有出现近场动力学(PeriDynamics)模拟断裂、破碎等。

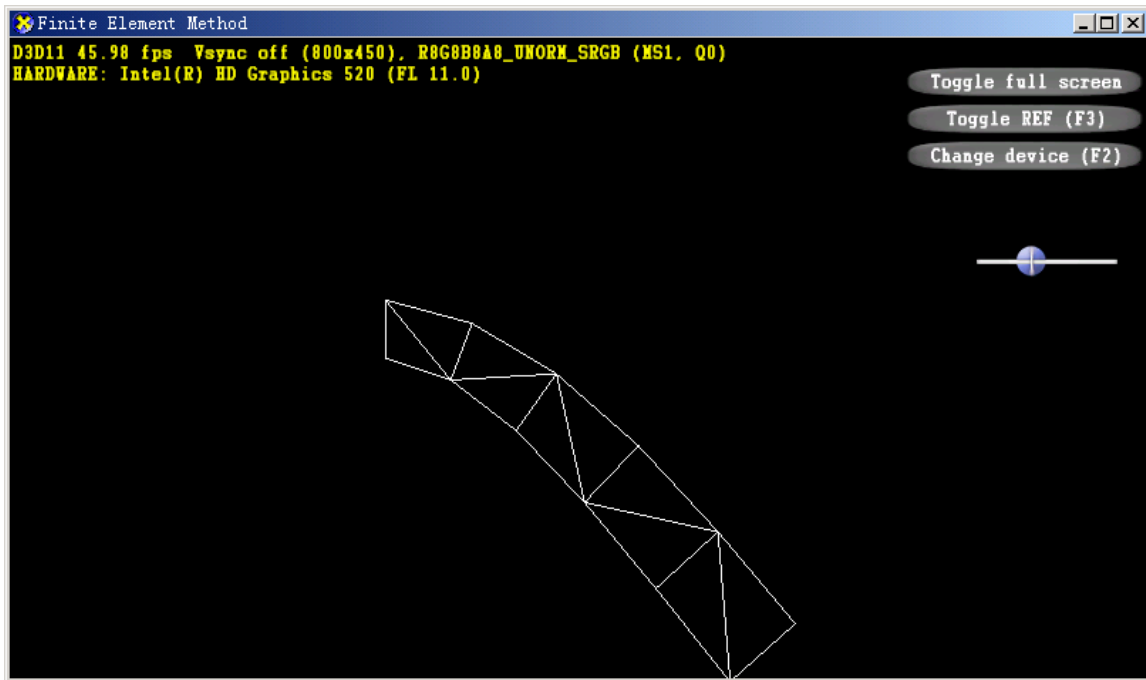
物理模拟的基本要求。按照 Robert Bridson(Fluid Simulation for Computer Graphics 作者)的说法，看起来像，求解起来稳定，算起来快。

物理模拟需要的数学基础。基本是数值线性代数的内容，阅读本节涉及的概念需要有限元、线性方程组的求解、SVD，极分解的知识。

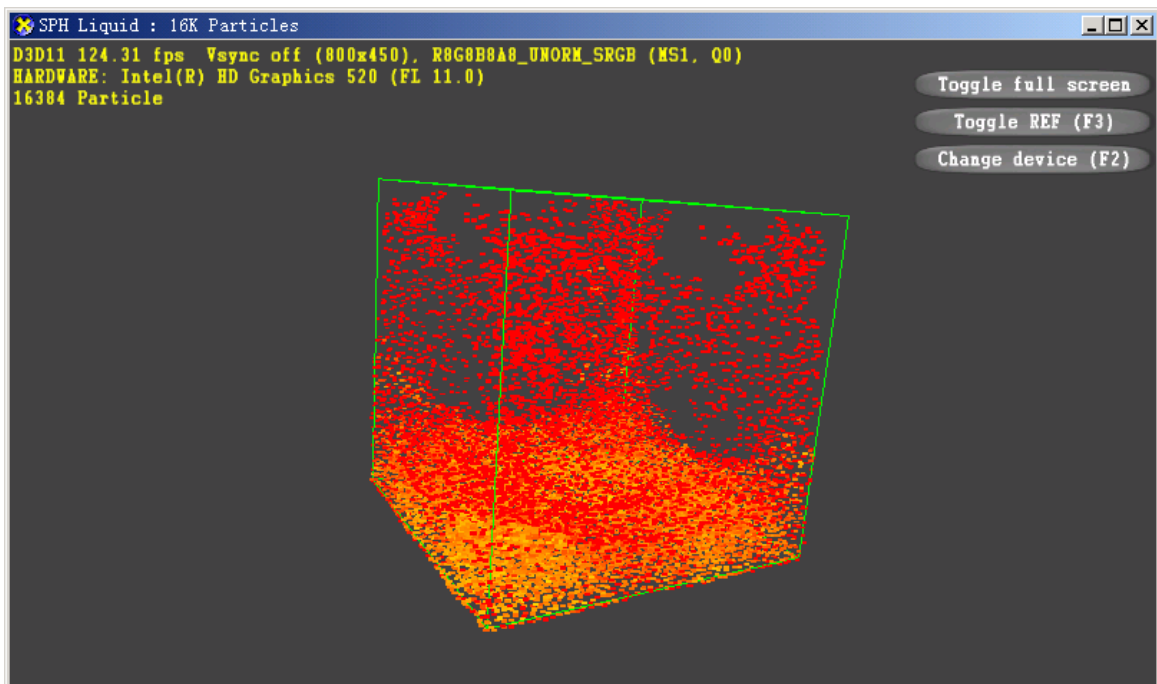
## 2. 模拟方法概述

由于物理模拟基本按照计算力学的方法进行，就简单对图形学中使用的方法给予简单概述。

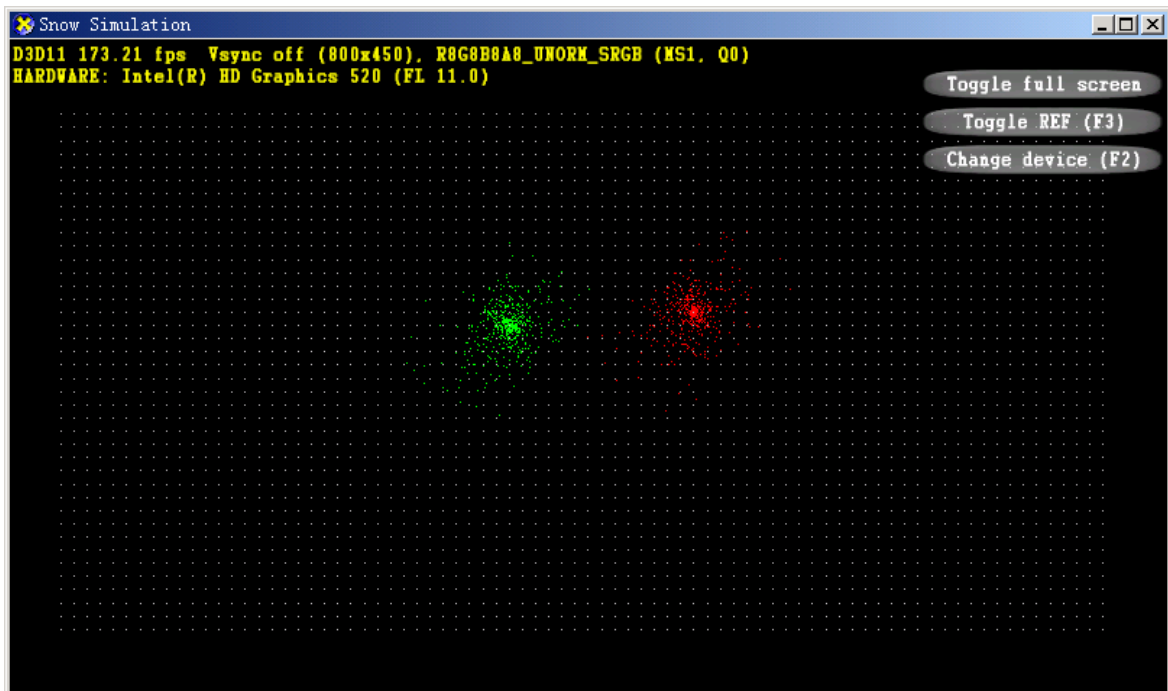
(1)有限元方法。有限元基本思想就是将物体离散成单元，每个单元满足其力学的本构方程(大致就是力与位移、弯曲与力矩等关系)。通俗讲就是将每个单元位移/转角与外作用力下的方程列出，而每个单元与其他相邻单元有相互作用，这个相互作用列出来就是一个线性方程组，从数值上求解这个方程组即可得到位移/转角。此方法广泛用于工程结构等领域，产品化的软件也多，但在图形学上模拟较大变形下有失真，见下图。一个桁架在向下的作用力下的大变形。



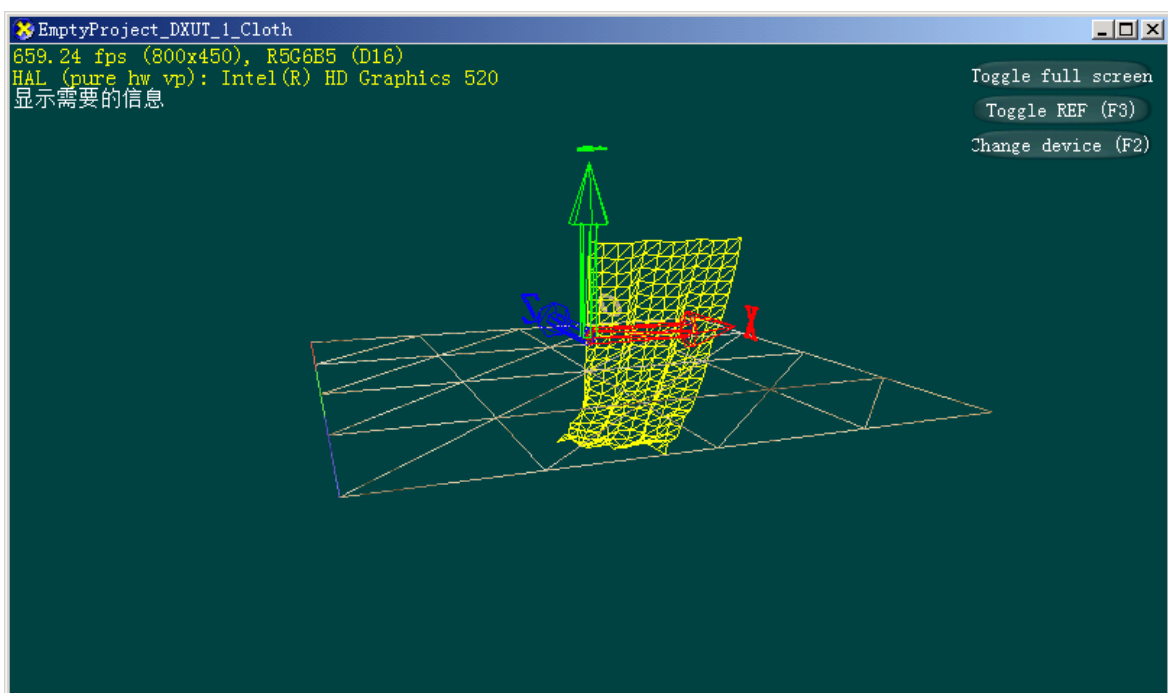
(2) 无网格法。无网格法基本思想是将物体离散成一个个小质点，通过计算每个小质点周围的密度来计算其受力，再通过此小质点上的作用力来计算此质点运动，从而模拟整个物体的情况。此方法在工程用于模拟高速碰撞的侵切和碰撞。在计算机的流体模拟中，使用此方法模拟流体飞溅达到很好的效果，其缺点是搜索周围的粒子很耗时(将空间划分成小网格的分治法是个不错的方法)。下图是使用 SPH 方法模拟流体。



(3) 物质点法。在流体模拟中有使用结拉格朗日视角和欧拉视角来描述物体的运动，但两种方法各有优劣。物质点法采用了两种方法的有点，即在计算时在两个视角中切换(方法的大致思路是将一种方法中不宜计算的物理量映射到另一个网格中，让两种方法的优势得到体现)，在计算力学中爆炸模拟等，见 <http://comdyn.hy.tsinghua.edu.cn/> 有详细的介绍。有文献比较了物质点法比无网格有优势。SIGGRAPH 也有文章介绍此方法模拟雪的各种效果。下图是两个雪球碰撞（但代码还未写完...，仅是未模拟前的效果）



(4) 图形学中 Muller 提出的基于位置的动力学(Position Based Dynamics)方法。大致思路是沿梯度方向在时间增加移动，通过约束函数来修正步长和方向。下图是使用此方法模拟布料的运动。



## 2. 有限元方法的局限性

上面讲解了使用有限元方法模拟物体的变形，并用代码实现了一个简单桁架结构在受到外部垂直力的作用下，各个杆件结构的变形效果，在实际的结构工程中，使用有限元方法在仅能模拟小变形下的应力和小变形(一般认为是 3%-5%)，但在大变形情况下失真较大，见下左图 1a。同样，在图形学中，Muller 也提供了一个失真严重的例子图 1b。

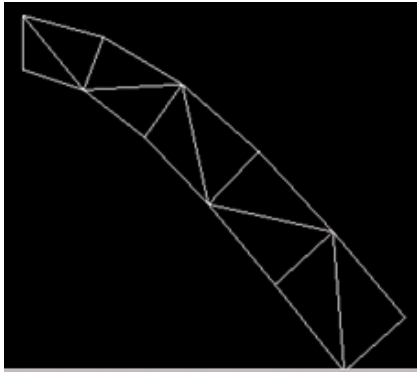


图 1a



图 1b

## 二、大变形的模拟方法

解决方法。为解决此失真，工程结构中使用共旋法（Corotational）来模拟薄壳的大变形，图形学中 Muller 使用了刚度弯曲法（Stiffness Warped）方法，并提供了展示效果。现在就对 Muller 提出的实现方法进行分析，逐步讲解计算思路。

大变形模拟的基本思想。在力学工程上，对于物体的大变形，一般使用非线性有限元来模拟，但由于非线性有限元在求解非线性方程组中计算量较大，且求解切线刚度矩阵的逆时计算困难，因此急需寻找一种快速有效的方法，其中共旋法有较大优势，与此类似，Muller 在图形学中提出的刚度弯曲法方法也与此相同，能同样也能模拟大变形。这两种方法的基本思想都是提取每个单元中的旋转部分，在刚度矩阵的计算时一并考虑旋转部分的影响，降低小变形大位移下失真。从上面的有限元计算方法看，桁架结构仅能计算轴向力，并未考虑每个杆件的旋转，因此在大变形的情况下必然存在失真现象。本小节的方法则将桁架中的每个三角形作为一个单元，在变形时考虑此三角形的旋转，这样建立的数学模型更能反映真实受力情况下的变形效果。

有限元方法是按照每个单元受力与位移的关系列出方程，计算其位移；在大变形中，每个单元（结点）会发生平移和旋转，刚度弯曲法的基本思想就是将每个单元（结点）先旋转回去，剩下的就是平移量，就可以使用有限元方法计算受力与位移的关系了。有限元方法计算结点位移如式(1)：

$$\mathbf{Ax} = \mathbf{b} \quad (1)$$

其中， $\mathbf{A}$ 表示总体刚度矩阵， $\mathbf{x}$ 表示各个结点的位移， $\mathbf{b}$ 表示各个结点所受外力，与有限元书籍中的表示不一样（一般有限元方法使用， $\mathbf{K}\boldsymbol{\epsilon} = \boldsymbol{\sigma}$ ，其中 $\mathbf{K}$ 表示刚度矩阵， $\boldsymbol{\sigma}$ 表示应力， $\boldsymbol{\epsilon}$ 表示应变）。刚度弯曲法的计算表达式为：

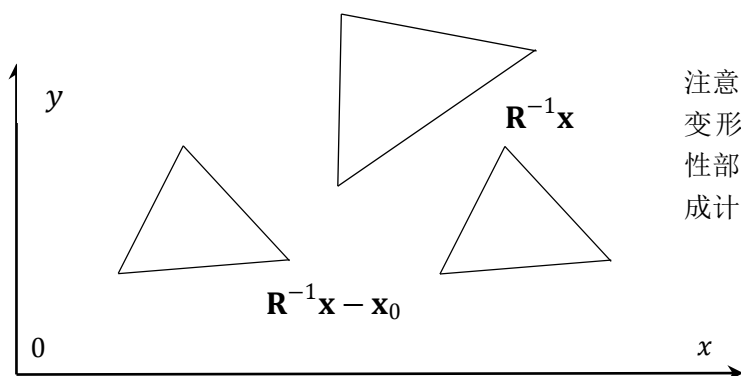
$$\mathbf{RA}(\mathbf{R}^{-1}\mathbf{x} - \mathbf{x}_0) = \mathbf{b} \quad (2)$$

(2) 式中 $\mathbf{R}$ 表示每个单元的旋转部分， $\mathbf{x}$ 表示当前位置， $\mathbf{x}_0$ 表示初始位置。等式(2)其表示的意义就是，将当前位置逆旋转 $\mathbf{R}^{-1}\mathbf{x}$ （目的是为了消除旋转的影响），与初始位置的差值 $(\mathbf{R}^{-1}\mathbf{x} - \mathbf{x}_0)$ ，即在线性方向的位移增量，在与刚度矩阵的乘积 $\mathbf{A}(\mathbf{R}^{-1}\mathbf{x} - \mathbf{x}_0)$ ，再旋转回去 $\mathbf{RA}(\mathbf{R}^{-1}\mathbf{x} - \mathbf{x}_0)$ ，就是外力 $\mathbf{b}$ 。其思想还是有限元的位移与应变的关系，只不过刚度矩阵发生了变化，这种变化是一个旋转矩阵，因此称为弯曲刚度。

从上面的原理可以分析，如果旋转矩阵为单位阵，则表达式

$$\mathbf{A}(\mathbf{x} - \mathbf{x}_0) = \mathbf{b} \quad (3)$$

就是有限元方法，因此弯曲刚度法就是有限元方法的扩展，有限元方法是弯曲刚度法的特例。



注意：逆旋转部分可以将变形变换到有限元的线性部分，再借助有限元完成计算。

使用刚度弯曲法方法的核心是求出每个有限单元的旋转信息，如何从变化前后求出变换矩阵，再从变换矩阵分解出旋转信息是这种方法的核心。现在的问题是，每个有限单元是否有变换矩阵？这个矩阵如果存在是否唯一，是否可以从变换矩阵中分解出旋转部分？

按照以上的思路具体讲解分析如下：几何图形经过矩阵后变形成为另一个形状，变换的方式有平移（即将图形沿轴向移动一定的距离），旋转（绕某固定轴旋转），缩放（以某固定点放大或缩小比例），错切（在某方向上，对图形的每个点到某条平行于该方向的直线的有向距离做放缩），这些单独的变换可以使用一个矩阵来描述。而在实际的变换中，可能这些矩阵已经相乘合并成一个矩阵，现在已知几何图形变换前后的信息（各个顶点的位置），能否计算出这个变换矩阵。从数学上说，就是通过变换前和变换后的顶点信息，能否找到这个变换矩阵，这个矩阵是否唯一，能否从这个矩阵中提取旋转、平移矩阵。此外，如果是多个有限单元（如两个有限单元组成的一个矩形），是否有一个矩阵将此矩形变换成另一个矩形（即求总体刚度矩阵的问题）。

为了将问题简单化，先考虑一个平面问题，为讲解叙述清晰，首先分析矩阵如何求解，其次分析如何分解。

### 三、变换矩阵

#### 1. 变换矩阵的形式

再将问题使用语言描述如下，一个平面三角形经过某个矩阵变换后到平面的另一个位置，现已知三角形在变形前后的坐标，是否能根据前后坐标关系反推变换矩阵。具体如下图 a 所示。

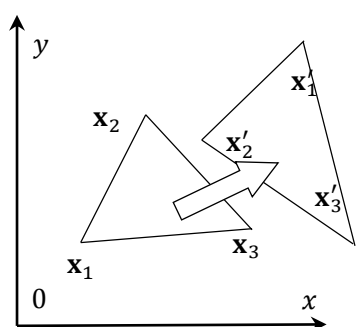


图 a

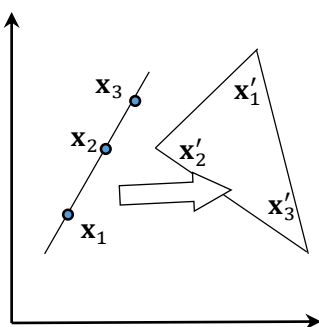


图 b

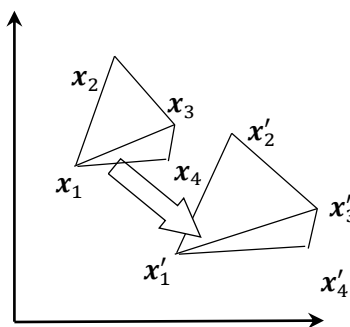


图 c

设三角形某个顶点的坐标为 $(x, y)$ ，使用向量表示成 $\mathbf{x}$ ，即 $\mathbf{x} = [x \ y]^T$ ，变换后的位置为 $\mathbf{x}' = [x' \ y']^T$ ，三个顶点分别为 $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ ，经过矩阵 $\mathbf{M}$ 变换后的坐标则为 $\mathbf{x}'_1, \mathbf{x}'_2, \mathbf{x}'_3$ ，可以使用数学表述为式(4)，如图

a 所示:

$$\begin{cases} a_{11}x + a_{12}y = x' \\ a_{21}x + a_{22}y = y' \end{cases} \quad (4)$$

使用矩阵方式描述如(5)式:

$$\mathbf{M} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad (5)$$

$$\mathbf{M}\mathbf{x} = \mathbf{x}'$$

以上变换只有旋转、缩放、错切, 为增加平移, 则需要增加一个平移项, 表达成如式(6):

$$\begin{cases} a_{11}x + a_{12}y + d_1 = x' \\ a_{21}x + a_{22}y + d_2 = y' \end{cases} \quad (6)$$

使用矩阵方式表达成式(7):

$$\mathbf{M}\mathbf{x} + \mathbf{d} = \mathbf{x}' \quad (7)$$

其中 $\mathbf{d} = [d_1 \ d_2]^T$ , 这样就描述了平面上任何一个顶点的变换形式, 但描述方式需要一个矩阵和一个向量。

是否存有一个矩阵能完整描述全部变换呢? 计算机图形学中使用了齐次坐标方式, 即增加一个维度来描述, 只不过增加的维度使用常量, 如 1, 具体如 $\mathbf{x} = [x \ y \ 1]^T$ , 变换后的坐标表示为 $\mathbf{x}' = [x' \ y' \ 1]^T$ , 则每个顶点的变换矩阵 $\mathbf{A}$ 则为式(8):

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad (8)$$

实际上矩阵中的元素 $a_{31}, a_{32}, a_{33}$ 也存在明确的几何意义, 分别表示 $x$ 轴在 $1/a_{31}$ 处的灭点,  $y$ 轴在 $1/a_{32}$ 处的灭点, 整体缩放的系数, 参见文献, 如果让灭点在无穷远处, 即平行投影,  $a_{31} = 0, a_{32} = 0$ , 且无缩放,  $a_{33} = 1$ , 则此变换为式(9):

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \quad (9)$$

则可以满足上述的要求, 读者可以自行验证变换与前面的方式一致。

那么三角形的任何一个顶点的变换表达式为

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11}x + a_{12}y + a_{13} \\ a_{21}x + a_{22}y + a_{23} \\ 1 \end{bmatrix} \quad (10)$$

可以看出, 使用齐次坐标方式描述顶点更加合理, 在将顶点增加一个维度的情况下, 描述顶点的变换更加全面, 矩阵的各个分块也有明确的几何意义, 其中 $\mathbf{A}$ 中的 $\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ 部分用于旋转, 错切, 拉伸等, 与前面的方式一致, 分块 $\begin{bmatrix} a_{13} \\ a_{23} \end{bmatrix}$ 表示平移部分。

有了以上描述变换的矩阵, 现在进一步思考, 有了变换前后的顶点坐标, 是否可以求出变换矩阵? 从上面的推导可以看出, 一个顶点是不能反过来退出变换矩阵, 两个顶点也不能, 只有三个顶点才能求解, 具体如下, 整个三角形的三个顶点 $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ 的变换后 $\mathbf{x}'_1, \mathbf{x}'_2, \mathbf{x}'_3$ , 可以表示成式(11)

$$\mathbf{x}'_1 = \begin{bmatrix} x'_1 \\ y'_1 \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11}x_1 + a_{12}y_1 + a_{13} \\ a_{21}x_1 + a_{22}y_1 + a_{23} \\ 1 \end{bmatrix} \quad (11)$$

$$\mathbf{x}'_1 = \begin{bmatrix} x'_1 \\ y'_1 \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11}x_1 + a_{12}y_1 + a_{13} \\ a_{21}x_1 + a_{22}y_1 + a_{23} \\ 1 \end{bmatrix}$$

$$\mathbf{x}'_3 = \begin{bmatrix} x'_3 \\ y'_3 \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_3 \\ y_3 \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11}x_3 + a_{12}y_3 + a_{13} \\ a_{21}x_3 + a_{22}y_3 + a_{23} \\ 1 \end{bmatrix}$$

有式(11)，要求出变换矩阵 $\mathbf{A}$ 的6个系数，那么将矩阵的系数作为未知量表达成式(12)：

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1 & y_1 & 1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_2 & y_2 & 1 \\ x_3 & y_3 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_3 & y_3 & 1 \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{21} \\ a_{22} \\ a_{23} \end{bmatrix} = \begin{bmatrix} x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \\ x'_3 \\ y'_3 \end{bmatrix} \quad (12)$$

书写成另一种形式（或者采用置换矩阵变换），如式(13)。

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 \\ x_2 & y_2 & 1 & 0 & 0 & 0 \\ x_3 & y_3 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1 & y_1 & 1 \\ 0 & 0 & 0 & x_2 & y_2 & 1 \\ 0 & 0 & 0 & x_3 & y_3 & 1 \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{21} \\ a_{22} \\ a_{23} \end{bmatrix} = \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \\ y'_1 \\ y'_2 \\ y'_3 \end{bmatrix} \quad (13)$$

简写为式(14)：

$$\mathbf{Pa} = \mathbf{y} \quad (14)$$

其中矩阵 $\mathbf{P}$ ，向量 $\mathbf{a}$ ， $\mathbf{y}$ 分别表示为：

$$\mathbf{P} = \begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 \\ x_2 & y_2 & 1 & 0 & 0 & 0 \\ x_3 & y_3 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1 & y_1 & 1 \\ 0 & 0 & 0 & x_2 & y_2 & 1 \\ 0 & 0 & 0 & x_3 & y_3 & 1 \end{bmatrix} \quad (15)$$

$$\mathbf{a} = [a_{11} \quad a_{12} \quad a_{13} \quad a_{21} \quad a_{22} \quad a_{23}]^T$$

$$\mathbf{y} = [x'_1 \quad x'_2 \quad x'_3 \quad y'_1 \quad y'_2 \quad y'_3]^T$$

到此为止，可以得到如下结论，可以根据三角形的三个顶点的前后信息来求解变换矩阵。但求解需要在什么条件下可以求解，即矩阵 $\mathbf{P}$ 非奇异的情况存在解。使用语言直观描述为：一个平面上的三角形（未退化）经过变换后，可以通过变换前后的三角形坐标值来反推这个变换矩阵。在几何上更直观的解释是。三角形的三个顶点在一条直线上，经过任意矩阵变换后，不可能通过变换后的顶点位置和变换前的顶点位置求出此变换矩阵（参见图 b），也就是说，只要是三角型（三个顶点不共线），就可以求出此变换。如果有一个平面四边形，在变形后，也不能通过其顶点来反推变换矩阵，因为矩阵奇异（更进一步描述，是矩阵欠定），则变换矩阵需要两个，更详细的分析见（3）。

## 2. 验证变换的正确性

现在使用 3 个小例子来验证这种变换的正确性。在上小节理解的情况下，此小节可以略过，具例子的目的在于方便理解。

例子 1。假设三角形未变换，即顶点坐标前后一致，则变换的方程是



$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (16)$$

按照已知条件列出的方程如下，即变换前点的坐标等于变换后点的坐标。

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 \\ x_2 & y_2 & 1 & 0 & 0 & 0 \\ x_3 & y_3 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1 & y_1 & 1 \\ 0 & 0 & 0 & x_2 & y_2 & 1 \\ 0 & 0 & 0 & x_3 & y_3 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} \quad (17)$$

可以，验证，只有 $a_{11} = 1$ ， $a_{22} = 1$ ，其余元素为0时，方程才成立，也就验证了矩阵 $\mathbf{A}$ 的数值。

例子 2。仅只有平移下的变换。变换矩阵为

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & a_{13} \\ 0 & 1 & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \quad (18)$$

则方程组如下，每个坐标点平移了相应的长度。

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 \\ x_2 & y_2 & 1 & 0 & 0 & 0 \\ x_3 & y_3 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1 & y_1 & 1 \\ 0 & 0 & 0 & x_2 & y_2 & 1 \\ 0 & 0 & 0 & x_3 & y_3 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ a_{13} \\ 0 \\ 1 \\ a_{23} \end{bmatrix} = \begin{bmatrix} x_1 + a_{13} \\ x_2 + a_{13} \\ x_3 + a_{13} \\ y_1 + a_{23} \\ y_2 + a_{23} \\ y_3 + a_{23} \end{bmatrix} \quad (19)$$

例子 3。三角形变形成直线，或者说是将三角形在某直线上的投影。此直线方程为 $y = kx + b$ 。则以上方程为

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 \\ x_2 & y_2 & 1 & 0 & 0 & 0 \\ x_3 & y_3 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1 & y_1 & 1 \\ 0 & 0 & 0 & x_2 & y_2 & 1 \\ 0 & 0 & 0 & x_3 & y_3 & 1 \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{21} \\ a_{22} \\ a_{23} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ kx_1 + b \\ kx_2 + b \\ kx_3 + b \end{bmatrix} \quad (20)$$

这样的变换存在，并且可以求出，但反过来就不能求出，因为矩阵奇异。这样的情况可以使用以下的投影方式来理解：如果将三角形的顶点投影到某个坐标轴（投影到 $x$ 轴上，则上式中的 $k = 0$ ），按此坐标轴上的顶点信息（变换后）是不可能恢复原来三角形的顶点信息的。

### 3. 简单四边形的变换矩阵

现在考虑平面上简单四边型（两个三角形共边）的情况，见图 c，按照上述的结论，则有两个矩阵分别将三角形的三个顶点 $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ 变换到 $\mathbf{x}'_1, \mathbf{x}'_2, \mathbf{x}'_3$ ，另一个矩阵将三角形的三个顶点 $\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_4$ 变换到 $\mathbf{x}'_1, \mathbf{x}'_3, \mathbf{x}'_4$ 。其中 $\mathbf{x}_1, \mathbf{x}_3$ 和 $\mathbf{x}'_1, \mathbf{x}'_3$ 是两个相邻三角形的公共点。现在按以上思路列出方程则，即变换第一个三角形的矩阵同上，则第二个三角形的变换矩阵如下，其中变换矩阵的系数为 $b$ 。

$$\begin{aligned} \mathbf{x}'_1 &= \begin{bmatrix} x'_1 \\ y'_1 \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11}x_1 + a_{12}y_1 + a_{13} \\ a_{21}x_1 + a_{22}y_1 + a_{23} \\ 1 \end{bmatrix} \\ \mathbf{x}'_3 &= \begin{bmatrix} x'_3 \\ y'_3 \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_3 \\ y_3 \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11}x_3 + a_{12}y_3 + a_{13} \\ a_{21}x_3 + a_{22}y_3 + a_{23} \\ 1 \end{bmatrix} \end{aligned} \quad (21)$$

$$\mathbf{x}'_4 = \begin{bmatrix} x'_4 \\ y'_4 \\ 1 \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_4 \\ y_4 \\ 1 \end{bmatrix} = \begin{bmatrix} b_{11}x_4 + b_{12}y_4 + b_{13} \\ b_{21}x_4 + b_{22}y_4 + b_{23} \\ 1 \end{bmatrix}$$

这样四个点的变换矩阵如下

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & a_{11} & a_{12} & a_{13} & 0 & 0 & 0 & 0 & 0 & 0 \\ & 0 & & a_{21} & a_{22} & a_{23} & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & 0 & 0 & 1 & a_{11} & a_{12} & a_{13} & 0 & 0 & 0 \\ & & & & & & a_{21} & a_{22} & a_{23} & 0 & 0 & 0 \\ & & & & & & 0 & 0 & 1 & b_{11} & b_{12} & 1 \\ & & & & & & & & & 0 & 0 & 1 \\ & & & & & & & & & 0 & 0 & 1 \\ & & & & & & & & & & & & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \\ x_2 \\ y_2 \\ 1 \\ x_3 \\ y_3 \\ 1 \\ x_4 \\ y_4 \\ 1 \end{bmatrix} = \begin{bmatrix} x'_1 \\ y'_1 \\ 1 \\ x'_2 \\ y'_2 \\ 1 \\ x'_3 \\ y'_3 \\ 1 \\ x'_4 \\ y'_4 \\ 1 \end{bmatrix} \quad (22)$$

注意，这样的变换矩阵不唯一。此结论可以从几何上得到结论，选择不同的三角形就对应不同的变换矩阵，因此取顶点 $\mathbf{x}_4, \mathbf{x}_2, \mathbf{x}_3$ ，再取一组顶点， $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ ，则得到的变换矩阵是不同的。

更多三角形的变换矩阵可以在两个三角形的情况下推广，因此，可以这么说，有一个总体刚度矩阵将多个三角形变换的另一位置。

#### 4. 三维空间矩阵情况

在三维空间的推导类似，也可以有以下结论，三维空间中的4面体，在经过变换后，如果已知变换前后的四面体的顶点坐标，同样可以找到此变换矩阵，只不过此矩阵为 $12 \times 12$ ，使用同样的方法求出矩阵的各个分量。

使用数学描述空间四面体的位置坐标 $\mathbf{x} = [x \ y \ z \ 1]^T$ 变换后的坐标 $= [x' \ y' \ z' \ 1]^T$ ，变换矩阵如下

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (23)$$

要求解的矩阵方程形式为

$$\begin{bmatrix} x_1 & y_1 & z_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ x_2 & y_2 & z_2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ x_3 & y_3 & z_3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ x_4 & y_4 & z_4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_1 & y_1 & z_1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_2 & y_2 & z_2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_3 & y_3 & z_3 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_4 & y_4 & z_4 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_1 & y_1 & z_1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_2 & y_2 & z_2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_3 & y_3 & z_3 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_4 & y_4 & z_4 & 1 \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{14} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{24} \\ a_{31} \\ a_{32} \\ a_{33} \\ a_{34} \end{bmatrix} = \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \\ x'_4 \\ y'_1 \\ y'_2 \\ y'_3 \\ y'_4 \\ z'_1 \\ z'_2 \\ z'_3 \\ z'_4 \end{bmatrix} \quad (24)$$

在求解 $4 \times 4$ 的矩阵时，依然可以使用克拉姆法则直接求解，无需解一个非对称的线性方程组，具体见代码。

## 四、极分解

### 1. SVD 的数学几何意义

为了清晰讲解极分解，先引入矩阵的奇异值分解（SVD）的概念。SVD 分解定义为，有矩阵  $\mathbf{A} \in \mathbf{R}^{m \times n}$ ，则  $\mathbf{A}$  可以表示成三个矩阵的乘积。

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \quad (25)$$

其中矩阵  $\mathbf{U} \in \mathbf{R}^{m \times n}$ ， $\mathbf{\Sigma} = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_n, 0, 0)$ ，且  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n$ ， $\mathbf{U} \in \mathbf{R}^{m \times m}$  和  $\mathbf{V}^T \in \mathbf{R}^{n \times n}$  为正交矩阵。在对角阵的元素为正的情况下，有唯一的分解。使用文字描述就是，有一个任意矩阵  $\mathbf{A}$ （行和列的数不一定相等），可以表示成三个矩阵的乘积，其中是对角阵  $\mathbf{\Sigma}$ （存在部分元素为 0），和两个正交阵。

SVD 的物理意义上可以这样理解，任何变换可以使用两个不同的旋转变换和一个缩放变换完成，在几何上可以这样理解，有一矩阵  $\mathbf{A}$ ，可以按以下步骤变换后与原矩阵等效，先旋转，即  $\mathbf{V}^T$ ，然后再缩放  $\mathbf{\Sigma}$ ，最后再旋转  $\mathbf{U}$ 。其特例就是对称矩阵的谱分解，即，当这个要变换的矩阵  $\mathbf{A}$  对称正定时，由对称则有  $\mathbf{U} = \mathbf{V}$ ，正定则有  $\mathbf{\Sigma}$  就是特征值， $\mathbf{U}$  就是特征向量组成的矩阵。换种方式讲解，SVD 也可以理解为将任意矩阵对角化的一种方式，只不过这种变化不能使用一组基实现，需要两组不同的基。按照上面的旋转要求，可以方便找到这个矩阵的旋转部分和非旋转部分，且非旋转部分是一个对称正定矩阵。

二维平面的数值实例。现以二维平面图形举例说明，由缩放、旋转、错切组成的矩阵，经过 SVD 分解后可以表示成旋转和缩放。下面是一个错切变换的矩阵，可以分解为旋转和缩放，其中  $\mathbf{x}$  表示某个平面顶点坐标。

$$\mathbf{A}\mathbf{x} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \mathbf{x} = \begin{bmatrix} 0.8506 & 0.5257 \\ 0.5257 & -0.8506 \end{bmatrix} \begin{bmatrix} 1.618 & 0 \\ 0 & 0.618 \end{bmatrix} \begin{bmatrix} 0.5257 & 0.8506 \\ 0.8506 & -0.5257 \end{bmatrix} \mathbf{x} \quad (26)$$

式 (26) 数值实例可以看出，带有错切变换矩阵变换某个图形时，可以先旋转，再按此主轴缩放，再旋转，最后与原变换一致。因此使用 SVD 来求旋转相当直观。

有了 SVD 工具，现在再来讨论顶点坐标使用齐次表达时是否具备以上特性。假设顶点变换的矩阵为

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \quad (27)$$

矩阵  $\mathbf{A}$  可以分解为平移和非平移部分的乘积（旋转和错切）：

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & a_{13} \\ 0 & 1 & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (28)$$

现在再对非平移部分进行 SVD 分级（这里没有给出具体数值实例，使用变量表达），下面的 4 个连乘的矩阵分别表示平移，旋转，沿主轴缩放，旋转。

$$\begin{bmatrix} 1 & 0 & a_{13} \\ 0 & 1 & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & a_{13} \\ 0 & 1 & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & 0 \\ u_{21} & u_{22} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_{11} & v_{12} & 0 \\ v_{21} & v_{22} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (29)$$

可以看出，在顶点使用齐次方式表达的变换矩阵的情况下，使用 SVD 同样能具同样的特性。可以将任意变换矩阵分解成平移部分和旋转和缩放。

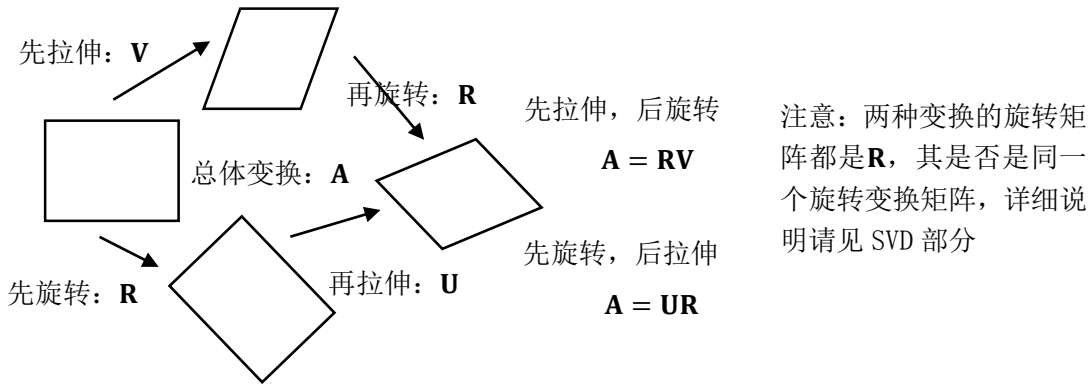
如果对平移矩阵直接进行 SVD 分解，如

$$\begin{bmatrix} 1 & 0 & 6 \\ 0 & 1 & 7 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -0.647 & 0.759 & 0.069 \\ -0.754 & -0.65 & 0.081 \\ -0.106 & 0.0 & -0.99 \end{bmatrix} \begin{bmatrix} 9.3268 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.107 \end{bmatrix} \begin{bmatrix} -0.06 & -0.08 & -0.994 \\ 0.759 & -0.651 & 0 \\ 0.647 & 0.754 & -0.106 \end{bmatrix} \quad (30)$$

则可以这样理解，平面上平移，可以在三维空间上旋转和缩放，与二维空间上的平移等效。当然这不在我们讨论的范围。

## 2. 极分解的概念和唯一性

在求解出变换矩阵后，可以使用矩阵的 SVD 分解提取出旋转部分。现在就使用 SVD 分解的方法来讲解极分解，下面以一个平面图形的变换过程来描述。在下图中，矩形（最左）经过某个矩阵变换成四边形（最右方），可以有两种方式，（1）先旋转（使用矩阵  $\mathbf{R}$  变换）再拉伸（使用矩阵  $\mathbf{U}$  变换），变换的中间结果在图的最下方；（2）或先拉伸（使用矩阵  $\mathbf{V}$  变换）再旋转（经过矩阵  $\mathbf{R}$  变换），变换的中间结果在图的最上方，最后的变换结果相同，矩阵上描述为一个矩阵可以由左极分解和右极分解，见下图。



以上两种变换方式在数学上描述就是旋转矩阵左乘或右乘。可能存在这样的疑问，上面的两种变换中，旋转矩阵是否都是一个？回答是，这两个旋转矩阵是唯一的，原因为，现在将矩阵表示成 SVD 分解方式，再表示成两种乘积方式。

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^{-1}\mathbf{U}\mathbf{V}^T = \mathbf{U}\mathbf{V}^{-1}\mathbf{V}\mathbf{\Sigma}\mathbf{V}^T \quad (31)$$

将其重新书写表达成如下形式，就是所说的左极分解和右极分解，具体如下：

$$\begin{aligned} \mathbf{A} &= (\mathbf{U}\mathbf{\Sigma}\mathbf{U}^{-1})(\mathbf{U}\mathbf{V}^T) \\ \mathbf{A} &= (\mathbf{U}\mathbf{V}^{-1})(\mathbf{V}\mathbf{\Sigma}\mathbf{V}^T) \end{aligned} \quad (32)$$

其中表示旋转的部分是正交矩阵  $\mathbf{U}\mathbf{V}^T$  和  $\mathbf{U}\mathbf{V}^{-1}$  矩阵，表示非旋转部分的是  $\mathbf{U}\mathbf{\Sigma}\mathbf{U}^{-1}$  和  $\mathbf{V}\mathbf{\Sigma}\mathbf{V}^T$  矩阵。其中矩阵  $\mathbf{\Sigma}$  表示沿正方向的缩放。从上面的得到的旋转矩阵可以看到，旋转部分都是  $\mathbf{U}\mathbf{V}^T$ ，即两种方式得到的旋转矩阵是一致的。

## 3. 极分解的数值方法

使用 SVD 来实现极分解计算量较大，具体的数值方法可以参见本书的数值线性代数部分。而实际上的极分解可以使用迭代法，但缺点是需要求解矩阵的逆，但在  $3 \times 3$  的矩阵中，求逆方便，相对 SVD 更快。

对三维空间，数值方法如下：

(1) 下面的矩阵  $\mathbf{A}$  是要进行极分解的矩阵：

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad (33)$$

(2) 再按下面的方式计算出一个新的矩阵  $\mathbf{R}$ ，再将此矩阵作为矩阵  $\mathbf{A}$ ，按下式计算，反复迭代。

$$\mathbf{R} = \frac{1}{2}(\mathbf{A} + \mathbf{A}^{-T}) \quad (34)$$

(3) 以上迭代法按以下方式计算，作为收敛准则，对矩阵 $\mathbf{E}$ 求范数，小于某阈值即可

$$\mathbf{E} = \frac{1}{2}(\mathbf{R}^T \mathbf{R} - \mathbf{I}) \quad (35)$$

(4) 在完成计算后，可以计算出左极分解和右极分解，有：

$$\begin{aligned} \mathbf{U} &= \mathbf{R}^T \mathbf{F} \\ \mathbf{V} &= \mathbf{F}^T \mathbf{R} \end{aligned} \quad (36)$$

(5) 在实际的代码中，需在代码中增加验证矩阵 $\mathbf{U}$ ， $\mathbf{V}$ 是否正交对称。

对于平面二维的情况，如果矩阵非奇异，有更简单的解析方式，具体计算步骤如下：

(1) 要进行极分解的矩阵 $\mathbf{A}$ 如下：

$$\mathbf{A} = \begin{bmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{bmatrix} \quad (37)$$

(2) 对矩阵 $\mathbf{A}$ 的元素计算两个变量

$$\begin{aligned} a &= f_{11} + f_{22} \\ b &= f_{21} - f_{12} \\ c &= a(a^2 + b^2)^{1/2} \\ s &= b(a^2 + b^2)^{1/2} \end{aligned} \quad (38)$$

(3) 按两个元素构造极分解矩阵

$$\mathbf{R} = \begin{bmatrix} c & -s \\ s & c \end{bmatrix} \quad (39)$$

## 4. 数值实例

按照上面的算法，对三阶矩阵数值验证情况如下，求解出逆阵的转置 $\mathbf{A}^{-T}$ ，计算结果如下

$$\mathbf{A} = \begin{bmatrix} 8.442 & 0.559 & 2.790 \\ 2.771 & 1.936 & 4.589 \\ 0.818 & 6.158 & 5.564 \end{bmatrix}, \mathbf{A}^{-T} = \begin{bmatrix} 0.157 & -0.126 & 0.025 \\ 0.105 & -0.402 & 0.279 \\ -0.139 & 0.464 & -0.133 \end{bmatrix}, \text{经 7 次迭代, 得到旋转矩阵}$$

$$\mathbf{R} = \begin{bmatrix} 0.998 & 0.058 & -0.021 \\ 0.039 & -0.332 & 0.942 \\ -0.047 & 0.941 & 0.334 \end{bmatrix}$$

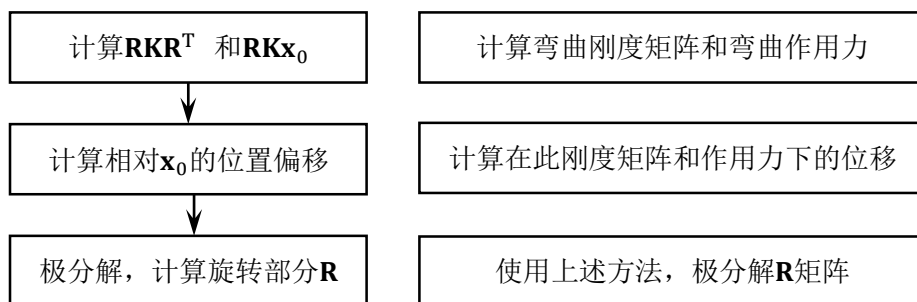
按照上式求出矩阵 $\mathbf{U}$ 和 $\mathbf{V}$ ，

$$\mathbf{U} = \begin{bmatrix} 8.497 & 0.341 & 2.702 \\ 0.341 & 5.184 & 3.872 \\ 2.702 & 3.872 & 6.124 \end{bmatrix}, \mathbf{V} = \begin{bmatrix} 8.398 & 2.779 & 1.056 \\ 2.779 & 3.790 & 3.225 \\ 1.056 & 3.225 & 7.618 \end{bmatrix}$$

更高阶的矩阵逆计算较耗时，不在此讨论。

## 五、代码的简单讲解

按照完全刚度法的计算思路，核心原理和计算流程如下：



## 六、核心代码的讲解

### 1. 程序的模块

代码总体概述。程序共有 6 个模块。各模块的功能为：CBeam.cpp 实现渲染和关键计算，CMat2x2.cpp 实现二维极分解；Matrix.cpp 实现矩阵与向量的定义；MatrixCaculate.cpp 实现矩阵之间，矩阵与向量的计算；Slover.cpp 实现线性方程组的求解，EmptyProject11.cpp 是主程序的入口。

新增核心模块。新增 CMat2x2 实现矩阵的极分解。定义了一个 2x2 矩阵的数据结构和 5 个函数。分别实现转置，求逆，乘积和极分解。

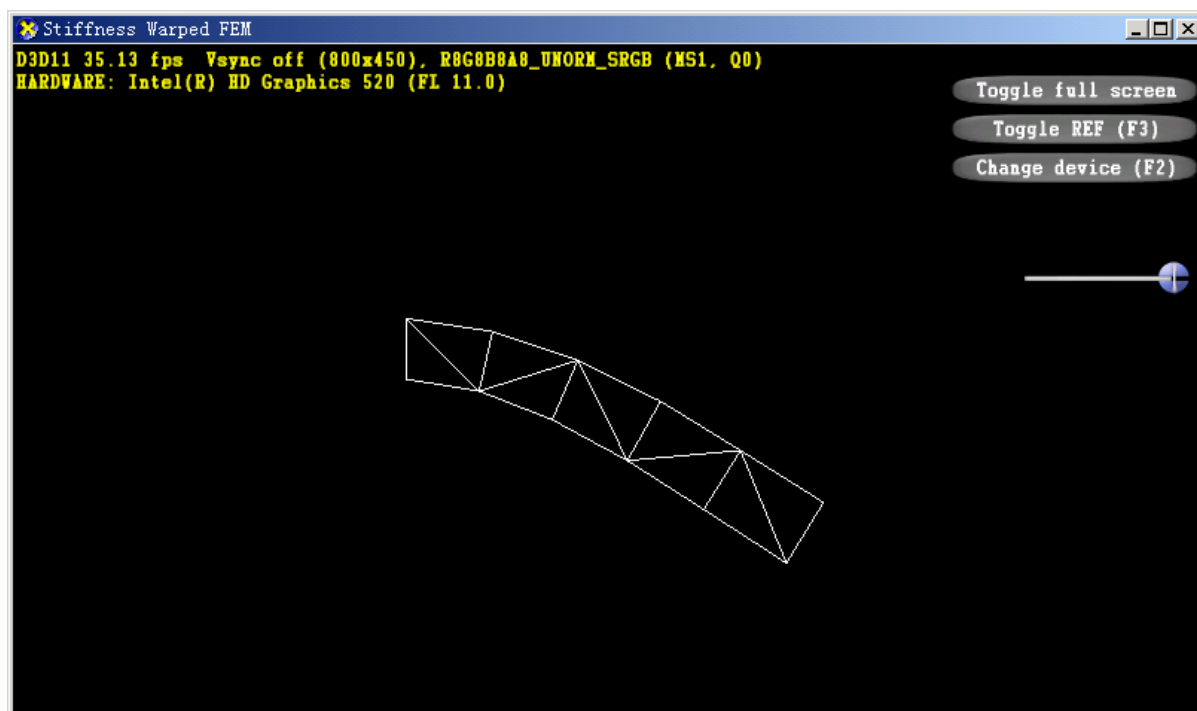
### 2. 核心模块的讲解

程序的核心在 CBeam.cpp 的 OnFrameMove() 中，实现以上原理的核心执行过程，实现了弯曲刚度矩阵的功能，具体如下

```

void CALLBACK CBEAM::OnFrameMove( double fTime , float fElapsedTime , float fForce )
{
    // 计算第一个框图中的变量: m_cMatRKRT = cMatR * cMatK * cMarR^(T), cVecf_t = cMatRK * cVecf
    // 矩阵和向量的维数为 24 维
    if( false == MatrixMultiplyMatrix( &m_cMatRK , &m_cMatR , &m_cMatK )) return ;
    if( false == MatrixMultiplyTransposeMatrix( &m_cMatRKRT , &m_cMatRK , &m_cMatR )) return ;
    if( false == MatrixMultiplyVector( &m_cVecf_t , &m_cMatRK , &m_cVecf )) return ;
    // 将上面计算的矩阵和作用力的维数降维 20, 矩阵和向量为 20* 20
    if( false == MatRKRT2MatA( &m_cMatA , &m_cMatRKRT )) return ; // 矩阵元素重新赋值
    if( false == Vecf2Vecb( &m_cVecb , &m_cVecf , m_cVecPosInit )) return ;
    // 将更新刚度矩阵和作用力传递给求解器, 矩阵和向量 20 * 20
    if( false == m_cSlover.UpdateMatrix( &m_cMatA )) return ;
    if( false == m_cSlover.Updateb( &m_cVecb )) return;
    // 求解线性方程组
    if( false == m_cSlover.Slover( &m_cVecx )) return;
    // 将计算结果升维到 24
    if( false == Vecx2VecPos( &m_cVecPos , &m_cVecx )) return ;
    // 计算总体变换矩阵
    if( false == SloveMatP( &m_cMatP , &m_cVecPosInit , &m_cVecPos )) return ;
    // 对总体变换矩阵进行极分解, 求出旋转矩阵
    if( false == MatrixPolarDecomposition( &m_cMatR , &m_cMatP )) return ;
}
  
```

### 3. 执行效果



## 七、总结和后续扩展阅读

刚度矩阵弯曲法在物体发生较大变形的情况下，能有效解决失真效果，但在变形更大的情况下仍有局限性，如何在此方法上改进是一个方向。对于物体更大变形情况下的物理模拟，如流体的飞溅，高速物体侵切可以使用无网格法来求解计算，在塑性变形，如橡胶，再考虑物态发生变化的情况，如堆积的雪发生融化，流体的凝固、固体的融化等方面的模拟，目前有物质点法模拟。

主要参考文献[综述部分涉及文献较多，未完全列入]

[1]Interactive Virtual Materials

[2]Real Time Physics Class Notes

[2]A Robust Method to Extract the Rotational Part of Deformations

[3]Matrix Computations

[4]物质点法

[5]数值线性代数

[6]无网格法