

硕 士 研 究 生 读 书 报 告



题目 转换式弹出书的计算设计报告

作者姓名 王昊阳

作者学号 21851411

指导教师 张志猛

学科专业 软件工程

所在学院 软件学院

提交日期 二○一八 年 十二 月

Report for Computational Design of Transforming Pop-up Books

A Dissertation Submitted to

Zhejiang University

in partial fulfillment of the requirements for

the degree of

Master of Engineering

Major Subject: Software Engineering

Advisor: Zhi-meng Zhang

By

Hao-yang Wang

Zhejiang University, P.R. China

2018

摘要

本文重点探讨了可以帮助普通用户制作转换式弹出书的计算工具。弹出书是指：当用户打开书，一个二维的图案就会从书里弹出。转换式弹出是指：当用户拉动拉环，一个初始的平面二维图案转换为一个新的二维图案，例如一家飞机转换为一个机器人。给定两个二维图案，本文探讨了一个自动化计算转换它们的三维弹出机制，还能输出一份让用户轻易制作最终模型的设计蓝图的方法。本文还解释了基础转换机制的理论研究。

**关键词**：转换式弹出， 计算机图形学， 遗传算法

Abstract

The paper discuss the computational tool to help ordinary users create transforming pop-up books. Transforming pop-up books are the ones that people could see a 2D pattern pop up from the page when it gets opened. In each transforming pop-up, when user pulls a tab, an initial flat 2D pattern turns into a new 2D patterns, for example, an airplane turns into a robot. Given the two 2D patterns, the paper discuss the an approach automatically computes a 3D pop-up mechanism that transforms one pattern into the other and also outputs a design blueprint that allows the user to easily make the final model. The paper also explain a theoretical analysis of basic transformation mechanisms.

**Keywords：**transforming pop-up, computer graphics, genetic algorithm

1引言

今天大多数人都玩过变形玩具，它们通过各种机制的结合来改变形状，例如从飞机到机器人。早些年的人们可能对弹出书更熟悉——当它们被打开，城堡或动物等就会从书页上弹出。而早些时候，这两个想法在一些出版物中被结合在一起，例如Reinhart设计的含有各种精致转换式弹出的畅销书。当用户拉动拉环，一个有纹理质地的初始二维图案就转换成一个新的二维图案。

设计这样一本书对于经验丰富的书页艺术工程师都是个难题。一位有才华的专家也需要经过数周的尝试和失败才能创造出有这样效果的机制。

研究者们已经考虑过自动化弹出书的设计过程。但是，至今为止弹出的种类只考虑描述一种有意义的对象：一个随意的平面设计转变为需要的对象。相比之下，转换式弹出描述两个不同的有意义的对象并由一个转换相连。因此，现有的自动化弹出设计方法不能直接用于转换式弹出设计。

**2 转换式弹出**

要解决的问题如下。图案是有质地的二维形状。给定两个图案作为输入，我们想要计算出一个能够平滑地在三维模式下转换这两个图案的机制。我们还想自动地生成一个蓝图，以允许用户通过粘贴和装配打印好的纸片来制作弹出。先考虑一些理论背景。

转换式弹出的关键要素是块，折叶和裂缝：

块是一个平面多边形，物理上是一片两边可以被涂画上不同质地的纸。一些块可以被结合起来描绘一个图案。

折叶是两个块联结并绕另一个旋转的共轴。

裂缝是一个块内部宽度极小的直线切割。另一个块可以通过这样一条裂缝。

文章视纸为可以围绕折叶旋转的刚体材料，并为简化，假设它没有厚度和重量。

从机械学上的观点来看，一个转换式弹出由一些我们可统称为基本元素的基本机制和其他功能性元素组成。这些基本元素通过少量预先定义好种类的联结来连接。当用户拉动或推动拉环，弹出通过一个初始平面二维图案通过三维的转换变为新的二维图案。转换由拉环驱动并且通过联结转换整个弹出。我们分别地把初始状态和最终状态视为未转换状态和转换后状态。

2.1 基本机制

我们用基于4R联结的三种基本机制。4R联结是一个有4个折叶和1个自由度的基本闭环机制。三种机制如下介绍。

平行机制。平行机制(PM)是由4个四边形块通过4个折叶循环连接组成的。在一个PM中，全部4个折叶都是平行的，而片有两对成对平行。

V折机制。V折机制(VM)是由4个块和4个折叶组成的。4个折叶的延长线交汇于一个点，即VM的椎体顶点。用αij代表折叶hi和hj之间的夹角，必须满足α14=α23和α12=α34。注意椎体顶点不需要实际上属于任何一个块。

滑落机制。滑落机制(SM)是由1个矩形块作为滑落块，1个折叶和1条裂缝组成的；它联合一个PM或者VM。裂缝位于联合的机制中的一个块上，且折叶位于同一机制的另一个块上。滑落块会通过这条裂缝。SM必须满足一些约束。第一，裂缝宽度要超过转换中通过裂缝的滑落块的最大宽度。第二，滑落块的长度应该比裂缝和折叶在转换中最大的距离更长。第三，当与PM联合时，SM的裂缝和折叶都要同PM的折叶平行；当与VM联合时，裂缝和折叶的延伸线都必须交汇在VM的椎体顶点。第一个约束保证了滑落机制不堵塞，第二个保证了滑落块不掉出裂缝，第三个保证了折叶和裂缝共面。滑落机制允许滑落块从暴露到隐藏或反之。

2.2 功能性元素

除了以上基本机制以外，文章还用到了另外一些功能性元素。

扩展块。如定义，一个PM或VM的块被限制为四边形或三角形，这限制了弹出可以表示的形状的范围。为了达到更高的灵活性，我们允许进一步的多边形在同一平面依附在一个PM或VM的块上。这样一种增强的小块被看作为扩展块(EP)。在实现中，作者限制扩展块有六个或更少的顶点来限制转换机制最优化中的参数个数。

基础块和拉环。转换式弹出中需要一个基础块和一个拉环来正确工作。基础块在转换中保持静态；它也作为参考平面工作。我们从PM和VM中选定一个根机制，并且之后基础块从属于根机制的块中选择一个。基础块可以为更好的美学标准延伸并一般形成该书一页上很大的一部分。拉环是一个矩形块，用户拉动或推动它来驱使弹出的转换。

2.3 联结树

总体的转换式弹出式通过连接基本元素构建的。转换是通过拉环驱使的，转换的移动是通过使整个弹出移动得联结来将一个元素转换到另一个。注意联结是定向的，即一个主动元素驱动另一个被动元素。接下来介绍允许的联结种类。

滑落联结。联结可能从一个PM或VM去向一个SM。移动通过共享的折叶和裂缝传动。

扩展块联结。联结可能从一个PM或VM去向一个EP。移动通过共享的平面传动。

折叶联结。联结可能从一个PM或VM去向另一个PM或VM。两个这样连接的基本机制共享一条折叶线，以及一对相邻的片被限制为共面的。注意这两个基本机制不必要在折页上有共同的结束点。根据两个基本机制的相对位置，折叶联结可以再细分为三种：对立折叶联结、内部折叶联结和增补折叶联结。

联结树。一个转换式弹出的复合机制形成了一棵联结树，它每个节点代表一个基本机制或一个功能性元素，且每条边代表一个联结。连接拉环的基本机制是根节点。EP和SM只能作为叶节点，同时PM和VM可以是叶节点或内部节点。拉环和基础小块不在联结树中被表示。

2.4 参数空间

以上想法描述了制作一个弹出可能的组分和它们的连接。但是，为了完全决定一个转换式弹出，我们还需要找到各种几何学参数的值，它们决定了块顶点、折叶、裂缝以及其他部件的位置。尽管这些位置在转换中会改变，在未转换状态下描述它们就足够完全决定一个弹出了。这些参数通过使用最优化算法调整来实现尽可能近地满足设计者目标的机制。我们首先考虑在基础平面如何定义这些参数。

PM，有4个折叶和8个顶点。为在二维基础平面定义这4条折叶线，我们需要2个参数来定义第一条线以及另外2个参数来定义第二和第三条线(与第一条线的距离)；第四条线被另外三条决定。由于8个顶点位于4条折叶线上，需要增加8个参数来定义它们。总体上，一个PM的参数空间是12维的。

VM，有1个椎体顶点，4个折叶和8个顶点。2个参数足够定义椎体顶点。因为4条折叶线交汇于椎体顶点，需要3个参数定义第一、二、三条线；第四条线被另外三条决定。同样需要8个参数定义8个顶点。总体上，一个VM的参数空间是13维的。

SM，有1个折叶，1条裂缝和1个滑落块。一个SM与一个PM或VM联合并受到约束所以折叶线和裂缝线平行于PM的折叶线或交汇于VM的椎体顶点。这些约束意味着需要2个参数来定义折叶线和裂缝线(存储与PM折叶的距离或与VM椎体顶点的角度)。我们还需要2个顶点来定义折叶线的2个端点。因为滑落块是矩形的，需要1个参数定义它的长度。我们不需要参数定义裂缝的两个端点，因为它们被滑落快在转换中的移动隐含地定义了。总体上，一个SM的参数空间是5维的。

最后，一个扩展块的参数空间最多为12维，因为它最多有6个顶点，每个二维。

整个弹出的总体几何学参数通过联系它每个基本机制和扩展块的几何学参数获得。但是，必须注意折叶联结。因为连接的两机制共享一条折叶线，我们不需要在两个机制中存储这条折叶线。对于所有的三种折叶联结类型，在两个基本机制之间添加一个联结将减少参数空间的两维。

概括：一个转换式弹出被它的联结树、组分和它们的连接以及以上描述的几何学参数完全地决定。

**3 算法**

3.1 总览

一个转换式弹出从未转换状态到转换后状态；每个状态都是一个二维图案。作者的目标是生成一个在未转换状态显得像用户给定的源图案并在转换后状态显得像用户给定的目标图案的弹出。在转换中，我们必须保证机制保持无碰撞的硬约束。生成的转换式弹出也应该满足两个软约束：一个形状约束和一个翻转约束。第一个是弹出在未转换和转换后状态的形状应该尽可能地近似匹配源图案和目标图案。第二个是在未转换和转换后状态都可见的任何块或其部分都应该在转换中翻转。如果它们的正面在未转换状态显示，那么它们的背面应该在转换后状态显示。这允许块的可见部分在未转换和转换后状态下分别用源图案和目标图案的质地不同地涂画。

总的算法分成四步：最优化、块提纯、质地涂画和蓝图生成。

最优化生成一个初始的弹出。一个能量函数对形状约束和翻转约束以及保证无碰撞的硬约束建模。为限制联结树的配置数量在合理界限，作者限制基本机制和扩展块的最大可用数量。通过考虑所有联结树的可能配置，并为每种配置最优化几何学参数来获得能量最少的弹出。由于能量函数非线性并且参数空间高维度，作者使用了一个遗传算法。

正如转换式弹出中块只含有较少顶点的多边形，最优化步骤限制弹出只能表示简单大致的形状。为了给图案确定一个更精细的形状，作者用到一个提纯步骤。在这个步骤，一些块被修剪到更加匹配用户给定的图案形状。由于只修剪块并且不给弹出添加任何新材料，机制结构和约束保持了原状——只要不修剪过多。

在修剪后，视源图案和目标图案的需要，往未转换和转换后状态的弹出的块上涂画质地。

最后，为用户输出一个构建转换式弹出的蓝图。将弹出的所有部分加上装配合适依附的标签打印到纸上。用户可以通过裁剪这些部分以及把它们粘贴在一起来制作弹出。

3.2 最优化

3.2.1 能量函数

根据最小化一个能量函数E(x)，x表示转换式弹出联结的配置和几何学参数，来定义最优化问题。

保证无碰撞被实施为一个硬约束。给定一个配置x，先检查转换中的碰撞，并且如果检测到，给能量函数E一个肯定比任何无碰撞配置更大的能量值。实现用了OBB树算法来检测碰撞，就像其他最近的机械的模型工作一样。为了避免与基础面的碰撞，设计者可以选择性地包含一个假设的足够大面积的地面小块。如果目标是将弹出放在一本书里，那这个选项应该被启用，但如果是一个独立式的弹出，这个选项是不必要的。

如果没有碰撞，那么能量函数E被定义为形状项和翻转项的和：

E = λsEs + (1 – λs)Ef.

形状项Es代表形状约束，偏好未转换和转后状态良好匹配用户给定图案的弹出形状。翻转项Ef代表翻转约束，惩罚任何在未转换和转后状态可见但不翻转的块区域。λs控制这两项的相关权值并设为0.4。

形状项。形状项解释了弹出在未转换和转换后状态S1和S2匹配用户给定的源图案和目标图案U1和U2的形状有多好。形状匹配代价包括一个轮廓匹配代价和一个区域匹配代价：

Es = Ec + Ea.

该式使用shape context方法来计算轮廓匹配代价并被定义为：

Ec = D(S1,U1) + D(S2,U2).

其中D(.)代表shanpe context匹配代价。在视线中，在每个轮廓上抽样100分来进行这个计算。

区域匹配代价度量了弹出和用户给定图案间区间覆盖的不同。它被定义为：

Ea =

其中\是集合差运算符；||·||代表了一个区域的部分区域；λa是一个权重。Si\Ui是冗余的区域。冗余区域相对的不重要，因为之后执行的块提纯步骤会修剪它们。但是，没有被表示的区域将展示基础块而它们不会成为转换式弹出的一部分。因此，设置λa=0.1来更强地惩罚未表示的区域。

翻转项。由于可见区域在未转换和转换后状态分别地被涂画上源图案和目标图案的质地，为避免质地涂画上的矛盾，应该避免区域在未转换和转换后状态中展示纸的同一侧。翻转项被简单定义为所有区域的违反约束的子区域总和：

.

其中Ri是一个块的任何在两个状态下同一面可见的区域。

3.2.2 能量最小化

首先描述解决最小化问题的基本方法然后讨论如何提升它的表现。

基本方法。为最小化能量函数，重复所有可能的联结树配置直到一个确定的最大规模。为了限制配置数量于合理界限，限制可用的基本机制和扩展块数量。观察到艺术家制作的转换式弹出普遍地包含三或四个基本机制，所以限制基本机制为五个或更少。扩展块数目被限制在12个或更少——太多会导致更小的块组成弹出，使得弹出更难制造且脆弱，还花费更多时间进行最优化。

由于能量函数是高度非线性的，用一个遗传算法(GA)来最优化几何学参数。为了约束搜索空间，所有几何学参数包括顶点和折叶位置，备用基于整数的离散像素定位编码，用1024x1204分辨率的像素点阵。在初始化中，所有参数被随机生成。对一个配置x的适宜函数通过-E(x)给出。突变通过为个体中每个参数添加一个正常地发散的值来进行。用到了单点交叉和锦标赛选择。在实现中，运行GA十次，每次迭代50代，使用1000个个体的规模，交叉可能性为0.4而突变可能性为0.3。

加速。由于参数空间的高维度，直接用向上文描述的GA是耗费时间的，要花费数十个小时来生成合理输出。最初实验显示了在很少的迭代后GA获得了一个全机制足够的整体结构，但花费了久得多的时间来最优化每个块精密的顶点位置。用这个观察所得，加入一个贪心策略来加速研究。特别地，在最优化期间，在GA重复中检测到好的块顶点并固定它们来减少答案空间的维度。为了增加好的块顶点潜在数量，块提纯在检测前完成。好的块顶点满足以下任意一条标准：

顶点属于翻转且位于两个输入图案边界上的块。

顶点属于翻转且在一个状态下位于输入图案边界，另一状态下目标形状内部的块。

顶点属于不翻转且在一个状态下位于输入图案边界，另一状态下被其他块阻隔的块。

进一步的加速通过在GPU上使用CUDA实现以上算法来达成。由于是高度并行的，GPU实现运行快的多。通常，贪心策略提供5到10倍的加速，而GPU实现比起CPU实现提供另外10倍加速。总的来说，加速减少处理时间从数十个小时到些许分钟。

3.3 块提纯

由基本最优化方法获得的初始转换式弹出有包含很少顶点的多边形块，所以只能表示粗糙简单的形状。因此，不可避免的仍有一些冗余区域(被弹出覆盖但不被用户给定图案覆盖)以及未表示区域(被用户给定图案覆盖但不被弹出覆盖)。这一步骤的目的是通过提纯一些块来减少冗余区域的大小。由于用一个高权重来惩罚未表示区域，它们很少出现在初始结果里。

轮流为未转换状态和转换后状态处理冗余区域。原则上，对于未转换状态，修剪所有的冗余区域，除了那些属于在转换后状态中可见区域的部分，因为修剪那些部分将影响转换后状态的外观。转换后状态的冗余区域用一种相似的方法修剪。理论上，以上描述的修剪模式可能潜在地打破机制的结构：一个折叶可能被完全修剪掉，一个块可能被修剪为两个不相连部分，或一个滑落机制中的滑块可能被修剪得太短。然而，没有在实践中观察到上述问题的出现，所以不包含额外的模式来避免它们。

3.4 质地涂画

在决定了机制和块集之后，必须把用户提供的质地涂画到块上去。和那些在未转换和转换后状态中可见的块一样，所有其他块也应该被涂画来使得转换过程期间的中间态外观看起来更自然。每个块的两面都必须被涂画。质地比对过程很简单，正如只需要直接不作任何变形地将源图案和目标图案比对到块上去。对每个块必须决定用哪个质地。

每一面轮流进行这一步。有以下决定：

如果这面在两个状态下可见(或部分可见)，那么只在未转换或转换后状态可见的部分被分别地用源图案和目标图案涂画。任何矛盾部分以及在两种状态下都不可见的部分用源图案描绘。

如果这面只在一个状态下可见(或部分可见)，那么整个面用单一质地描绘。

另外，两种状态都不可见的面整面用源图案质地涂画。

必须注意冗余区域和未表示区域。用源图案和目标图案的平均颜色填满任何荣誉部分来保证图案内部质地的连贯。未表示区域成为基本块的一部分并且不被包含到弹出的转换中但仍用比对上的源图案或目标图案的质地描绘。

3.5 蓝图生成

最终生成一个蓝图来帮助用户制作这个转换式弹出。记住一个块和任何进一步依附于它的多边形形成一个扩展块。一个给定的扩展块的正反两面都被打印在一张纸的同一侧，彼此接触。用户沿着正反的分界线折叠剪下的形状并将它跟自己粘起来提供两面的外观。

**4 小结**

论文提出了一个自动生成转换式弹出的方法。使用简单结构的三种基本机制，以及用不同方法改变和结合它们，允许作者使用最优化来设计流畅地在两个二维图案间转换的结构。最终输出是一个构建该机制的蓝图，以至于用户能轻松地制作真实工作的弹出。作者用很多对的形状证明了他们方法的效率。结果显示他们的方法可以生成视觉上令人愉悦的转换式弹出。

但是仍有不足之处。由于作者在他们的方法中使用的是多边形的块，所以对于有圆弧边界图案的操作很不便捷。还有一个限制是在方法中假定纸张厚度为零，如果折叶附近有太多层，可能造成制造机制的不正常折叠。解决这个问题可以使加入厚度项到能量函数中。最后，他们目前的算法没有考虑图案的语义。潜在的办法是要求设计者提供每个图案的语义分割并指出对应关系，最终最优化分段区域的匹配而不是整个图案的匹配。

参考文献

[1] Nan Xiao, Zhe Zhu, Ralph R. Martin, Kun Xu, Jia-ming Lu, Shi-min Hu. Computational Design of Transforming Pop-up Books[C]. ACM Trans. Graph. 37, 1, Article 8 (Dec. 2017), 14 pages.