

---

## 摘 要

织物仿真技术是将计算机辅助设计 (CAD) 技术应用到织物设计中, 通过计算机设计织物的纹样、图案, 模拟生成具有真实感效果的织物表面外观。

织物仿真技术的应用不仅可以提高纺织产品的设计效率, 将设计人员从繁杂的劳动中解放出来、使之能够将更多的时间和精力投入到产品构思及开发中去, 而且由于不必实际生产出样品就能够得到具有真实感的设计效果图, 生产周期大大缩短, 产品开发成本大幅降低; 同时加快了企业对市场的反应速度, 企业、供求之间的信息交流更加便捷、高效, 消除了生产过程中的盲目性, 增加了企业的竞争力。

纺织 CAD 技术在发达国家的纺织行业已经比较普及, 但在国内, 只有少部分具有一定规模的企业在纺织品的设计、生产中用到了 CAD 技术, 国内市场占有率仍然非常低。在织物外观模拟系统的开发方面, 国外的机构起步较早, 开发的系统也已经进入商品化阶段; 相比较之下, 国内的系统基本上都是处在实验室试验阶段, 没有开发完成比较成熟的商品化系统。

本文设计了一个在二维空间中实现织物平面效果模拟的系统: 在按照工艺标准的前提下, 以纱线为基本的单元, 通过在经纬两个方向各种不同纱线的排列、纵向的经纱和横向的纬纱的交织来模拟织物的外观。该系统使设计人员可以简单、方便地改变织物的组织、纱线排列, 随心所欲地对织物模拟图上各种颜色的经纬纱线进行调色配色, 淋漓尽致地将设计思想表达出来, 在织物上机织造之前就能看到实际织造结果。其中着重介绍了纱线紧密排列, 存在相互覆盖时的显示算法 (包括同种类型的纱线相互覆盖和不同类型的纱线相互覆盖), 单根纱线的平面模拟算法和生成纱线毛羽纹理的方法。接着, 将模拟环境延伸到三维空间, 用曲面拼接的方式简单构造了纱线表面的模型; 然后提出了一种模拟纱线交织形态的方法, 并作了精确的计算, 最后用上述模型和方法模拟了平纹织物的外观。

---

## Abstract

Applying CAD to textile designation, using computer software to design textile pattern, the technique of fabric simulation can generate realistic appearance of all kinds of fabrics.

With the application of fabric simulation technique, textile designation becomes quite easy and effective; designer will be released from multifarious and trivial labor and have more time or energy to engage in conceiving production; you can obtain realistic textile pictures and need not to weave it actually, this will lead to a great many benefits: shorten produce period rapidly, reduce running expense sharply, fasten enterprise's reaction to marketing, make convenient for information communication and producers will free from motiveless manufacture. Along with wild employment of electronic business, producers can utilize Internet to put out newly products, do business and intercommunication. Therefore, the producer will be more competitive.

The technique of fabric simulation has been wildly accept and put into use overseas. But inside China, there is only a fraction of fabric manufacturers take advantage of CAD technique during designation and production. And the marketing occupation of this software is still at a comparatively low level. In the respect of fabric simulation research and development, institutions overseas started off earlier, and have put out some commercial software; while the software in domestic is however just at the experimental phase.

This paper demonstrates a system based on 2-dimension, which implement plain fabric simulation. According to fabricate criterion, viewing thread as the basic unit, we generate the textile pictures by arranging threads, and crossing warps and wefts. This system renders textile designers change all the fabric properties freely, and express ones' ideas incisively and vividly. Meanwhile, this paper emphasizes the display algorithm of tightly arranged threads

---

overlap, thread picture simulation and feather generation. In succession, we stretch the simulating environment into 3-dimension. Structuring a simple model by patching up Bezier covers to figure the surface of a single thread, putting forward an accurate method to display the shape of thread interlacing, we simulated the tabby fabric.

## 致谢：

感谢我的导师许端清教授在我攻读研究生期间给我的关心和帮助，正因为您对我学业上的悉心指导，才使我有机会参加图形图像领域的研习开发工作，并让我能够顺利完成研究生阶段的学业。

本篇的撰写成稿，离不开马凌洲学长在织物模拟仿真课题中给我的耐心引导、真诚的提议和无私的帮助。我不仅从他那里受到了思路的启发，更学到了执着、踏实、勤奋的精神。

感谢和我一起度过研究生生活的同学、好友，以及所有帮助过我的人，正因为有了你们的帮助，我才能顺利走过研究生阶段的学习生活，衷心地感谢你们，我一生的良师益友！

# 第1章 背景介绍

## 1.1 纺织 CAD 与织物仿真背景介绍

### 1.1.1 CAD 与织物仿真的历史与现状

近几十年来,计算机技术飞速发展,在各行各业得到广泛的应用,例如在建造桥梁、飞机的工程设计过程中,几乎每个设计细节都被显示在电脑中,从而可以通过受力分析计算出所有可能的影响力。但是对于纺织行业,计算机的应用却

发展缓慢,传统的织物设计和织物生产在很大程度上仍停留在依靠建经验、直觉、试验与改正的基础上。近年来,随着新材料的开发、新制造技术的应用花样繁多的技术含量较高的织物广泛的进入了各行各业,从医药行业到土木工程,从娱乐业到军事工业、从航空到深海勘探,织物应用范围的不断扩大,要求织物的设计更加小心、精确,因为设计上的失误可能会引起非常严重的后果。这时借助计算机技术进行织物的设计与生产是势在必行,织物仿真 CAD 技术在世界各国得到了迅速的发展。但是很遗憾的是我国纺织行业 CAD 技术应用水平还很低,据不完全统计,中国纺织服装行业使用 CAD 系统的企业有近千家,普及率不到 2%,而在工业发达的欧美国家,纺织服装 CAD 系统在企业中的使用率高达 70%以上,我国台湾省纺织服装 CAD 的拥有量也在 30%以上。这表明,我们的纺织服装 CAD 系统的应用和国外存在着很大差距,对纺织 CAD 系统进行研究开发是一项紧迫而极具有实用价值的工作。以下对 CAD 发展历史、纺织 CAD 主要内容及其现状作一简单介绍

#### 1.1.1.1 CAD 技术的发展历史

- 50 年代——技术准备期

50 年代,大多采用用电子管计算机,用机器语言编程,其图形设备也仅具有输出功能,计算机主要用于科学计算。

- 60 年代——开发应用期

60 年代,美国麻省理工学院开发出 SKETCHPACL 交互图形处理系统的同时,首次提出了计算机图形学、交互技术等观点为 CAD 技术的应用和发展打下了理论基础。于是,美国的一些大公司开始研制 CAD 和 CAM 的实用技术,并且推出了许多商品化的 CAD 设备和 CAD 工作站。

- 70 年代——实用化期

70年代,出现了许多面向中小企业的CAD/CAM系统,CAD工作站的数量也急剧增加。此期间,出现了多种形式的图形输入输出设备。CAD/CAM与CAT开始接合,从而形成了计算机辅助工程(CAE)。

- 80年代——普通应用期

此期间,由于32位高档微机的问世及计算机外设的发展,CAD技术突飞猛进,图形系统和CAD/CAM的销售量与日俱增。

- 80年代中期以后——标准化、集成化、智能化的发展期

由于图形接口,图形功能日趋标准化,为CAD的推广创造了有利条件。随着人工智能和专家系统应用于CAD技术,使得工程数据库及管理系统,知识库及专家系统,用户接口管理系统集于一体,大大提高了自动化设计的程度。

### 1.1.1.2 纺织CAD的内容

在纺织品设计中,往往要进行大量的组织选择及设计,色纱选择及模纹设计及繁琐的织物设计计算。传统的设计,以上工作是通过手工完成的。所以每设计一个新品种,设计人员要付出艰巨的劳动,这样便造成纺织设计周期长,花样少,满足不了市场的需要。现代高新技术的发展,为纺织设计开辟了有效的途径。因而、纺织品CAD应包括:提供织物形态模拟功能和控制机织的CAD的功能。

CAD织物模拟功能是提供织物织成后的图样。系统只要输入颜色、组织及密度等,织物的最终效果就可在屏幕上显示出来,而不用打小样。CAD控制织机的功能指根据其设计可以直接或间接地控制机器的运行。系统通过扫描纹样,输入织物每一部分的组织结构,经过显示、修改、更正,然后通过穿孔卡或穿孔带或软盘或EPROM存贮模块输入机器来控制织造过程。

### 1.1.1.3 纺织CAD的现状

在纺织工业生产中,CAD技术的应用最早可以追溯到20世纪60年代,当时美国IBM公司首先研制成功了纺织工艺自动化系统,使提花织物生产过程中的花型设计从原来的手工方式设计、画图、冲版变为采用交互式的屏幕作图和自动冲版,实现了纹制工艺自动化,使得自动化设计在纺织行业的应用成为现实。此后,CAD技术在纺织行业得到了重视和发展。特别是近年来,由于市场竞争及消费者对产品的要求多样化、个性化、计算机越来越成为纺织品设计中不可缺少的工具。

在国外纺织品 CAD 的应用比较早,发展得比较成熟。美国罗姆公司采用电脑设计,无论织物的颜色、规格、图案等均能满足用户的要求;汉来、卢本期公司在改造过的织机上安装两套监视装置,分别监控图案、服装款式和首件新产品的生产情况,管理人员可在屏幕上观察全过程,从而减少了设计人员的用工、大大缩短产品的生产周期。BONAS 机械制造公司研制了一套称为自由式的提花织造系统,该系统可设计各种提花图案,系统能在原始图案呈现于屏幕上后,使光标快速移动以进行修改,各种图案可转任意角度或移动。1600 万种颜色组合可快速转化,用各种格式对数据进行存储等;该系统能在屏幕上按“纹版形式”以不同的颜色展现不同织造组织的效果,以方便地设计多层织物。在多臂机上,苏格兰纺织学院推出的三个不同能力等级的计算机辅助设计系统,可满足不同专业人员的需要,其最高一级 CAD 系统能提供 3000 经  $\times$  3000 纬面积的图案设计区域以及织物的俯视图和综框占有数据等。还能在经向和纬向设计出十二个图案的毛毯,能使数据文件中的图案连续显示,可用输入设备给某些经纱和纬纱上色,增加并插入线条,还能用彩色打印机“仿制”出织物的真实状态等等。另外,意大利、日本等国在针织大圆机及袜机上应用的 CAD 系统,可以方便设计并实现各种织物组织及花型的编织。德国 SIOLL 公司应用于横机上的 SIRLX 系统,可很方便地实现纹花、分针纱罗、锁边等结构的花型。

目前国外 CAD 技术正向着智能化、网络化方向发展。国外知名的 CAD/CAM 系统,如享有服装 CAD/CAM 系统“奔驰”美誉的德国艾斯特 (ASSYST)、法国力克 (LECTRA)、美国格勃 (GERBER)、加拿大派特 (PAD),斯洛文尼亚的 ArahWeave 等系统。

国内的 CAD 系统从起步到现在发展得也颇具特色,不少功能模块很有特点,如汉化界面的亲和度、简易的操作等,这些都比较适合国内用户的使用,而且价格仅为进口产品的一半或 1/3。比较有代表性的 CAD 系统有以下几种:中国纺织科学研究院开发的织物仿真 CAD 系统,浙江大学电气自动化研究所(浙大经纬)开发的纺织 CAD 系统,上海百锐纹织物计算机辅助设计系统等。

#### 1.1.1.4 纺织 CAD 产品分类

纺织行业由于门类较多,无单一通用的 CAD 技术,各类纺织专用的 CAD 技术开发应用在我国主要有以下八类,其中织物设计 CAD 和服装 CAD 的就应用较为广泛。

##### (1) 织物设计 CAD

织物设计 CAD 技术主要用于色织产品的设计,其主要原理是根据色织物的组织和色纱排列自动生成织物模拟图案,以替代产品试织打样工作。计算机彩色屏幕上的图案可随意修改、调色,也可与国际流行色卡配套使用。

主要功能:

- a. 对纱线的粗细、色泽、捻向、捻度等进行设计及修改;
- b. 利用编辑功能直接输入组织图或从组织库中调用组织图, 进行纹版排列和穿综设计;
- c. 根据生产要求输入参数, 系统自动生成总经根数、幅宽、上机箱幅等上机参数;
- d. 可从计算机上模拟现实织物小样图, 对织物外观进行选择, 同进还可以通过彩色打印机输出逼真的效果图。

## (2) 纺织 CAD (大提花 CAD)

纺织 CAD 系统是用于提花织物设计的专用系统, 它利用计算机强大的计算功能和高效率的图形处理功能, 改造传统的纺织工艺, 实现纺织工艺自动化。

主要功能:

- a. 对纹样可进行一次性扫描或分块扫描后再进行图案拼接。可以确定纹样色数、经纬密度、纹针数、纺格数等织物规格数规格数据, 在屏上显示平涂意匠图效果;
- b. 对组织图或图案可以进行任意裁剪、旋转、组合、接回头、迭加变色等图案编辑及纹样设计处理;
- c. 可随意修改样意匠、去杂色点、进行勾边、间丝设计、辅助组织处理等;
- d. 根据意匠图, 纹版样建库, 自动进行纹针和辅助针处理;
- e. 按照纹版处理信息, 用计算机控制孔机自动轧纹版或直接控制经纱运动。

## (3) 印花 CAD (印花分色制版 CAD)

印花 CAD 是利用计算机对织物印花图案进行设计及制作黑白稿, 可代替传统的手工画稿、扫描、连晒、感光制版等。

主要功能:

可以利用编辑功能直接进行图案设计, 可以将来样扫描输入, 进行拼接、接回头等工艺处理, 灵活自如地对图案进行设计修改; 对印花图案进行拼色处理及圆整修改处理, 提高图象质量; 根据不同要求由计算机自动分色或手工分色。将图案花样每套颜色单独保存为单色稿或黑白稿制作胶片、制网, 供圆网、平网、流筒印花机印花。

## (4) 绣花 CAD

对绣花花样进行计算机辅助设计，并将设计的花样格式转为绣花机的针法，而生成绣花机能够识别的纸带或磁碟，用于单头、多头电脑绣花机生产。

主要功能：

- a. 边框输入具有多种花版输入功能，系统还提供拉布补偿功能；
- b. 系统可进行自动、手动、单针、圆弧针、插针及多种特殊针法设计；
- c. 对花版可进行任意的移动、旋转、放缩、拷贝等处理，并有进行颜色辨认，连串复版等功能；
- d. 对花版可是进行全像管理，可进行编辑、删除、改名、索引、拷贝等，并可远程通讯；
- e. 花版检查功能提供了实际刺绣所需的所有参数，设计的花版可直接输入到绣花机，也可经纸带穿孔机轧孔或输出磁盘。

#### (5) 服装 CAD

采用计算机辅助技术对服装款式效果、结构样片、推版放样、排料进行设计、服装 CAD 设计精度高，可缩短设计周期十几倍到几十倍，并节省面料 2-5%。

主要功能：

- a. 利用系统图形功能、色彩变换功能及内存的各类人体模型进行服装款式构思设计；
- b. 款式设计后在屏幕上形成样版图形，可随意修改；
- c. 快速推版放样功能，一次可完成多个号型的工作；
- d. 自动排料，数控裁剪。

#### (6) 测配色 CAD

测配色 CAD 改变了传统上繁杂的人工打样配色、配色精度高、省工省时，而且可以降低染化料消耗，减少环境污染。

主要功能：

- a. 对来样进行测色分析，可测定织物色差、褪色牢度、沾色牢度、白度、亮度等指标；
- b. 根据测色对色号、染料进行检索；
- c. 系统可提供多个配方，根据操作者意愿自由配色。

#### (7) 针织 CAD

针织圆纬 CAD, 将针织线圈抽象成一个个方格, 用规定的符号画在格内, 以示织物线圈组织规律, 然后在计算机屏幕上进行花型设计, 经软件处理, 转换为电脑提花织机通用的数据格式进行生产。

主要功能:

- a. 可进行花型输入;
- b. 具有画点、线、多边形、填色、换色等功能;
- c. 具有缩放拷贝、选色毛巾、对称拷贝、旋转及透明等功能;
- d. 可进行包边、接回头、调色、绘制中英文字及插入、删除等图案设计;
- e. 数据转换模块利用图像压缩、编码、格式化为不同的信号, 由磁盘或芯片进行传送。

#### (8) 针织横机 CAD

可以计算出羊毛衫、羊绒衫等产品生产工艺数据, 输出工艺图和工艺单于针织横机生产。

主要功能:

- a. 系统根据输入的参数对款式、织物密度、组织及各部位尺寸进行自动计算并打印出上机工艺单;
- b. 系统根据工艺科分别计算出各部位线圈个数, 并根据输入和各种组织圈重, 结合织耗等因素计算出衣片重, 进而计算出产品克数、用纱量和成本;
- c. 根据设计出的衣片, 将各项收放针、抽中针、起口、落片、翻天覆地针等操作折合成机头转达数, 在衣片工艺单上打印输出;
- d. 对设计和各部位衣片尺寸表、各种子规格的工艺单打印输出, 并具有产品检索及非标准产品工艺单打印输出等功能

### 1.1.2 软件设计目标及其意义

一个产品的诞生需要经过许多环节, 从产品调研开始, 直至正式投产需要花费大量的人力、物力和时间。随着计算机技术的发展, 微机在织物设计中的应用日益普及, 尤其是计算机辅助设计 (CAD) 技术, 可以将设计人员从繁杂的劳动中解放出来, 使之能够将更多的时间和精力投入到产品构思及开发中去, 所以对纺织 CAD 软件产品进行研究开发, 是具有很强的现实意义和应用价值的。

通过对前文提到的目前市场上应用较多的产品的分析, 发现主要存在以下几个问题: 虽然机织物 CAD 软件在市场上种类繁多, 功能比较齐全, 但只是解决了少数

织物的外观模拟问题，织物的外观和图案模拟效果不太真实是一个突出的问题。大多数机织物 CAD 系统仅是考虑了纱线密度和纱线形状的配色模纹图。由于较少考虑组织结构对织物外观的影响，只能模拟经纬纱异色的织物，对有较长浮长线的组织模拟则会严重失真。

在考虑了织物的交织规律后，并在二维模拟显示的基础上增加了一些三维显示的思想，有的 CAD 软件可以模拟经纬纱同色或异色的单层组织织物、多重组织和多层组织的外观和图案，模拟的织物有立体感。能反映同色经纬纱的交织效果和织物中纱线的弯曲现象，对织物外观的模拟效果虽有明显改善，但仍不能彻底改善模拟效果。

影响织物外观最重要的因素是织物组织结构和纱线颜色，织物的表面起伏不定，有突起，也有下凹，加斜纹组织、凸条组织、蜂巢组织等，布面的凹凸在光照下，形成明暗外观效果，经纬纱在织物中的位置也不像组织图案中那样均匀分布，纱线有移位，有弯曲，甚至还有重叠。显然，织物不是一个二维的平面，而是一个三维的物体。要真实地模拟机织物的外观和图案，就必须进行三维图象模拟。通过对市场上主要产品的分析，发现在商用的 CAD/CAM 系统中，没有对织物进行物 3D 建模，Gerber Garment 技术和 Info Design System' s 软件程序也仅仅是动画。虽然国外很多软件系统声称可以模拟织物的三维图像，其实仅是将模拟的织物二维图案贴在三维物体上，用来显示人体着装时面料的悬垂性和图案的变化，或者显示装饰织物用作沙发布或桌布时的织物图案的空间伸展效果，这都不是真正的三维模拟机织物面料本身的图像。

近年来，计算机辅助设计制造技术日趋成熟，应用也越来越广，纺织业的各个方面，如印花图案，织物组织结构，服装设计等，采用计算机技术后，不仅大大提高了效率，而且质量等也大为提高。我国是纺织品大国；产品要跟上国际快节奏，多变化的潮流；必须尽快普及应用计算机辅助设计制造技术。

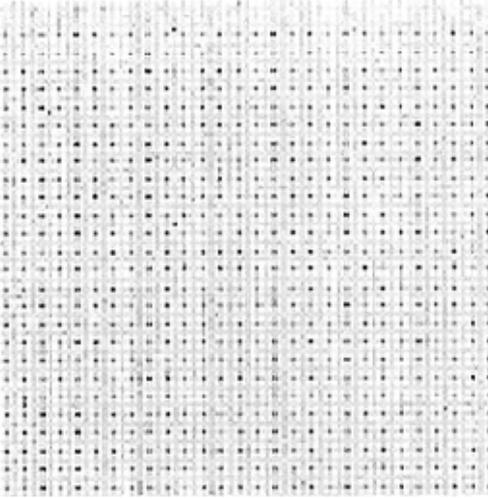
通过对上面的问题分析，设计出一个三维织物纺织系统，是具有很强的实用价值和广阔的市场前景的。

在设计本系统时，主要从以下几方面考虑：

- 系统以纺织工程设计方法为依据，对色织物组织结构、外观图形以及生产工艺的设计过程采用计算机进行自动处理，尽量替代人工处理方法。
- 系统设计时，与色织物设计中设计人员设计的意图、分析方法、实际经验密切结合，提高系统的实用性和正确性。
- 系统要满足灵活性要求，各个子系统互相独立，可随时扩充，便于维护和修改，能适应各种变化。

### 1.1.3 技术背景

目前在平面织物仿真领域应用主要有两类：以工艺需求为基础的商用软件和以研究为目的的研究性软件。商用的织物模拟软件大体上采用了相同的数据结构和模拟方法：以纱线为基本的单元，通过各种纱线不同的排列、交织来模拟织物的外观，取得了比较逼真的织物外观效果。而研究为目的软件数量较少，其中有人提出用小波分析方法模拟（如下图）（参考文献 4），有一定的理论价值。



在三维织物纺织中，在光照效果方面，主流的方法是用 BRDF(双向散射分派函数)来计算生成织物各向异性的光照效果（参考文献 3），最近微软研究院提出的旋转光片方法也获得了非常逼真的效果(参考文献 8)。在织物结构方面，一些文献（参考文献 5）提出了：把纱线作为织物最基本的元素，利用计算机辅助几何设计技术开发一个基于纱线的 3D 模型。最基本的思想是把每根纱线看成是沿着中心线紧密缠绕的曲线。缠绕的曲线和中心线路径交织成可变化的纱线外形，而中心线的排列构成了纱线排列。另一些文献开创了这样的方法：考虑到织物在结构上是很多小单元重复组成的，所以用原子模型代表直的纤维段，将这些纤维段重复应用到织物元素中，从远处观察，就能得到有真实感的图像(参考文献 6)。

### 1.1.4 纺织 CAD 应用前景展望

目前我国的纺织行业中，无论从改造传统生产方式的经济效益来看，还是从其开发应用的研究成果来看，织物设计 CAD、花型 CAD、服装以及印染 CAD（包括测配色 CAD）等的推广应用条件较为成熟。

1. 织物设计 CAD 技术较为成熟，从软件开发的水平看与国外同类软件相当，该系统适用于毛纺织和色织等的产品设计，优点是投资少，见效快。毛纺织和色

织企业目前国内约 1.200 多家,但目前应用 CAD 系统的仅有 10%左右,因此在推广方面具有较大潜力。

2. 目前全国约有 3,000 余家印染厂,就数量而言,印染 CAD 技术的应用大有发展前途,其主要内容包括印花图案 CAD 及花型 CAD 系统,它们在软件开发和推广应用方面都已经取得了较为显著的成果,涉及到丝、巾被、针织、印染等行业,量大面广。

今后, CAD 技术的主要发展趋势是集成化、网络化和智能化;同时,价格低、性能高的工作站和微机工作站将成为 CAD 系统的主流机种,而 CAD 软件功能的开发将是发展 CAD 技术的关键。

计算机辅助织物设计在国内外的迅猛发展和广泛应用,为产品设计人员提供了先进的设计手段,它突破了传统的设计方式,用显示器屏幕和鼠标等代替了意匠纸和笔,设计者可以在屏幕上很方便地设计和修改织物的组织,使设计者从繁重的手工设计中解放出来,可以有更多的精力用于新产品的构思和开发之中。但这并不是织物 CAD 技术发展的终极,一方面,现行的 CAD 系统在织物组织(特别是复杂组织)设计功能上有待进一步完善,现行 CAD 系统离真实模拟织物外观的要求还有相当的距离,因此,从单纯的组织设计向组织外观效果的真实模拟方向发展将仍是织物 CAD 技术的研究主题,而从织物的组织结构入手,研究织物的几何模型,从而进行三维模拟的研究将是实现外观效果真实模拟的较为理想的途径,国外正在开展这方面的研究。另一方面,计算机辅助设计的最终目的是为了生产,由计算机根据设计参数控制机器工作,这就是 CAM(计算机辅助制造)的内容,而由设计、生产、成本核算到管理等一系列企业活动则是 MS(计算机集成制造系统)的内容,因此,从 CAD 到 CAM 直至 CIMS,对机织物 CAD 系统提出了更高的要求,也为机织物 CAD 技术的发展指明了方向。

随着计算机软件、硬件的不断发展和更新换代,我国纺织 CAD 技术也将向着更高级、更方便、更实用的方向发展,向着机电一体化方向发展。这需要从事纺织 CAD 研究及软件开发的人员,做大量的研究和开发工作。这将极大的推动我国纺织工业的发展,为我国纺织工业从设计到生产实现电脑全自动控制提供保证,从而提高我国纺织工业的国际竞争能力。

## 1.2 同类产品比对

目前,市场上比较成熟的织物模拟仿真软件有: ArahWeave, Photoweave, Pointcarré, 中国纺织科学研究院开发的织物仿真 CAD 系统, 百锐纹织物计算机辅助设计系统(上海)和浙大经纬的产品。下面将对各软件进行介绍:(根据各自官方的发布)

## 1.2.1 ArahWeave

ArahWeave 系统和同一领域其它的竞争者不同，它是一个多任务系统，能够最大程度地发挥各个子系统的效用，其性能上的特性有：

- 所有的窗口都能够移动、调整大小，并且根据需要提供各种显示视图。
- 打印输出不会阻碍其他的工作。
- 在织物定制时可以同时使用好几种工具，提供了个性化的设计环境。
- 不论你对哪个部分的参数作修改，模拟图像都会实时的进行调整，保持了输入数据和显示输出的一致性。

同时，该系统在模拟织物时提供了相对简单的操作。所有要做的就是将以下元素结合：（图 1 是用该软件模拟生成的效果）

- 组织或提花纹版
- 经纬纱的排列
- 纱线的颜色、数量和结构
- 纺织的密度

运行平台：Linux

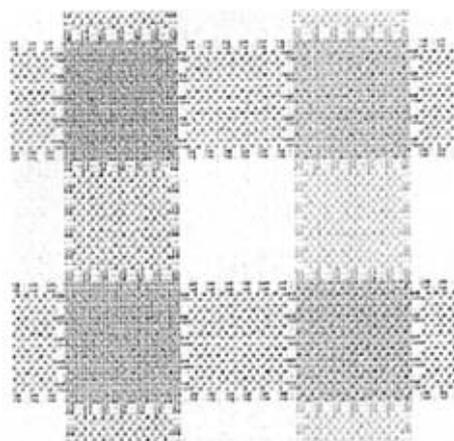


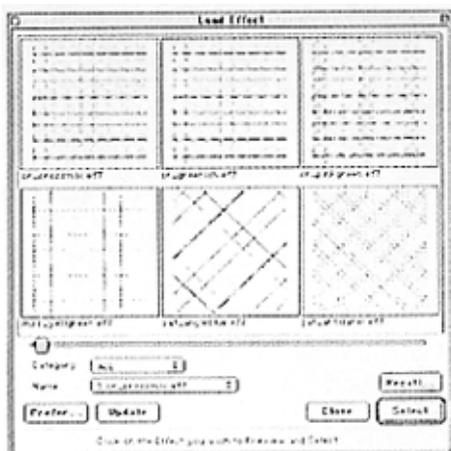
图1 (ArahWeave)

## 1.2.2 Photoweave

该软件宣称在经纱或纬纱方向可以使用至多16种不同的纱线；能够纺织出复杂的织物，展现了完美的真实感效果，操作简单易用。

操作步骤：

1) . 从纱线库中选择纱线（下左图）



2). 编辑经纬纱的颜色及排列 (下左图)

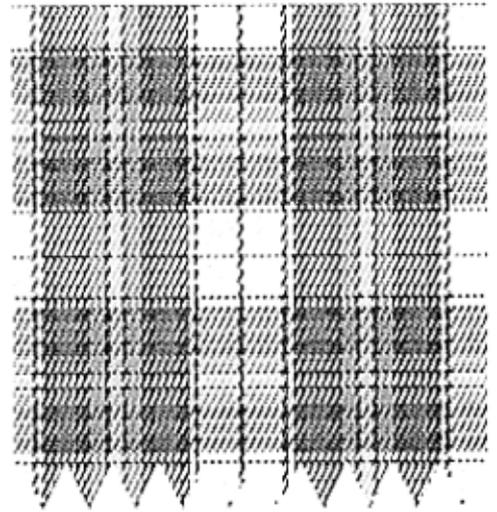
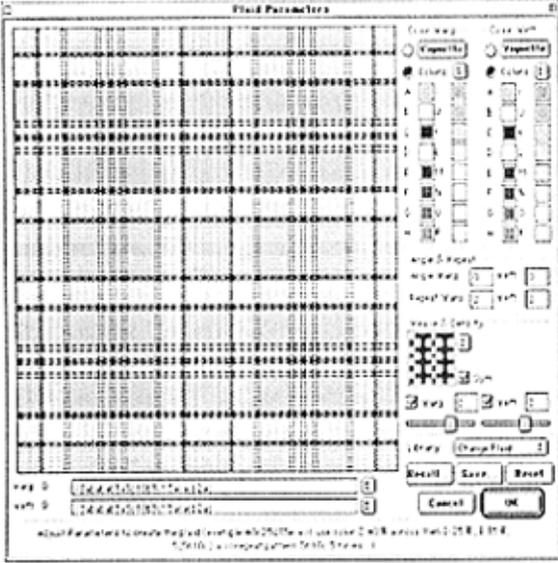
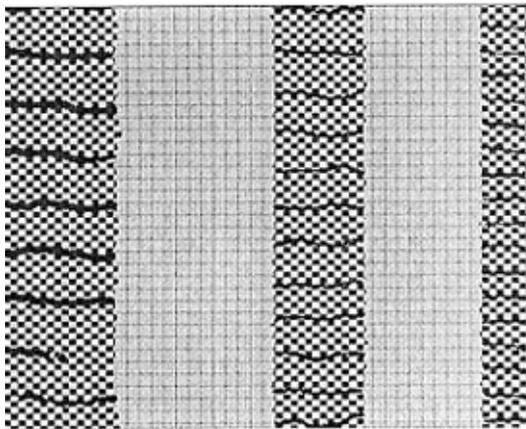


图 2 (Photoweave)

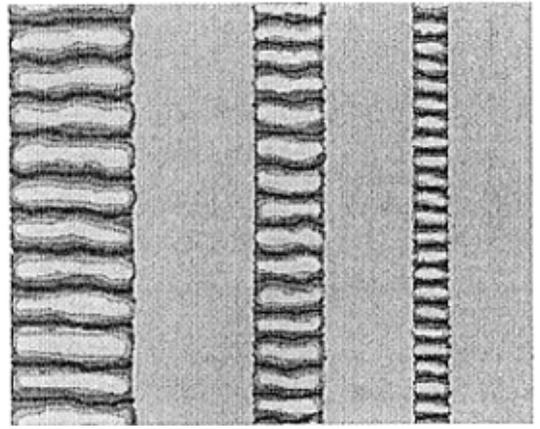
3). 预览与应用 (上右图)

### 1.2.3 Pointcarré

该系统最大的特点是通过手绘方式模拟了泡泡纱的褶



在纱线组织上绘制褶皱



生成褶皱图像

利用原来的模拟图像于褶皱图像的叠加模拟出了泡泡纱的效果

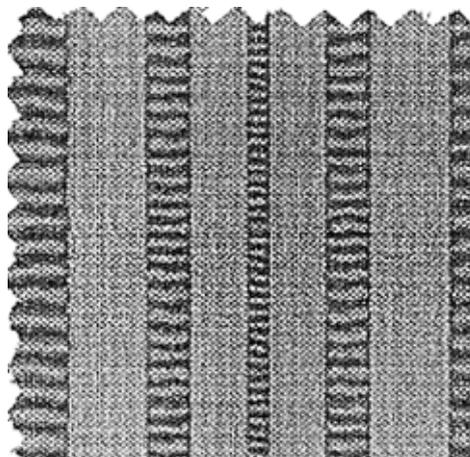


图 3 (Pointcarré)

## 1.2.4 中国纺织科学研究院织物仿真 C A D 系统

中国纺织科学研究院开发的织物仿真 C A D 系统将计算机图形技术和织造工艺结合,是适用于毛纺织厂和色织厂的织物辅助设计系统。该系统具有很强的织物仿真模拟功能,在很大程度上能代替织物的小样机打样工作。它具有极好的织物组织输入和纱线设计功能,可以设计单色纱、混纺纱、合股纱、竹节纱、毛粒子纱,并能模拟由这些纱线织成的织物,可以随意调整纱线的颜色,甚至可以模拟起毛拉毛后整理加工后织物的表面状态,并能在高级彩色喷墨打印机上打印出逼真的织物模拟图像的纸样。它不但能用于普通组织的织物设计,而且可用于复杂组织如纬二重、经二重、双层组织包括表里换层组织的织物设计。该系统产生一幅织物模拟图只需几十秒钟到几分钟,而且组织、纱线排列、纱线种类、颜色等均可随意变换,大大缩短了设计周期和减少了工作量,效率是普通手工打样的几十倍甚至几百倍。它开阔了设计人员的思路,提供了一个设计人员自由驰骋的空间。近两年来,中国纺织科学研究院开发的织物仿真 C A D 系统已有一百多套在全国二十多个省市自治区的纺织工厂里投入使用,产生了很好的经济效益和社会效益。

主要功能和特点:

- 纱线设计、存储和读取,可自建纱线库;
- 多种组织输入方法和复杂组织的生成和存储、读取;
- 织物模拟图像的自动生成和织物起毛表面状态的模拟;
- 织物设计数据文件的存储和读取;
- 织物设计数据显示;
- 上机图设计,照图穿综或自动穿综,自动生成纹板图;
- 织物模拟图像的彩色打印输出,可以获得和实物 1 : 1 的密度和花型;

规格:组织循环 $\leq 200 \times 120$ ;色纱排列循环  $1024 \times 768$  根;综片数 $\leq 20$ ;彩色打印输出 1670 万色,提供 8000 色(20 页)色卡;

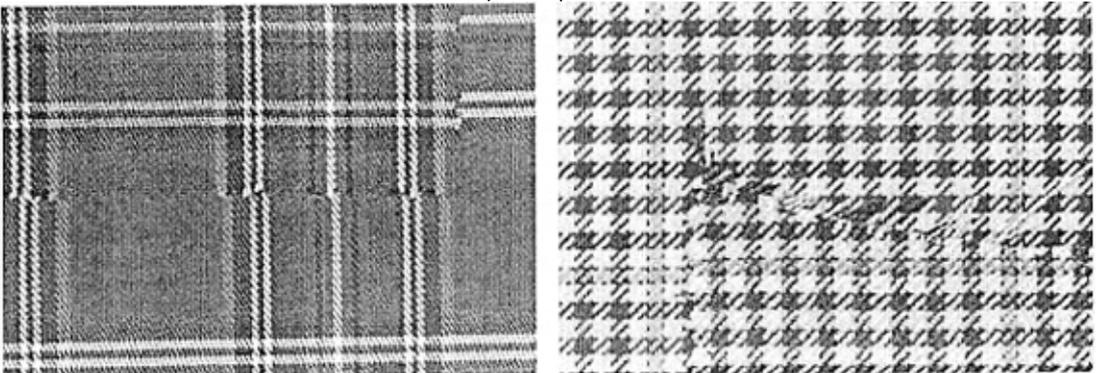


图 4 (中国纺织科学研究院开发的织物仿真 C A D 系统)

### 1.2.5 百锐纹织物计算机辅助设计系统

该系统提供了布样模拟与着装模拟；通过布样与着装模拟，使用户能够在织造前预先了解织物的基本风貌，着装效果，可以选择不同纱线与不同的织造材料，以及经纬间隔，起毛量，不同套色及不同角度等，以选择到最佳的效果。

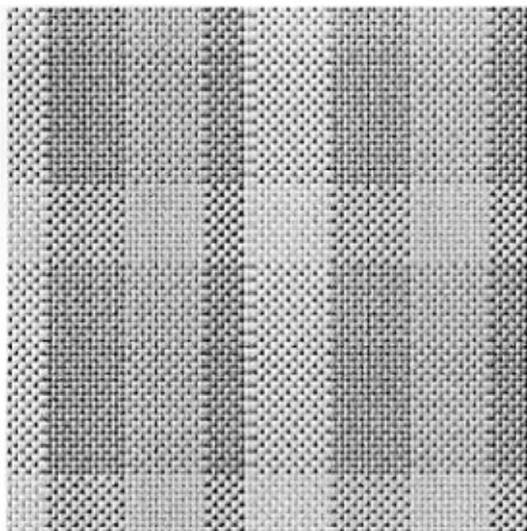


图 5(百锐纹织物计算机辅助设计系统(上海))

### 1.2.6 浙大经纬纹织物模拟系统

该系统采用先进的计算机仿真技术，模拟机台的织造过程，在计算机上真实展现织物的表面立体效果。现在主要有商标模拟、领带模拟、经编模拟等。主要适用于商标、领带、经编、毛巾、提花等轻纺企业；可自由设置模拟条件，支持打印输出；可用模拟效果检查纹版正确性，快速的试样，配色效果

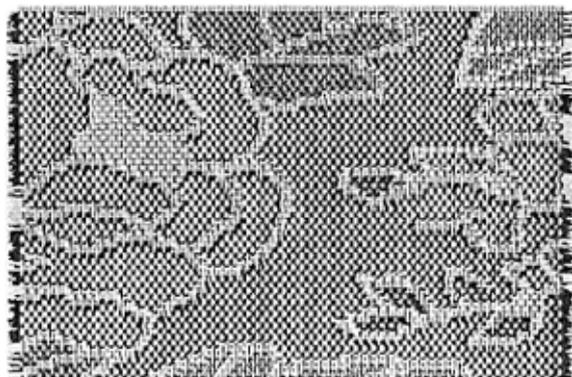


图 6(浙大经纬)

## 第2章 织物外观模拟的目标及总体构思

在计算机图形学中,对织物的真实感视觉效果已有越来越多的应用。我们对于织物模拟这一课题的研究,是为了设计出这样一个织物外观模拟系统,该系统将计算机辅助设计(CAD)技术应用到织物设计中,通过计算机设计织物的花纹、图案,模拟生成具有真实感效果的织物表面外观。这个系统需要拥有以下特性:提供一个科学的,便捷的织物设计平台。这个平台不仅应该包括织物设计中各个元素的设计工具,而且需要有一些工具支持织物的总体属性(例如织物外观的模拟,以及织物的变形、褶皱效果)的定制。

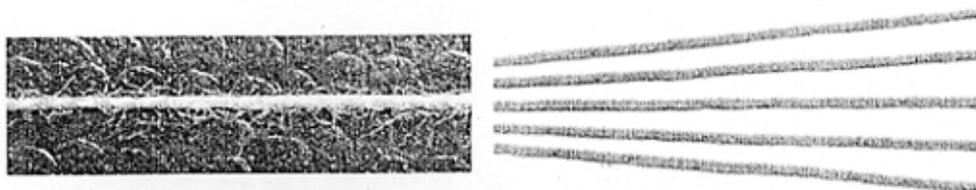
织物仿真技术的应用不仅可以提高纺织产品的设计效率,而且由于不必实际生产出样品就能够得到具有真实感的设计效果图,生产周期大大缩短,产品开发成本大幅节约;同时加快了企业对市场的反应速度,企业、供求之间的信息交流更加便捷、高效,企业可以将以前的“看样品定货”改变为“看效果图定货”,消除了生产过程中的盲目性;随着电子商务在企业中的广泛应用,生产企业发布新产品、销售企业订购新产品等流通环节都可以在网上完成,设计生产企业将新产品的实物效果图发布在网上,任由销售商来挑选,增加了企业的竞争力。一些纺织物、编织物的建模技术已经在计算机图形学的文献中被提出。(参考文献6)目前,这个领域的主要集中在两个方面:

- 1) 研究织物在物理和工艺上的属性和现象
- 2) 研究织物的视觉属性

我们对该课题的研究也主要落在这两个方向中:用生产、工艺上的参数、属性作为输入,模拟出既符合工艺属性又有真实感效果的织物图像。

### 2.1 织物结构:

众所周知,织物是由纱线组成的。而每一根纱线又是由数以千计的纤维互相缠绕而成的。纱线可以以各种不同的方式被组合,从而使得织物的外观千变万化。从制造过程来看,织物有两种基本的结构:纺织织物和编织织物。这里,我们暂时把研究的方向限定在纺织织物的范围内。在纺织织物中,所有的纱线被安排成沿着互相垂直的两个方向排列;纵向的经纱和横向的纬纱以不同的密度、材质(由纤维的长短、多少、粗细决定)和交织方式组合成或光滑或粗糙的织物表面。从纱线的组成上讲,每根纱线都包括了一些微纤维,它们沿着纱线轴盘旋缠绕(下左图),当然也有这样的特例:纱线由一束纤维并排构成(下右图)。



所以纺织织物的特性是由以下几个方面决定的：

1. 组成织物的纱线的属性，例如纱线的颜色、粗细与结构
2. 经纱和纬纱的排列密度和排列方式
3. 经纬纱的交织方法。

织物的上述特性决定了它的千变万化的外观和属性。

## 2.2 织物模拟的总体构思：

首先要解决的问题是在二维空间中实现织物的平面模拟。

该问题的数据模型是这样的

输入：上面提到的织物的三类特性

输出：织物的模拟仿真图像

设计工具：Visual C++ 6.0 以上版本

然后将模拟空间延伸到三维空间，使得设计人员能够获得关于织物属性的更全面的信 息，并且获得更精确的视觉效果

设计工具：Visual C++ 6.0 以上版本，OpenGL

## 第3章 平面织物仿真

### 3.1 设计思路

纺织 CAD 技术在发达国家的纺织行业已经比较普及，但在国内，只有少部分具有一定规模的企业在纺织品的设计、生产中用到了 CAD 技术，国内市场占有率仍然非常低。在织物外观模拟系统的开发方面，国外的机构起步较早，开发的系统也已经进入商品化阶段；相比较之下，国内的系统基本上都还处在实验室试验阶段。

在二维空间中对织物外观模拟系统开发，存在的主要困难是：

- 1、如何真实的表现纱线。包括真实的表现纱线的质地、形状、毛羽效果。
- 2、如何真实的表现纱线的交织效果。包括纱线交织结构分析，多重织物纱线之间互相覆盖的显示，现实交织过程中的纱线形态，纱线因为交织产生的阴影效果的生成。

考虑到模拟最终以图像形式输出，所以我们采用逐个像素描绘的方法来模拟织物外观。而在数据结构上以纱线为基本的单元，通过各种纱线不同的排列、交织来体现织物的结构。

以下四个小节先介绍该方法中用到的几种数据结构及其关系；再以图例说明程序中的各模块功能及它们之间调用一返回的关系；接着列举了程序中用到的主要算法，其中着重阐明了用于产生纱线毛羽效果的 Perlin 算法；然后是输入数据的详细介绍及软件的使用说明；最后直观地展示了用这个系统模拟生成的织物图像。

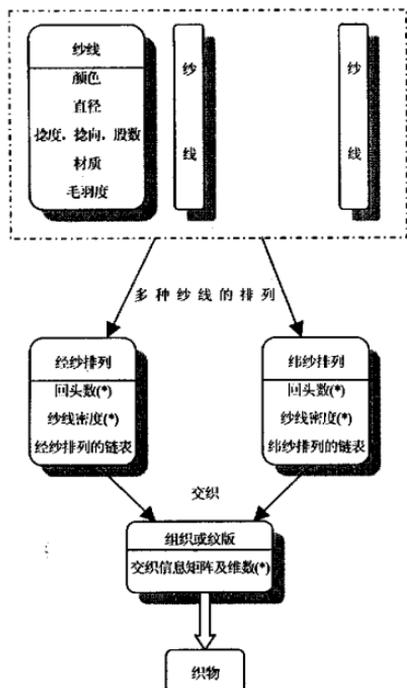
### 3.2 数据结构

#### 3.2.1 织物的基本结构

绝大多数的机织物纺织原理都是相似的：首先选出织物要用到的纱线；然后，将各种纱线平行并排，竖排的称为经纱，横排的称为纬纱；最后，经纱与纬纱互相交织形成了纺织品。

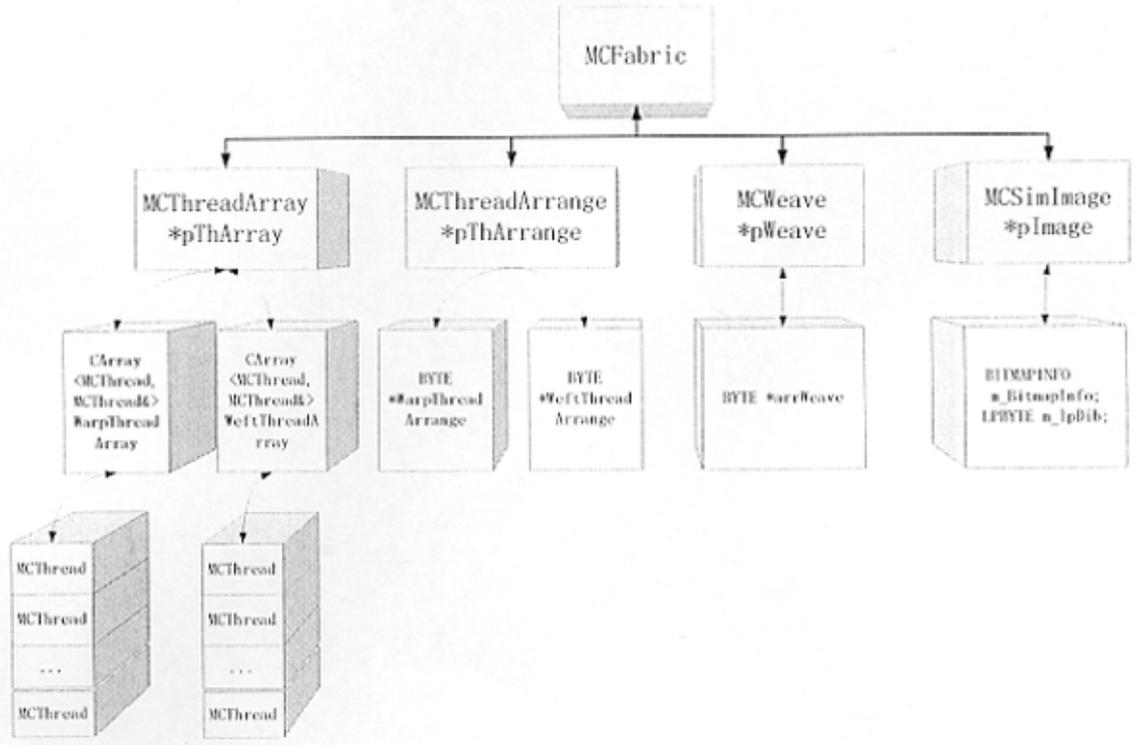
针对织物的结构和纺织过程，我们提出了模拟仿真用到的三种主要的数据类型：纱线（描述单根纱线的属性），色纱排列（表现纱线的排列规则），组织和纹

版（体现经纱、纬纱的交织结构），三者的属性及关系如下：



### 3.2.2 数据结构的具体实现

#### 3.2.2.1 程序中各主要的类，主要数据及其关系图



类的名称	对应的实体	类的作用	与其他类的关系
MCFabric	织物	存放模拟织物所需要的各种数据, 模拟结果数据, 和模拟方法	总管各个类
MCThreadArray	纱线数组	存放、管理所有纱线的属性, 是单根纱线属性的数组	实体是 MCFabric 类中的成员变量, 管理 MCThread 数据
MCThreadArrange	色纱排列	存放、管理纱线排列的数据	实体是 MCFabric 类中的成员变量
MCWeave	组织或纹版数据	存放、管理组织或纹版数据	实体是 MCFabric 类中的成员变量
MCSimImage	织物图像	存放织物模拟输出图像	实体是 MCFabric 类中的成员变量
MCThread	单根纱线属性	单一纱线的各种属性	实体是 MCThreadArray 的成员变量
CRenderThreadSect	纱线段绘制方法	纱线段绘制算法的实现	实体的方法在 MCFabric 中被调用

### 3.2.2.2 详细描述主要类的属性及其主要功能

描述单根纱线属性的类 **MCThread**

纱线是纺织织物结构的基本单元, 也是织物模拟图像中象素显示什么颜色的依据

```
class MCThread
{
public:
/*-----纱线的基本属性-----*/
    COLORREF color;    // 纱线的原始颜色, RGB 模式表示
    float    m_fDiameter; // 纱线的直径(mm)
    int      m_TwistsDir, m_nTwists; // 纱线的捻向 (0-无捻向, 1-正向, 2-
    逆向)、捻度
/*-----纱线的基本属性-----*
```

```

/*-----绘制纱线段时用到的一系列属性-----*/
int    m_ColumnColors;    //绘制纱线圆柱体时效果用几种颜色区分
int    m_ColumnGrads;    //绘制纱线圆柱体效果时各个颜色之间的变
化幅度
int    m_ShadowColors;    //绘制纱线交织的阴影效果时用几种颜色区分
int    m_ShadowGrads;    //绘制纱线交织的阴影效果时各颜色之间的变化
幅度
float   m_Slope;          //纱线捻度的斜率
int    m_TwistColors;    //绘制纱线捻度效果时用几种颜色区分
int    m_TwistGrads;    //绘制纱线捻度效果时各个颜色之间的变化幅度
int    m_PixelGrads;    //绘制纱线捻度效果时各个颜色的象素宽度
bool   m_CheckFuzz;    //对本纱线是否添加毛羽效果
/*-----绘制纱线段时用到的一系列属性-----*/
};

```

管理纱线数组的类 **MCThreadArray**

```

class MCThreadArray
{
public:
    //经纱纱线数组,数组的元素是单根纱线的属性
    CArray <MCThread, MCThread&> WarpThreadArray;
    CArray <MCThread, MCThread&> WeftThreadArray; //纬纱纱线数组
//从*.thr 文件导入纱线数据(具体方法见使用指南)
    void LoadFromThrFile(FILE* thrFile);
    void SaveAsThrFile(FILE thrFile);           //保存.thr 纱线数据文件
};

```

管理色纱排列的类 **MCThreadArrange**

```

class MCThreadArrange
{
public:
    int nWarpThread, nWeftThread; //经纱 色纱排列中 有几项 e.g. 2 A3B, 共 5 项
                                   //纬纱 色纱排列中 有几项....
    BYTE *WarpThreadArrange; //经纱排列顺序 E.g. 经纱排列为 2A3B,则
                               //WarpThreadArrange[0]='A'-'A'=0

```

```

//WarpThreadArrange[1]='A'-'A'=0
//WarpThreadArrange[2]='B'-'A'=1
//WarpThreadArrange[3]='B'-'A'=1
//WarpThreadArrange[4]='B'-'A'=1
BYTE *WeftThreadArrange; //纬纱排列顺序,规则同经纱排列
int nWarpDensity, nWeftDensity; //经纬纱的密度有几种,指以下两个数组的
元素个数
THREADDENSITY *WarpDensity; //经纱的密度
THREADDENSITY *WeftDensity; //纬纱的密度

public:
void LoadFromLstFile(FILE *lstFile); //从文件中导入*.lst 数据
int GetWarpThreadIndex(int thread); //求第 thread 根经纱在纱线排列中的
对应数值
int GetWeftThreadIndex(int thread); //求第 thread 根纬纱在纱线排列中的
对应数值
};
管理组织或纹版数据的类 MCWeave
class MCWeave : public CObject
{
public:
int nWeaveType; //0: 用组织数据模拟; 1: 用纹版数据模拟

int nWarpLoop, nWeftLoop; //当采用组织数据时,记录最小的组织循环由几
列几行组成
BYTE *arrWeave; //当采用组织数据时,存放组织矩阵
CWb *pWb; //当采用纹版数据时,存放纹版矩阵

void LoadFromWvFile(FILE *wvFile); //从*.wv 文件导入组织数据
BOOL IsLift(int warp, int weft); //计算第 warp 根经纱和第 weft 根纬纱的交织
//情形,返回 1 表示经纱覆盖纬纱;返回 0 表
//示纬纱覆盖经纱
};

```

管理织物图像管理类 MCSimImage

```

class MCSimImage
{
public:
    BITMAPINFO m_BitmapInfo;    // 模拟输出图像头
    LPBYTE m_lpDib;            // 图像内容

    // 根据图像的宽、高、分辨率创建图像
    BOOL CreateImage(int width, int height, int dpi=600, int bitcount=24);
    void ShowImage(CDC* pDC);    // 显示模拟输出
    BOOL SaveAsBMP(LPCTSTR lpszPathName);    // 保存织物模拟图像
};

```

实现纱线段绘制算法的类 **CRenderThreadSect**

```

class CRenderThreadSect : {
public:
    COLORREF m_OriginColor;    //原始的纱线颜色
    CSize m_Size;              //纱线段的长度,宽度
    int m_ThreadType;          // 0 :经纱; 1:纬纱
    int m_OriginX;             //需要绘制点的原始位置
    int m_OriginY;
    int m_DPI;                 //生成图像的分辨率
    float m_TransparentFact;    //生成毛羽效果的毛羽透明度参数

    float PerlinNoise2D(int x, int y); //计算某一点的 Perlin 函数
    bool ReadThreadSectInfo(MCThread *thread, CSize size, bool ThreadType,
int DPI, int x, int y, COLORREF &FuzzColor, float transparent_fact, float
FeatherLength, float FeatherDistance);

    //读入各数据
    void SetShadowElement(int x, int y); //计算添加阴影效果后像素点的颜色
    void SetColumnElement(int x, int y); //计算添加圆柱体效果后像素点的颜色
    void SetTwistElement(int x, int y); //计算添加捻度效果后像素点的颜色
    COLORREF SetTexture(int x, int y); //计算添加纱线贴图效果后像素点的颜色
    COLORREF Fuzz(int x, int y, COLORREF color); //增加毛羽效果后, 某一点
的颜色值
    COLORREF GetPixelColor(int x, int y); //计算该像素最终的输出颜色

```

};

存放模拟织物所需要的各种数据，模拟结果数据，和模拟方法的类 **MCFabric**

```

class MCFabric
{
public:
    void InitialFabrics(int maxwarp, int maxweft, int dpi, int mode);
                                                // 初始化织物属性数据
    void DoSimulation();                       // 开始模拟

private:
    void CreateFabricGrid();                 // 设置纱线栅格
    void CreateThreadEdge();                // 设置每一根经纱的左右边界、以及纬纱的上下
    边界
    void CreateCoverFactor();               // 建立各组织点的覆盖系数 CoverFactor, (用于
    3.3.1 中的优先级计算)
    void LocatePixelInGrid(float x, float y); // 确定织物图象上任一点(x,y)所在的
    栅格点
    int LocatePixelInWarpOrWeft(float x, float y, int recursion=4);
    // 确定织物图象上的任意一点(x,y)的绘制内容，经纱，纬纱还是空白
    COLORREF DrawPixelWithColor(int x, int y, int &light);
    // 确定织物图象上的任意一点(x,y)上绘制的颜色值
    int GetRelativePos(int x, int y, int &x0, int &y0, int warporweft, MCThread
    *pThread, CSize &size, COLORREF &FuzzColor, float &transparent_fact, float
    &FeatherLength, float &FeatherDistance);
    // 计算图像中某一点与所属的纱线段之间的相对位置

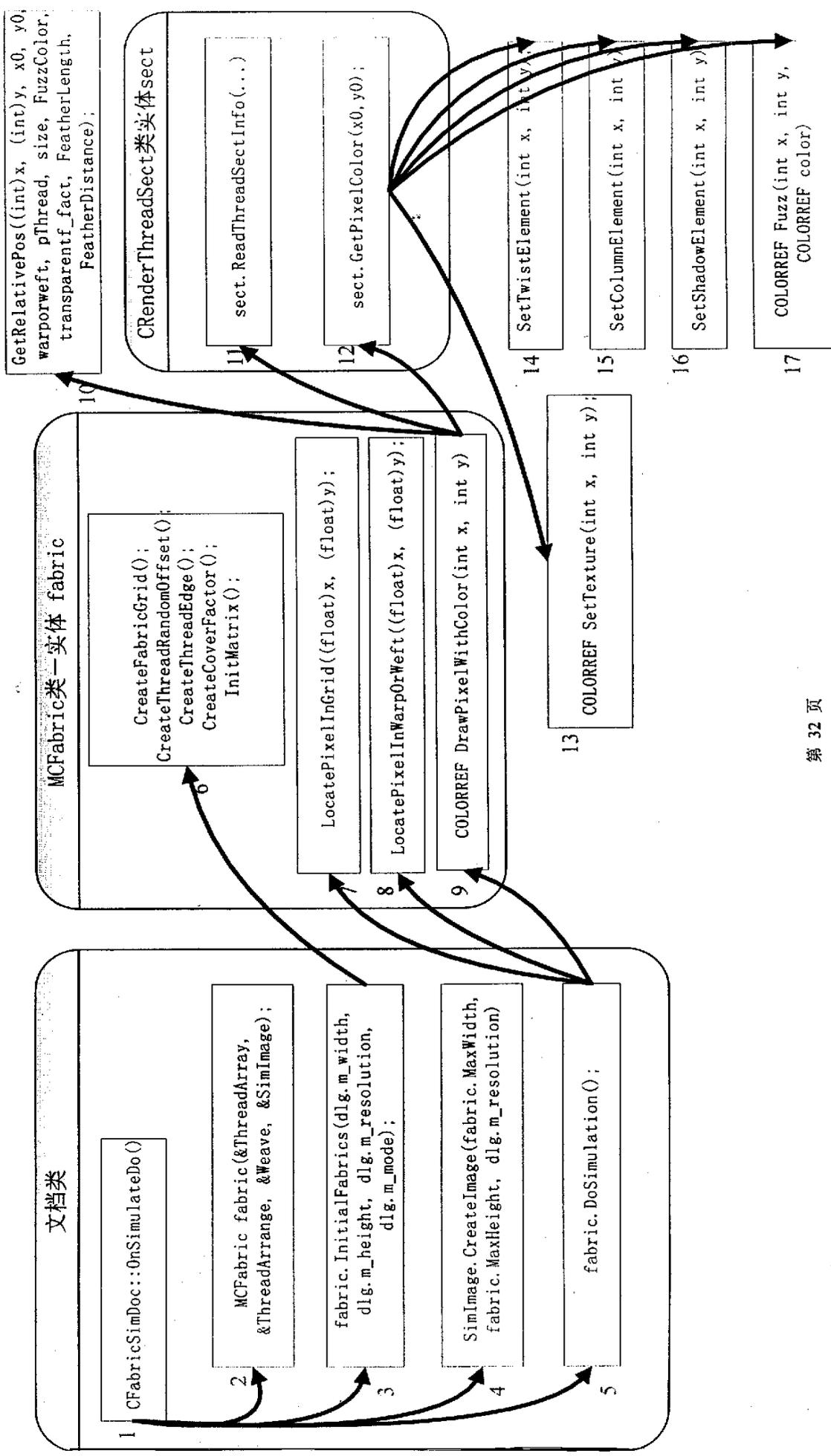
    MCFabric(MCThreadArray *pt, MCThreadArrange *pa, MCWeave *pw,
    MCSimImage *pi)           // 传递相关参数，构造 MCFabric 实体
private:
    MCThreadArray *pThArray;
    MCThreadArrange *pThArrange;
    MCWeave *pWeave;
    MCSimImage *pImage;
};

```

### 3.2.2.3 织物图像的逐点描绘

采用精细输出效果时，逐点描绘织物图像的模拟过程如下：

1. 开始模拟，调用 `CFabricSimDoc::OnSimulateDo()`
2. 过程 1 调用 `MCFabric fabric(...)`，读取织物模拟的各项参数到 `MCFabric` 类的对应成员变量 (`pThArray`, `pThArrange`, `pWeave`, `pImage`) 中
3. 过程 1 调用 `fabric.InitialFabrics()`，初始化织物的宽、高，分辨率和输出模式
4. 过程 3 调用一系列初始化函数  
`CreateFabricGrid();CreateThreadRandomOffset();`  
`CreateThreadEdge();CreateCoverFactor();InitMatrix();`
5. 根据输出图像的宽、高、分辨率初始化图像数据  
`SimImage.CreateImage(fabric.MaxWidth, fabric.MaxHeight,`  
`dlg.m_resolution)`
6. 调用 `fabric.DoSimulation()`，开始模拟，逐点绘制图像
7. 过程 6 调用 `LocatePixelInGrid((float)x, (float)y)`，判断当前绘制的象素处于哪个纱线段栅格
8. 过程 6 调用 `LocatePixelInWarpOrWeft((float)x, (float)y)`，判断当前点应该画经纱、纬纱还是空白
9. 过程 6 调用 `COLORREF DrawPixelWithColor(int x, int y)`，计算当前点输出颜色
10. 过程 9 调用 `GetRelativePos((int)x, (int)y, x0, y0...)`，计算当前的点与纱线段左下角的相对位置
11. 过程 9 调用 `sect.ReadThreadSectInfo(...)`，读入当前纱线的各项属性
12. 过程 9 调用 `sect.GetPixelColor(x0, y0)`，计算纱线段中某一点的输出颜色
13. 过程 12 调用 `COLORREF SetTexture(int x, int y)`，选择了用纱线贴图方式模拟时，纱线段中某一点的输出颜色
14. 过程 12 调用 `SetTwistElement(int x, int y)`，计算添加捻度效果后象素点的颜色
15. 过程 12 调用 `SetColumnElement(int x, int y)`，计算添加圆柱体效果后象素点的颜色
16. 过程 12 调用 `SetShadowElement(int x, int y)`，计算添加阴影效果后象素点的颜色
17. 过程 12 调用 `COLORREF Fuzz(int x, int y, COLORREF color)`，增加毛羽效果后，某一点的颜色值



## 3.3 核心算法

### 3.3.1 纱线互相覆盖的算法

在实际的织物中，当纱线排列比较紧密时，相邻的纱线可能存在互相覆盖的现象。这时，我们可以采用什么样的算法来模拟这种现象呢？

#### 3.3.1.1 同种类型的纱线相互覆盖：

下面以经纱覆盖经纱为例，介绍同种类型的纱线相互覆盖技术。将织物按照经纬纱的密度划分栅格，在某个纱线交织的组织点（假设是经组织点），经纱的轴心就是栅格的轴心；当经纱的直径比栅格要宽时（例如经纱纱线的密度是 50 根/厘米，平均每根纱线占了 0.2 毫米的宽度，而经纱的直径是 0.4 毫米/根），纱线的显示就有可能变化，

根据实物的观察，我们把输出的情形归纳一下，得出这样几类结论：

1. 在经组织点栅格只包含该组织点的经纱
2. 在经组织点栅格被相邻组织点的经纱覆盖
3. 在经组织点栅格既包括该组织点的经纱又包括相邻组织点的经纱

由于存在这样的规律：浮长越长的经纱，越不容易被覆盖；显示在织物表面的纱线，一定的浮长下出现出中间粗，两头细的形态。

由此，我们提出了如下算法（以经纱覆盖经纱为例）：

解决的问题：同种类型的纱线重叠，在重叠部分模拟输出时，应该显示什么

适用的条件：a) 左右相邻的组织点都是经组织点 b) 经纱的直径较大，存在着相互覆盖

算法描述：对每个组织点设置优先级；经浮长 = 1 的点，组织点的优先级为 0；经浮长 = 2 的点，组织点的优先级为 1；经浮长  $\geq 3$  的点，浮长两头的组织点的优先级为 1，其余的组织点优先级为 2。当前描绘的组织点存在经纱相互覆盖时，覆盖部分显示优先级高的组织点的纱线；当相邻的组织点优先级相同时，重叠的部分平分显示

应用举例：

纱线栅格及组织点的优先级设置：图 1

纱线排列：（部分经纱重叠）图 2

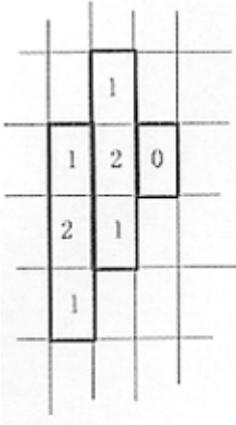


图 1

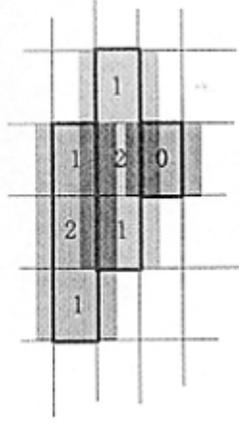


图 2

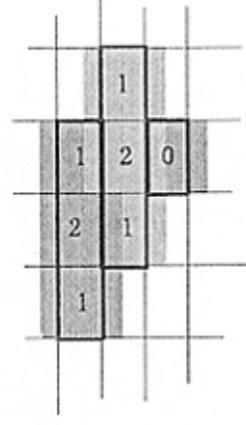
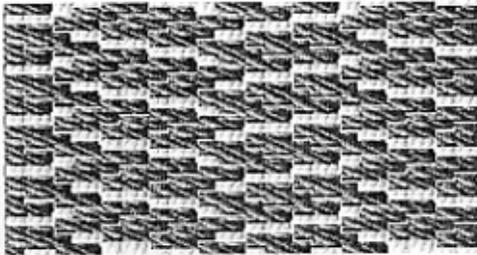


图 3

经过优先级算法计算后的输出结果：图 3  
应用效果展示：



### 3.3.1.2 不同类型的纱线互相覆盖：

不同类型的纱线互相覆盖有两种类型，即经纱覆盖纬纱，纬纱覆盖经纱。通过观察织物样本，我们发现这样的规律：

- 对于纬纱，在与经纱交织的浮长两端，部分纬组织点被经纱覆盖；
- 对于经纱，在与纬纱交织的浮长两端，部分经组织点被纬纱覆盖；

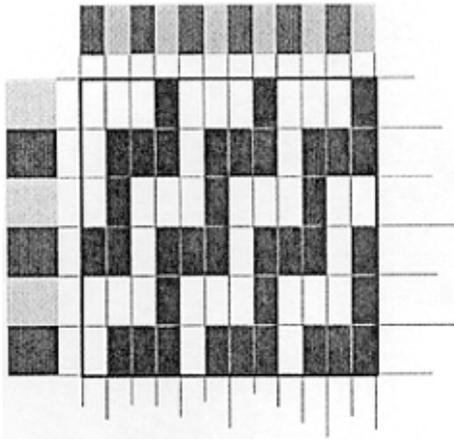
我们同样采用设置优先级的算法来模拟这种现象

对于经组织点，如果它的左边或者右边的组织点是纬组织点，则将该纬组织点的优先级降 1；

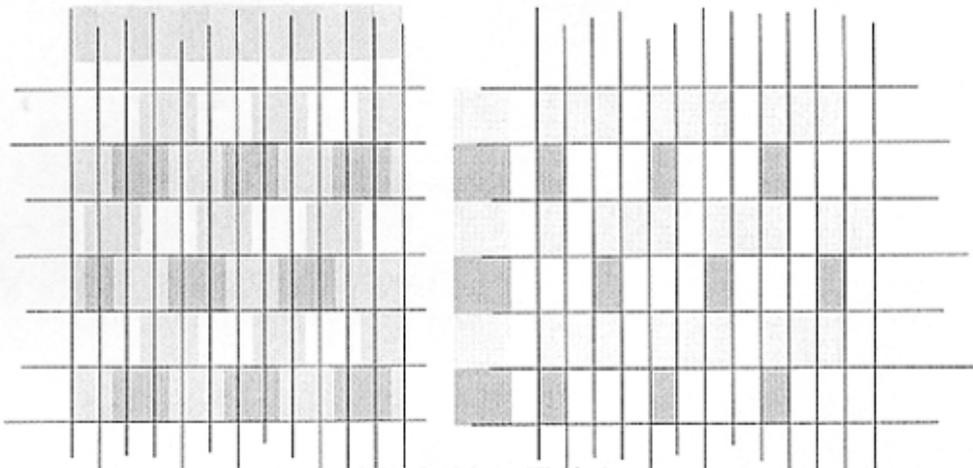
同样，对于纬组织点，如果它的左边或者右边的组织点是经组织点，则将该经组织点的优先级降 1；

结合以上两种算法的应用举例：

黑白组织及色纱排列

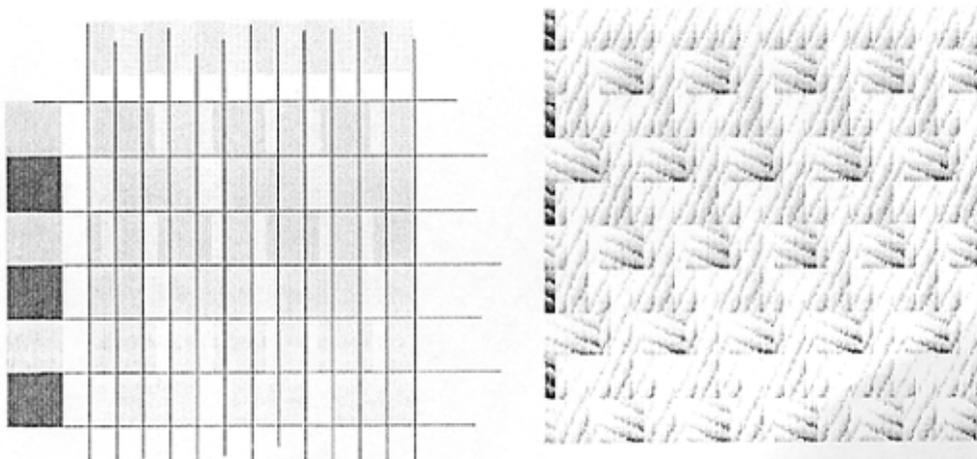


假设纬纱的栅格高度恰好等于纬纱的直径，而经纱的直径大于栅格的宽度。则两种颜色交替排列的经纱的原始位置（左图）和两种颜色交替排列的纬纱原始位置（右图）为：



根据以上两种优先级算法，最终的结果如下图所示：

左图（深色的部分体现了黑白组织），右图是同样参数模拟的结果



### 3.3.2 纱线的模拟方法

由于大部分纱线是由数量庞大的纤维构成的，其微观结构十分复杂。单根纤维的直径只有用精密仪器才能测量，所以除非放大或者以相当近的距离观察，否则单根纤维是微不足道、不易被察觉的。但是，正是纤维的聚集和它们的空间关系决定了纱线的视觉效果。

如何表示纱线在织物中的显示形态，是一个非常重要的问题。有些文献（参考文献 9）中采用了圆柱体或弯曲的椭圆柱体对纱线进行建模，然后采用了局部光照明模型来计算纱线表面上每一处的亮度值和颜色值。

采用圆柱体模型来模拟交织中的纱线，方法比较简单，但是，它并不能真实地表现交织中纱线的形态。因为，纱线并不是简单的圆柱体，它是由更细的纤维按照一定的捻度加捻而成，所以，它的表面并不是光滑的，而是有螺旋状的凹槽，在光线作用下，出现明暗相间的螺旋状纹理。

Xu(参考文献 8)在模拟针织物的过程中，采用了旋转的光片（Lumislice）模型来模拟针织物中的纱线，得到的效果非常的逼真，在实际应用中，光片（Lumislice）的生成，以及光片在织物形成中的应用，都是建立在三维结构之上的，有不小的计算量。

我们采用了一种比较简单的二维方法来模拟每一根纱线：

- a. 从视觉效果来讲，纱线的最基本外观是细长的圆柱体，假设纱线平放，其投影到二维平面就是一个长方形

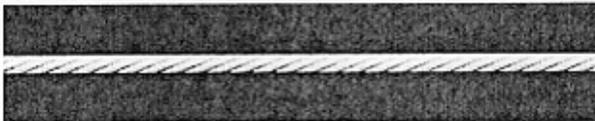


- b. 为了体现纱线的圆柱体的光照效果，对 a 生成的图像叠加渐变的纹理



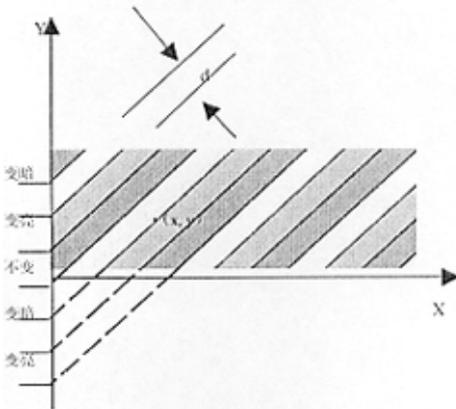
- c. 纱线的捻度纹理是由数根股线螺旋的缠绕而的。

在二维图像中，这种纹理可以近似的用颜色渐变，斜率相等的一组平行直线来体现。将捻度纹理叠加到 b 的图像中，得到



具体的方法是：

假设：用三种颜色等级模拟捻度纹理；纱线的捻度已知（可推得平行线的斜率为  $k$ ）；每捻的宽度为  $d$



- ① 读入纱线图象上某一点  $(x_0, y_0)$  的 RGB 颜色值  $(r, g, b)$
  - ② 点  $(x_0, y_0)$  所在的平行直线为  $y = kx + b$ ,  
 $b = y_0 - kx_0$ ；根据  $b$  的值所在的段的不同，决定点  $(x_0, y_0)$  在添加了捻度纹理之后是变亮、不变还是变暗，以及亮度变化的颜色级别。上图的例子中点  $(x_0, y_0)$  变亮一个颜色等级。所以  $(x_0, y_0)$  在添加了捻度纹理之后，颜色值变为  $(r+1 \times \text{grads}, g+1 \times \text{grads}, b+1 \times \text{grads})$  ( $\text{grads}$  是各个颜色级别之间颜色变化的幅度)
  - ③ 把  $(x_0, y_0)$  的新的颜色值存入图像
- d. 对  $c$  中产生的图像叠加毛羽纹理（其方法将在下一节重点介绍）



### 3.3.3 Perlin 噪声介绍(用于生成纱线毛羽效果)

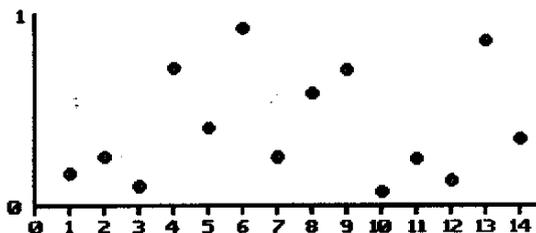
很多人在他们的程序中用随机数来创造不确定性，使得物体的运动、行为看起来更加自然或者生成各种各样的纹理。使用随机数来生成各种效果的方法当然有他们的用处，但是大部分时候他们的输出太粗糙以至于显得不自然。这里展示了一种用途广泛的方法，它能够用于生成几乎所有的自然纹理，而且，它的输出能够很容易地被应用到各种程序中。

当你在观察自然界的很多事物时，也许你已经注意到了它们是由不规则的几

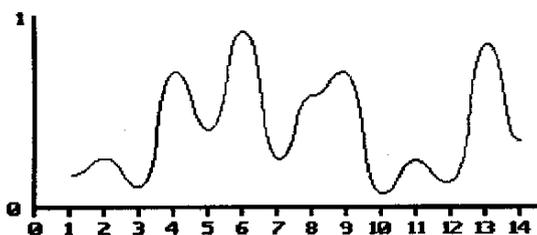
何图形构成的。这些几何图形的大小级别不同。一个典型的例子就是连绵排列山脉的外形轮廓。它包括了高度上的变化级别：巨大的变化级别表现了山脉，中等的变化级别表现了山丘，较小的变化级别表现了大石头，微小的变化级别表现了小石子...你可以继续不断的细化。观察自然界的事物：田野上杂草的分布，海中的水波，蚂蚁的运动，树枝的随风飘动，大理石上的花纹，风。所有这些现象都展示了同样的规律，事物的形态是由不同变化级别的同样的图案组合而成的。Perlin 噪声函数就通过在不同级别上的噪声函数叠加重构了这些自然的图案。创建一个 Perlin 噪声函数，你要做两件事：创建一个噪声函数，一个插值函数。

### 噪声函数

一个噪声函数从本质上来说就是一个有种子的随机数生成器。它有一个整型的参数，基于这个参数返回一个随机数。如果你用同一个参数输入两次，噪声函数两次产生相同的返回值。噪声函数的这一特性是非常重要的。否则，Perlin 函数将产生毫无意义的数。



这张例图是由一个噪声函数生成的。X 轴上的每个点都被赋予了一个介于 0 和 1 之间的随机数。

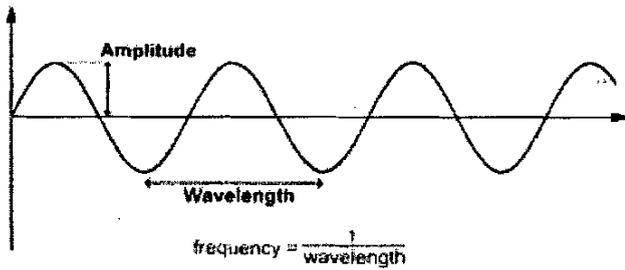


通过这些随机数之间的平滑插值，我们可以给出一个参数是非整型的连续函数。我们将在下文中讨论各种不同的插值方法。

### 振幅和频率的定义

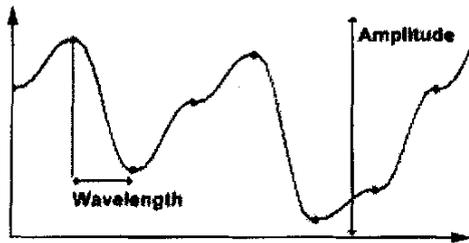
在我们继续向下讨论之前，让我来定义一下本文中用到的振幅和频率的含义。

● Sin 函数



Sin 函数的波长是一个波峰到另一个波峰之间的距离。振幅是波峰的高度。频率被定义为  $1/\text{波长}$ 。

● 噪声函数



在这个噪声函数的例图当中，红色的点指示出了随着自变量的变化而变化的随机数值。例图中，振幅是函数曲线最大值和最小值之间的差距。波长是一个红点到下一个红点的距离。同样的，频率被定义为  $1/\text{波长}$ 。

创建 Perlin 噪音函数

现在，假设我们已经有了以下光滑连续的函数，它们各自的频率和振幅不同，当你把所有这些函数叠加起来的时候，你就得到了一个非常漂亮的噪声函数，那就是 Perlin 噪声函数。

假设已经有如下噪声函数

Amplitude : 128  
frequency : 4

Amplitude : 64  
frequency : 8



Amplitude : 32  
frequency : 16



Amplitude : 16  
frequency : 32



Amplitude : 8  
frequency : 64



Amplitude : 4  
frequency : 128



当把以上 6 个函数叠加，你就可以得到下图

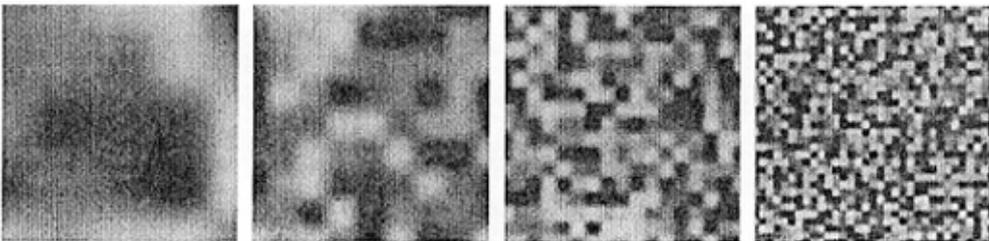
Sum of Noise Functions = ( Perlin Noise )

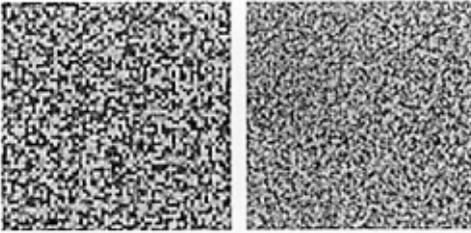


你可以看到这个函数有着大、中、小等不同的曲线变化级别。你甚至能够把他想象成一系列山脉的轮廓。事实上很多应用程序就是用这种方法来生成山水风景的，他们用了二维的噪声，我们接下来也会讨论到。

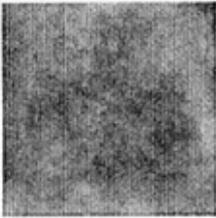
当然你也可以在二维空间中做同样的事。

一些二维空间中创建的噪声函数





把以上二维函数叠加，将产生如下噪声图案



#### 持续性参数 (persistence)

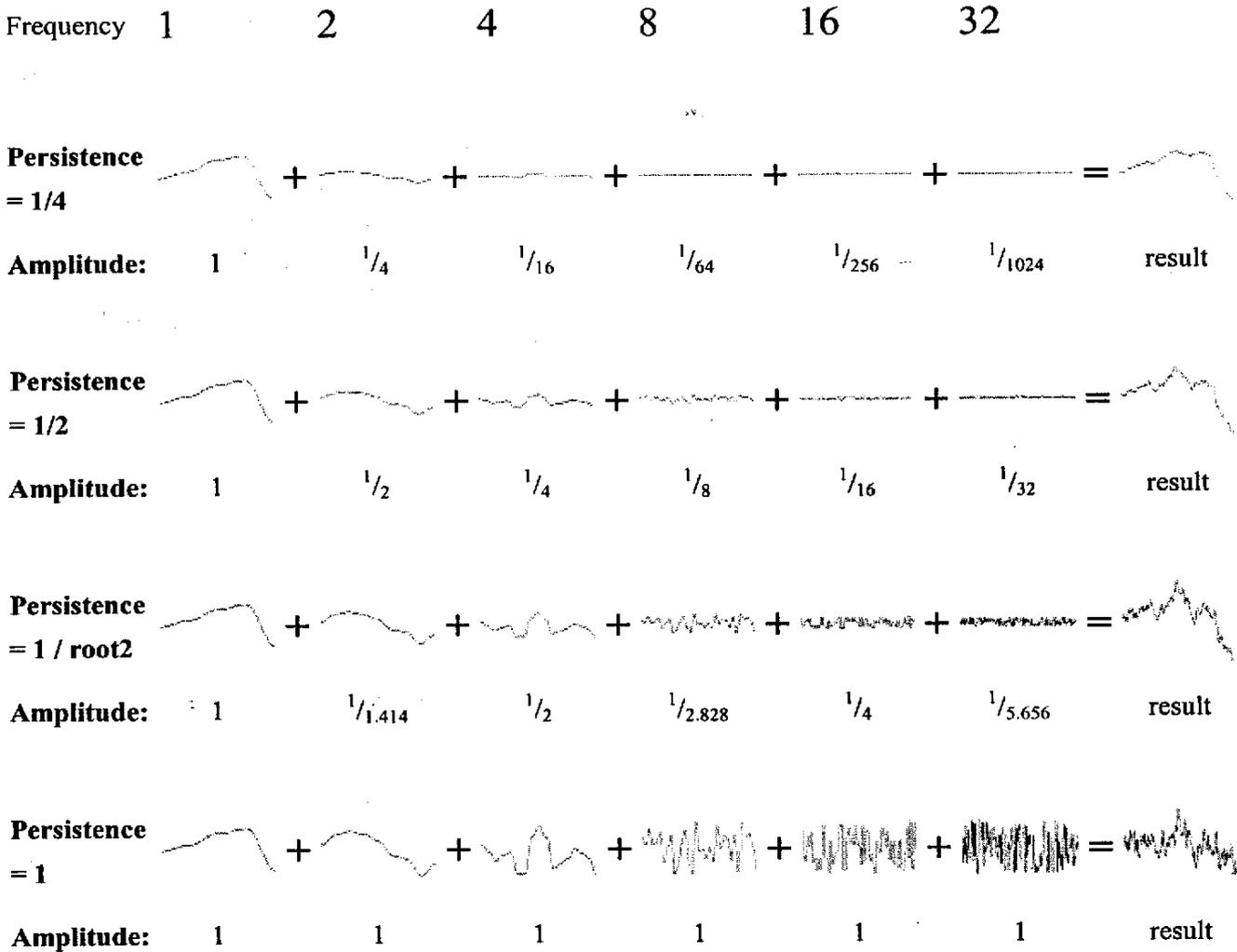
当你在叠加这些噪声函数的时候，你可能想知道各个噪声函数的振幅和频率是多少。一维的例子中，在一系列被叠加的图案之间，我们采用了频率加倍、振幅减半的措施。当然，你也可以在不同的噪声函数之间创建不同的频率和振幅变化步幅。举例来说，创建一个光滑起伏的山丘，你可以把振幅大、频率低的噪声函数和振幅很小、频率很高的噪声函数叠加。当你想创建相对平坦但是多岩石的山丘轮廓时，可以使用较小的振幅和较低频率。

为了说得更明白，而不是老是重复频率、振幅这几个词，我们使用了持续性参数这个值用来指示每个频率下的振幅。这个词最早是被 Mandelbrot 发明的，他是分形技术的创始人之一。根据他的定义，有很多高频信号的噪声持续性参数的数值较低。而我们对持续性参数的定义是这样的：

$$\text{frequency} = 2^i$$

$$\text{amplitude} = \text{persistence}^i$$

这里的  $i$  指的是叠加中用到的第  $i$  个噪声函数。下图展示了不同的持续性参数对 Perlin 噪声的输出产生的影响。每个 Perlin 函数都是 6 个源噪声函数的叠加。我们能够从下图中总结出随着持续性参数的变化，Perlin 函数的变化规律。



### Octaves

Octaves 指的是一系列被叠加的噪声函数。之所以这样称呼是因为每个噪声函数都是前一个噪声函数的频率加倍所得，而在音乐中，Octaves 有着相同的特性。事实上，究竟要叠加多少个 Octaves 完全取决于你自己。你可以选择增加或减少 Octaves。然而，我可以给些建议。如果是打算用 Perlin 函数在计算机屏幕上绘制图像，屏幕的象素点是最基本的单位，所以当 Octaves 的频率高于象素点的频率时，效果将变得不可见，屏幕象素可能不足以显示所有这些高频噪声函数的微小细节。因此，有些 Perlin 噪声采用了这样的方法：叠加尽可能多的噪声函数直到到达屏幕显示分辨率的极限。

另一种好的办法是叠加噪声函数直到它们的振幅足够小以至于不能产生什么效果。事实上什么时候振幅能够达到足够小取决于持续性参数的数值、Perlin 函数的总体振幅和屏幕的分辨率。

## 创建噪声函数

我们需要怎样的噪声函数呢？噪声函数从本质上来说就是一个随机数生成器。然而它并不像平时我们程序中用到的其他随机数生成器一样，每次调用的时候都生成一个不同的随机数。这个噪声函数需要的随机数是由一个或多个参数计算生成的。特别的，当每次你输入相同的参数时，它计算产生相同的返回值。而当输入的参数不同时，它的返回值是不同的随机数。

现在我们已经清楚了我们需要怎样的随机数生成器，下一步就是去找这样一个函数。下面的代码就是一个符合要求的函数。它的返回值是介于-1.0 和 1.0 之间的一个浮点数。

```
function IntNoise(32-bit integer: x)

    x = (x<<13) ^ x;

    return ( 1.0 - ( (x * (x * x * 15731 + 789221) + 1376312589) & 7fffffff)
/ 1073741824.0);

end IntNoise function
```

如果你需要一个不同的随机数生成器，我建议你复制上述函数，然后改写一些数字。那些看起来很大的数字都是质数，所以修改以后的数字也应该是质数。你可以写一个质数生成器的函数来完成这一任务。

## 创建插值函数

在完成了噪声函数之后，我们要做的是把它的输出变得连续而平滑。这里也有很多可供选择的方法，当然产生的效果也各不相同。一个标准的插值函数有 3 个参数：a,b,x。a 和 b 是插值函数的两个端点，x 是 0 到 1 之间的一个数。插值函数根据 x 的值，返回一个 a 到 b 之间的数。如果 x 等于 0 它返回 a，如果 x 等于 1，它返回 b，如果 x 介于 0 和 1 之间，那么，它返回 a 和 b 之间的一个数。

### 线性插值



用线性插值的效果看起来不太好，不过很多人用这种方法来画轮廓。这种的算法简单快速，而且在实际应用中，它至少是可行的。

```
function Linear_Interpolate(a, b, x)

    return a*(1-x) + b*x

end of function
```

### Cosine 插值



显而易见，用这种方法绘制的曲线比用线性插值方法绘制的曲线更加光滑，所以它是一种更好的方法。虽然它会稍稍降低一些运行效率，但是基于它产生的效果，这么做是值得的。

```
function Cosine_Interpolate(a, b, x)

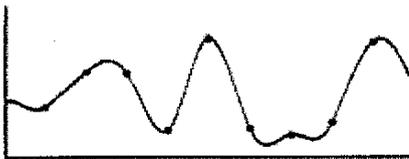
    ft = x * 3.1415927

    f = (1 - cos(ft)) * .5

    return a*(1-f) + b*f

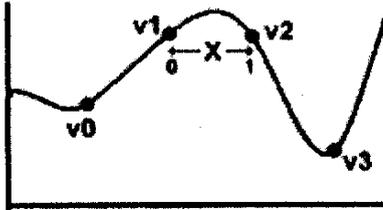
end of function
```

### 三次插值



用这种方法可以给出非常光滑的曲线，但是同时我们也得为此付出代价：运行速度的损失。我不能确定它比用 Cosine 方法具体好在哪里，总之它是可选的方法之一。这种方法何止浅的方法相比有点复杂：以前的方法都用到了 3 个参数，而三次插值方法共用到了 5 个参数。

我们用 V0、V1、V2、V3 来代替 a 和 b, x 的含义和之前相同



$V_0 = a$  之前的一点

$V_1 = a$  点

$V_2 = b$  点

$V_3 = b$  之后的一点

```
function Cubic_Interpolate(v0, v1, v2, v3, x)
```

```
    P = (v3 - v2) - (v0 - v1)
```

```
    Q = (v0 - v1) - P
```

```
    R = v2 - v0
```

```
    S = v1
```

```
    return Px3 + Qx2 + Rx + S
```

```
end of function
```

### 让噪声变的光滑

出了插值方法以外，你也可以采用光滑噪声函数的输出来使它看起来不那么粗糙，体现在二维和三位空间中就是减少方格的出现率，或者说是减少锯齿。我们能够料想到光滑输出是可行的。任何写过图像模糊滤镜或者火焰算法的人都会对这一过程很熟悉。

把噪声函数中的一系列点联系起来，取各点和它相邻点的平均值，就是一种光滑的方法，代码如下：

一维光滑噪声：

```
function Noise(x)
```

```
    ...
```

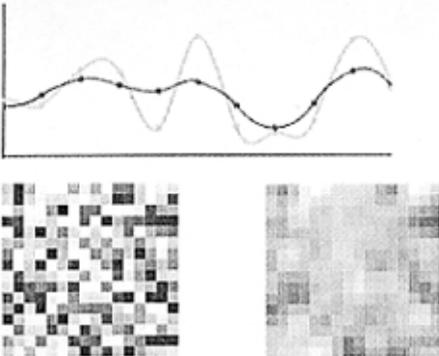
```
end function
```

```
function SmoothNoise_1D(x)
    return Noise(x)/2 + Noise(x-1)/4 + Noise(x+1)/4
end function
```

## 二维光滑噪声

```
function Noise(x, y)
    ...
end function

function SmoothNoise_2D(x, y)
    corners = ( Noise(x-1, y-1)+Noise(x+1, y-1)+Noise(x-1, y+1)+Noise(x+1,
y+1) ) / 16
    sides = ( Noise(x-1, y) +Noise(x+1, y) +Noise(x, y-1) +Noise(x, y+1) )
/ 8
    center = Noise(x, y) / 4
    return corners + sides + center
end function
```



上面是一维、二维光滑效果前后比对。我们能够观察到光滑后的噪声是比较令人满意的，没有像光滑前的噪声那样极端（黑白分明），噪声变化的频率被减半了。这种方法更多的被用在二、三维的空间中，它的作用是减少噪声产生的锯齿形。不幸的是，这种方法同时也降低了对比度，越是光滑，对比度降的越多。

将上述内容整合

现在我们已经了解了多种噪声函数和插值方法，接下来要做的就是把它们整合起来，创建一个 Perlin 噪声函数。这个过程事实上是几个插值噪声函数的叠加。也就是说，Perlin 噪声只是一个函数，你传给它一个或多个参数，它返回一个值。以下是一个简单的 Perlin 函数。

Perlin 函数的主体是一个循环。每个循环体所做的是将各个 Octave 叠加，然后频率加倍。各个循环体都调用到了不同的噪声函数，把第  $i$  个 Octave 用到的噪声函数的记为 Noise  $i$ 。现在，我们不必为每个 Octave 都写一个噪声函数，因为所有的噪声函数本质上都是相同的，都用到了 3 个很大的质数。当然你可以重用相同的一段代码，或者改变一下 Noise  $i$  中的质数

一维 Perlin 函数代码

```
function Noise1(integer x)
    x = (x<<13) ^ x;
    return ( 1.0 - ( (x * (x * x * 15731 + 789221) + 1376312589) & 7fffffff)
/ 1073741824.0);
end function
function SmoothedNoise_1(float x)
    return Noise(x)/2 + Noise(x-1)/4 + Noise(x+1)/4
end function
function InterpolatedNoise_1(float x)
    integer_X = int(x)
    fractional_X = x - integer_X
    v1 = SmoothedNoise1(integer_X)
    v2 = SmoothedNoise1(integer_X + 1)
    return Interpolate(v1 , v2 , fractional_X)
end function

function PerlinNoise_1D(float x)
    total = 0
    p = persistence
    n = Number_Of_Octaves - 1
    loop i from 0 to n
        frequency = 2i
        amplitude = pi
        total = total + InterpolatedNoisei(x * frequency) * amplitude
    end of i loop
    return total
end function
```

## 二维 Perlin 函数代码

现在，我们很容易就能把同样的代码用到创建二维或更多维数的 Perlin 噪声

## 函数中

```

function Noise1(integer x, integer y)
  n = x + y * 57
  n = (n<<13) ^ n;
  return ( 1.0 - ( (n * (n * n * 15731 + 789221) + 1376312589) & 7fffffff) /
1073741824.0);
end function

function SmoothNoise_1(float x, float y)
  corners = ( Noise(x-1, y-1)+Noise(x+1, y-1)+Noise(x-1, y+1)+Noise(x+1, y+1) ) / 16
  sides   = ( Noise(x-1, y)  +Noise(x+1, y)  +Noise(x, y-1)  +Noise(x, y+1) ) / 8
  center  = Noise(x, y) / 4
  return corners + sides + center
end function

function InterpolatedNoise_1(float x, float y)
  integer_X   = int(x)
  fractional_X = x - integer_X
  integer_Y   = int(y)
  fractional_Y = y - integer_Y
  v1 = SmoothedNoise1(integer_X,      integer_Y)
  v2 = SmoothedNoise1(integer_X + 1, integer_Y)
  v3 = SmoothedNoise1(integer_X,      integer_Y + 1)
  v4 = SmoothedNoise1(integer_X + 1, integer_Y + 1)

  i1 = Interpolate(v1 , v2 , fractional_X)
  i2 = Interpolate(v3 , v4 , fractional_X)

  return Interpolate(i1 , i2 , fractional_Y)
end function

function PerlinNoise_2D(float x, float y)
  total = 0
  p = persistence
  n = Number_Of_Octaves - 1
  loop i from 0 to n
    frequency = 2i
    amplitude = pi

    total = total + InterpolatedNoise_1(x * frequency, y * frequency) * amplitude
  end of i loop
  return total
end function

```

## Perlin 噪声的应用

在完成了 Perlin 函数之后，我们能用它来做些什么呢？

### 在一维空间中的应用：

**模拟生物的运动轨迹：**有生命的对象很少长时间静止不动。将 Perlin 函数应用在游戏中的人物身上，时不时的调整一个虚拟人物的关节点的位置，使得他看起来更加生动。

**模拟手绘线条：**用计算机绘制的线条是笔直的，也许看起来不自然、不友好。我们可以把 Perlin 噪声应用到一个摇摆画线算法中，使得用计算机绘制的线条更加像人手绘制的线条。

### 在二维空间中的应用：

**山水画** 这是对二维 Perlin 函数的完美应用。它不像细分算法，需要把每次循环的结果保存在内存中。每一点的绘制都非常简单。色块的边界具有不规则性，而且能绘制出微小的细节。这体现了自然界的物体具有多重变化级别的特性。绘制的属性也很容易设置。

**云朵** Perlin 函数也同样适用于云朵的绘制。

**材质生成：**用 Perlin 噪声能够绘制出各种各样的材质效果，而且可以根据需要设定生成的图像的大小。这比重复平铺一定大小的材质图像产生的效果更加接近自然。

### 在三维空间中的应用：

**三维云彩：**我们可以用 Perlin 函数生成三维的云彩，当然，同时必须用到光线跟踪算法。

**云彩动画：**用三维 Perlin 函数生成 随着时间轴的变化而变化的二维云彩图像，即云彩的动画效果

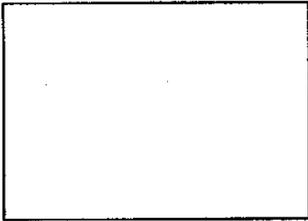
**体纹理：**一些光线跟踪程序用逐步雕刻的方法来生成三维纹理。这种方式不会产生像用二维纹理三维贴图方法时出现纹理扭曲变形的情况。

### 在四维空间中的应用：

**有动画效果的三维文理和云彩：**把 4 维 Perlin 函数看作是随时间轴变化而变化的三维云彩或纹理



上图中的陆地，云，水都是由 Perlin 函数生成的 (by Terragen)



用 3D Perlin 噪声生成的云彩动画

用 Perlin 噪声生成纹理

用 Perlin 函数来生成纹理能够生成各种奇异的纹理。而且生成无限大的纹理时只需占用很小的内存空间。无论是大理石、木材还是漩涡状的图案，几乎所有的纹理都能用 Perlin 方式来绘制。通过雕刻一块实体材料，还可以定义三维的体纹理。这样就可以为任何形状的物体定制纹理，而不用担心纹理变形的问题。尝试着改变持续性参数和频率，就可以改变纹理的属性。以下是一些用 Perlin 函数创建的纹理：

	<p>标准的三维 Perlin 函数，由 4 个函数叠加而成，持续性参数分别为 0.25 和 0.5</p>
	<p>用小的持续性参数可以创造出鲜明的边缘</p>



混合几个 Perlin 函数,可以得到有趣而复杂的纹理。这个纹理是由两部分组成的:一个持续性参数较小的 Perlin 函数(记为 P1)用于绘制水滴的形状,另一个持续性参数较大的函数用于绘制底纹。其中 P1 又是有另外两个函数组成的:一个绘制了条纹,一个绘制了斑点。



大理石底纹理可以用 Cos 函数来模拟

```
texture = cosine( x + perlin(x, y, z) )
```



用 Perlin 函数绘制木纹

```
g = perlin(x, y, z) * 20
```

```
grain = g - int(g)
```

木头上的肿块纹理是高频噪声在某一方向上的伸展造成的

```
bumps = perlin(x*50, y*50, z*20)
```

```
if bumps < .5 then bumps = 0 else bumps = 1t
```

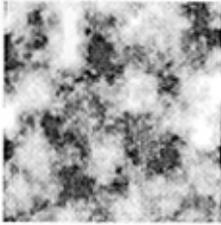


### 3.3.4 将 Perlin 算法应用到纱线中,生成纱线的毛羽效果:

纱线毛羽的生成大致分为以下几个步骤:

#### 1. 生成 Perlin 噪声纹理图像

Perlin 函数是由 Perlin 在八十年代提出的,它可以在某一空间生成连续的噪声。如今,Perlin 函数被成功地应用在大理石、火焰及云彩等自然纹理的模拟中。简单的说,Perlin 函数是一系列噪声函数的叠加,这些噪声函数频率递增,振幅递减。二维 Perlin 函数输入的是二维平面中某点的坐标  $(x,y)$ ,输出的是属于 $[-1,1]$ 的一个浮点数  $f$ 。将  $(f \times 128 + 127)$  作为  $(x,y)$  的 RGB 颜色值得到了下图。



(参数设置: 初始频率  $f = 1/30$ ; persistence = 0.5)

#### 2. 生成毛羽纹理图像

对于二维坐标系中的每一个点计算它的 Sobel 梯度算子。计算方法如下:  
 设要计算坐标点  $(x,y)$  的梯度,

$p0_{(x-1,y-1)}$	$p1_{(x,y-1)}$	$p2_{(x+1,y-1)}$
$p3_{(x-1,y)}$	$(x, y)$	$p4_{(x+1,y)}$
$p5_{(x-1,y+1)}$	$p6_{(x,y+1)}$	$p7_{(x+1,y+1)}$

先计算  $(x,y)$  周围 8 个点 Perlin 函数值, 记为  $p0 \sim p7$ ;

然后, 计算

$$\text{buf1} = p0 + 2 * p1 + p2 - p5 - 2 * p6 - p7;$$

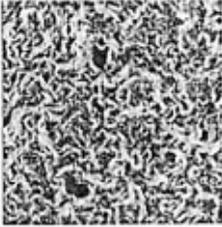
$$\text{buf2} = p7 + 2 * p4 + p2 - p5 - 2 * p3 - p0;$$

$$\text{noise\_f} = (|\text{buf1}| + |\text{buf2}|) / 2;$$

if(noise\_f > 1)

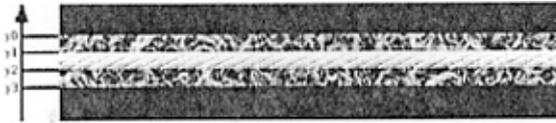
$$\text{noise\_f} = 1.0;$$

计算的结果得到  $\text{noise}_f \in [0,1]$ 。把  $\text{noise}_f$  作为不透明度的参数,当  $\text{noise}_f = 0$ , 完全显示背景色;  $\text{noise}_f = 1$ ,完全显示前景色, 当  $\text{noise}_f \in (0,1)$ ,显示的颜色是  $\text{noise}_f \times \text{前景色} + (1 - \text{noise}_f) \times \text{背景色}$ 。设前景色为白色, 背景色为黑色, 得到图象:



### 3. 毛羽纹理叠加

根据纱线的毛羽长度  $d$ ,把  $b$  步骤中生成的图像作为纹理叠加到纱线图像上。实际显示的颜色为:  $\text{noise}_f \times \text{前景色} + (1 - \text{noise}_f) \times \text{背景色}$  (前景色为白色, 背景色为纱线图像的颜色)。

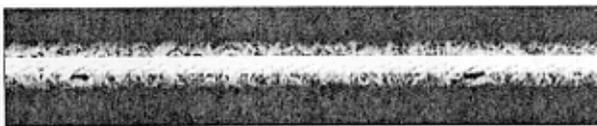


4. 为了体现毛羽边缘渐淡的效果, 将毛羽纹理从  $y_1$  到  $y_0$ ,从  $y_2$  到  $y_3$  做线形插值, 使得毛羽边缘的不透明度递减



### 5. 控制毛羽度的方法探索

由以上步骤可得毛羽度主要由  $\text{noise}_f$  控制, 在  $\text{noise}_f$  接近 1 的地方, 毛羽较密; 在  $\text{noise}_f$  接近 0 的地方, 毛羽较疏。所以要改变毛羽度, 可以在步骤 3 之前对  $\text{noise}_f$  开方, 使得  $\text{noise}_f$  的值域仍旧在  $[0, 1]$ , 但是平均值变大。  
 $\text{noise}_f$  开平方的效果:



$\text{noise}_f$  开三次方的效果:



## 6. Perlin 函数的一个其他应用

在织物模拟的过程中偶然发现，若将步骤 2 中的公式改为

```
buf1 = p0 + 2*p1 + p2 - p5 - 2*p6 - p7;
```

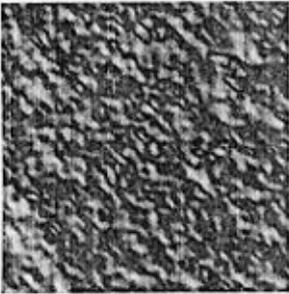
```
buf2 = p7 + 2*p4 + p2 - p5 - 2*p3 - p0;
```

```
noise_f = (|buf1| + |buf2|)/2;
```

```
if(noise_f > 1)
```

```
    noise_f = 1.0;
```

则 perlin 函数能够模拟出逼真的岩石效果：



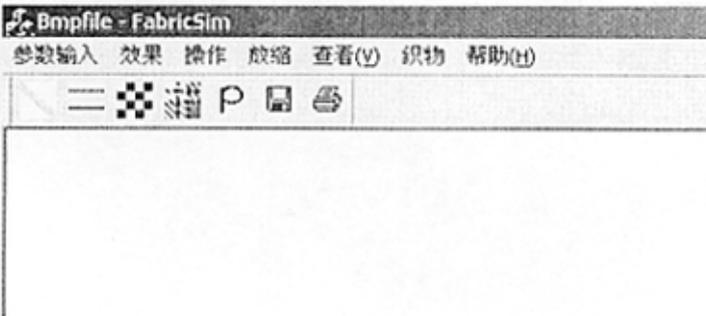
### 3.4 输入数据描述及软件使用说明书

织物仿真系统是利用计算机图形技术，将设计人员的意图以一种具体的算法转换组织，以模拟的方法快速、形象、直观地在显示器上显示出来的系统。设计人员在计算机上输入织物组织、纱线排列、纱线种类等数据后，计算机会自动生成模拟的实物图像。本系统使设计人员可以简单、方便地改变织物的组织、纱线排列，随心所欲地对织物模拟图上各种颜色的经纬纱线进行调色配色，淋漓尽致地将设计思想表达出来，在织物上机织造之前就能看到实际织造结果，将设计人员从繁杂的劳动中解放出来、使之能够将更多的时间和精力投入到产品构思及开发中去。

#### 3.4.1 安装

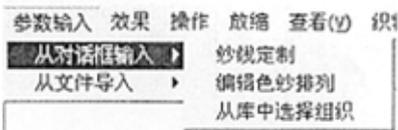
复制可执行文件 FabricSim 和数据文件夹 data

##### ● 运行界面



菜单项目

##### 1. 参数输入:



两种参数输入方法:

- 对话框输入，依次分别填写“纱线定制”、“编辑色纱排列”、“从库中选择组织”三个对话框；
  - 从文件导入，分别导入“纱线”、“色纱排列”“组织或纹版”三类文件
2. 效果:

4 种输出效果：黑白组织、配色模纹、色纱排列、精细效果

3. 操作：

开始模拟/暂停

4. 缩放：

放大/缩小

5. 查看：

显示/隐藏 工具条和状态栏

6. 织物：

保存织物输出文件

打印织物输出文件

7. 帮助：

工具条项目与菜单项目的对应



参数输入→从对话框输入→纱线定制



参数输入→从对话框输入→编辑色纱排列



参数输入→从对话框输入→从库中选择组织



操作→开始模拟



以纱线贴图方式模拟（菜单中没有该项的对应项目）



织物→保存织物输出文件



织物→打印织物输出文件

### 3.4.2 操作与演示

#### 3.4.2.1 运行

双击 FabricSim 文件

#### 3.4.2.2 输入想要模拟的织物的各项参数

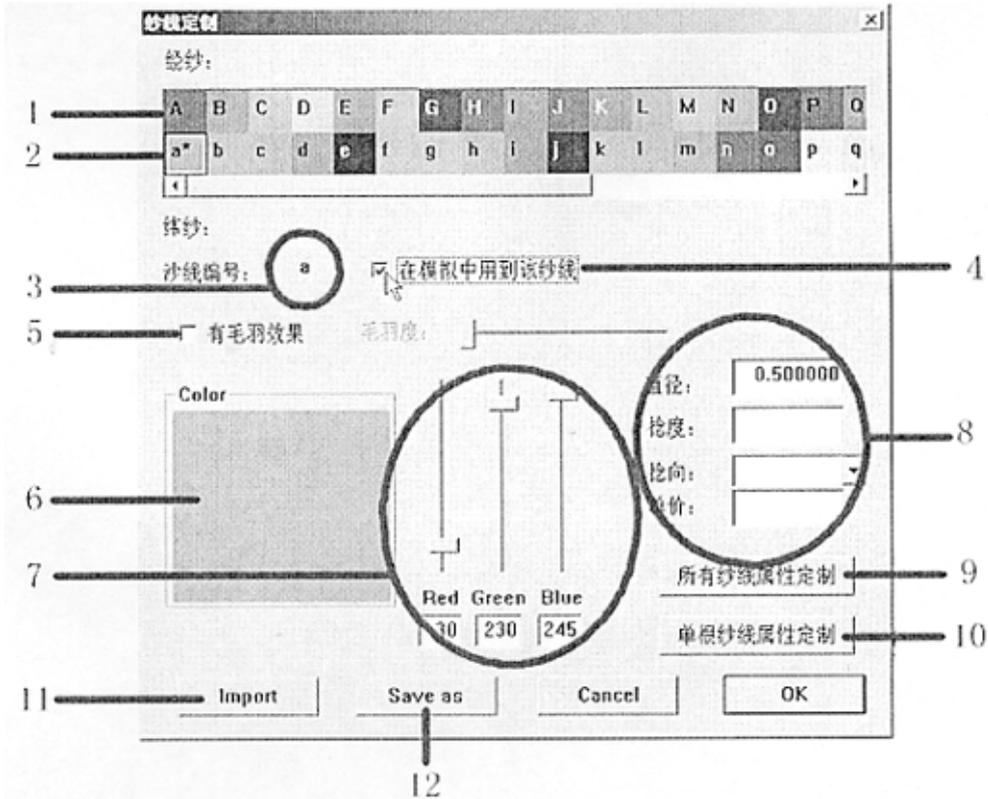
要完成一块纺织面料的模拟,首先要对织物的各项属性进行合理的定制,并非任意参数的输入都能产生正确结果,例如纱线排列的过密将导致生产和模拟上

的不可行。

本软件在设置织物参数是提供了两类方法：通过对话框输入的方法和通过文件导入的方法。无论哪一类方法都可以定制织物模拟的参数。

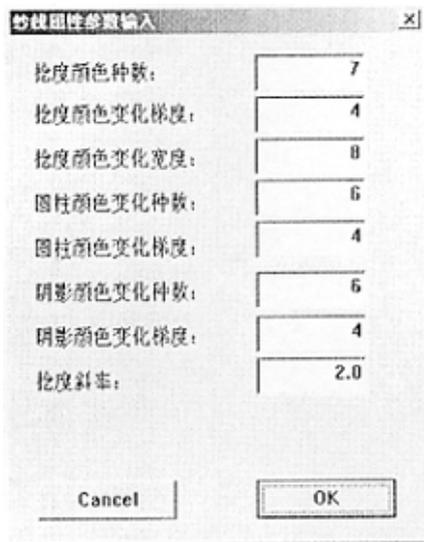
### 3.4.2.2.1 定制纺织织物中需要的各种纱线

a. 对话框输入



1. 所有经纱的颜色，每种经纱都用一个大写的字母来表示。字母右边如果有星号，代表在织物模拟中用到了这根纱线。点击列表中的某个项目就可以对单根经纱的属性进行定制
2. 所有纬纱的颜色，每种纬纱都用一个小写的字母来表示。字母右边如果有星号，代表在织物模拟中用到了这根纱线。点击列表中的某个项目就可以对单根纬纱的属性进行定制

3. 当前正在编辑的纱线编号，大写字母表示经纱，小写字母表示纬纱
4. 选择是否要在模拟中用到当前正在编辑的纱线
5. 选择是否为当前正在编辑的纱线添加毛羽效果
6. 当前正在编辑的纱线的颜色
7. 通过滑动条或者直接键入数字的方法输入当前正在编辑的纱线的颜色（使用 RGB 颜色模式，颜色值 $\in [0,255]$ ）
8. 纱线的直径（浮点数，单位是 cm，缺省值为 0.5cm）、捻度（浮点数）、捻向{0, 1, -1}和单价（浮点数）属性
9. 统一定制所有纱线的显示效果属性，这些属性实际上反映了纱线的材质



参数名称	参数意义	参数取值范围	缺省值
捻度颜色种数	用几种深浅不同的颜色来模拟纱线捻度效果	整数：[1,7]	7
捻度颜色变化梯度	体现纱线捻度效果的颜色深浅颜色变化幅度	整数：[0,8]	4
捻度颜色变化宽度	捻度效果中每种颜色的像素宽度	整数：(0,20]	8
圆柱颜色变化种数	用几种深浅不同的颜色来模拟纱线圆柱体的效果	整数：[1,6]	6
圆柱颜色变化梯度	体现纱线圆柱体形态的颜色深浅颜色变化幅度	整数：[0,8]	4

阴影颜色变化种数	用几种深浅不同的颜色来模拟纱线交织时的阴影效果	整数: [1,6]	6
阴影颜色变化梯度	体现纱线阴影效果的颜色深浅颜色变化幅度	整数: [0,8]	4
捻度斜率	纱线方向与捻度方向的夹角的 tan	浮点数 (0, 10)	2.0

10. 定制单根纱线的显示效果属性, 和统一定制显示效果时的方法一样, 但是只对被激活的单根纱线有效
11. 从文件导入纱线数据, 显示出导入纱线的各项属性, 作用和直接从菜单导入纱线属性文件相同
12. 保存纱线属性文件

b. 编辑文件并导入

用纯文本方式编辑或创建一个\*.thr 文件。

例如 Chewan.thr 这样一个文件:

经纱只有一种

这种经纱 (编号为 A) 的颜色 (RGB 模式)

这种经纱 (编号为 A) 的直径 (cm/根)

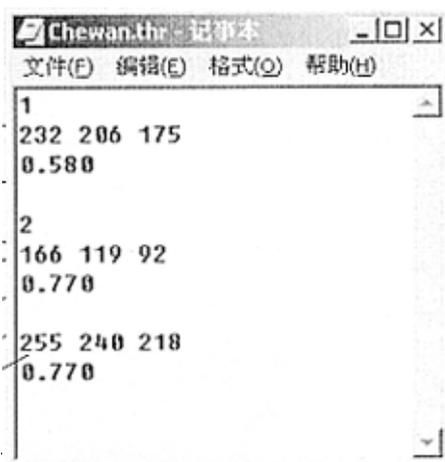
纬纱有两种

第一种纬纱 (编号为 a) 的颜色

第一种纬纱 (编号为 a) 的直径

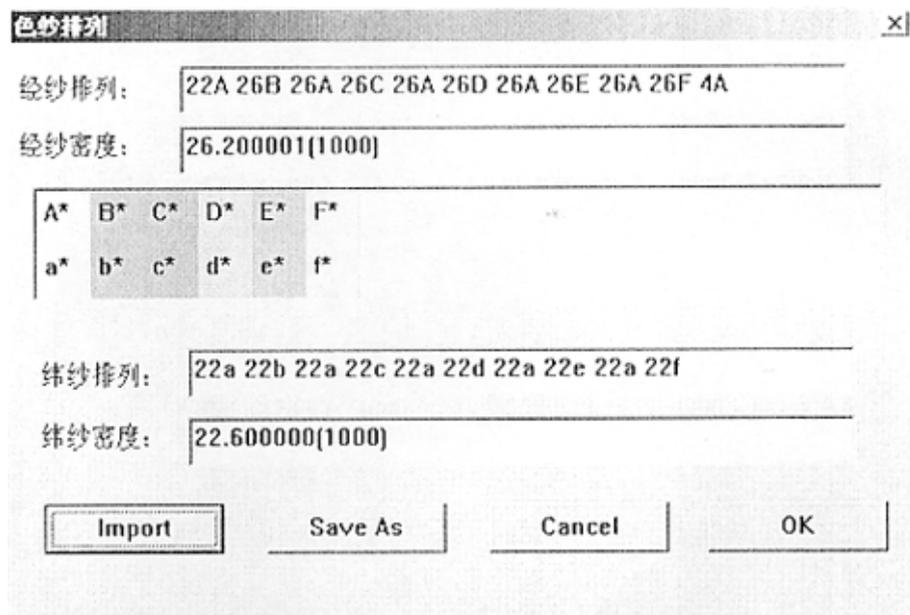
第二种纬纱 (编号为 b) 的颜色

第二种纬纱 (编号为 b) 的直径



### 3.4.2.2.2 使用 3.4.2.2.1 中选中的纱线编辑纱线排列

a. 对话框输入



只有在完成纱线定制对话框后，才能对纱线排列对话框输入数据。

对话框中间的列表框显示的是模拟中用到的所有经纱和纬纱。

**经纱排列：**所有经纱的排列顺序。例如上面对话框的经纱排列表示，经纱按照这样一种规律排列：22 根 A 号纱线，接下来 26 根 B 号纱线，再是 26 根 A 号纱线，25 根 C 号纱线... 一个排列循环包括 260 根经纱（22+26+26+26+26+26+26+26+26+26+4）。当经纱排满 260 根时，循环到排列开头：接下来又是 22 根 A 号纱线。

**经纱密度：**26.20(1000) 表示前 1000 根经纱按照每厘米 26.2 根的密度排列

**纬纱排列：**所有纬纱的排列顺序。格式同经纱。

**纬纱密度：**纬纱的密度。

注：在完成色纱排列的输入后，按下 OK 键，程序会对纱线的排列进行检查，经纱的排列只能用到列表中显示的经纱，纬纱的排列只能用到列表中显示的纬纱，否则程序会要求重新输入。

b. 编辑文件导入

用纯文本方式编辑或创建一个\*.lst 文件

经纱排列的一个循环包括 260 根

经纱的排列

纬纱排列的一个循环包括 220 根

纬纱的排列

经纱只有一种密度

前 1000 根经纱的密度为：每厘米 26.2 根

纬纱只有一种密度

前 1000 根纬纱的密度为：每厘米 22.6 根

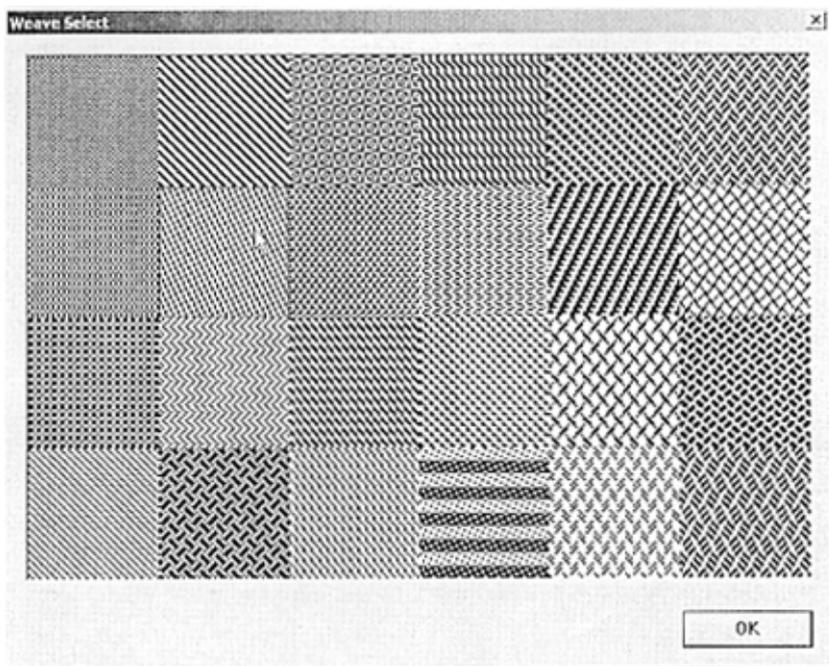


### 3.4.2.2.3 选择组织或纹版数据

#### a. 对话框选择输入

组织和纹版数据本质上都是由 0, 1 组成的矩阵。它描述了织物上的每一个经纬纱交织点的交织状况，是经纱覆盖纬纱（对应于数值 1），还是纬纱覆盖经纱（对应于数值 0）。所不同的是，组织数据规律性很强，是较小的矩阵的循环，数据量小，对应于一般的纺织织物；纹版数据规律性较弱，数据量较大，对应于提花织物。

对话框中只提供可选择的组织数据，纹版数据须由文件导入方式输入



对话框中的每一个选项对应于一种组织数据，黑色的区域表示经组织点（对应于数值 1 的区域），白色的区域表示纬组织点（对应于数值 0 的区域）。

#### b. 编辑文件导入

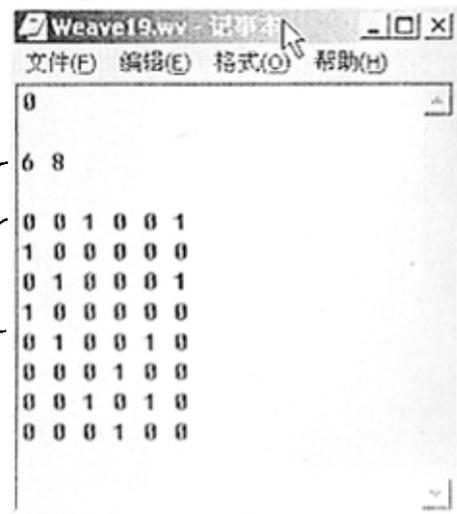
采用由文件导入的方式可以输入组织数据或纹版数据

组织数据:

表示以下信息是组织数据

基本的组织循环，包括多少列，多少行

组织矩阵



纹版数据：（纹版数据由其他应用程序生成，例如\*.wbf）

表示以下信息是纹版数据

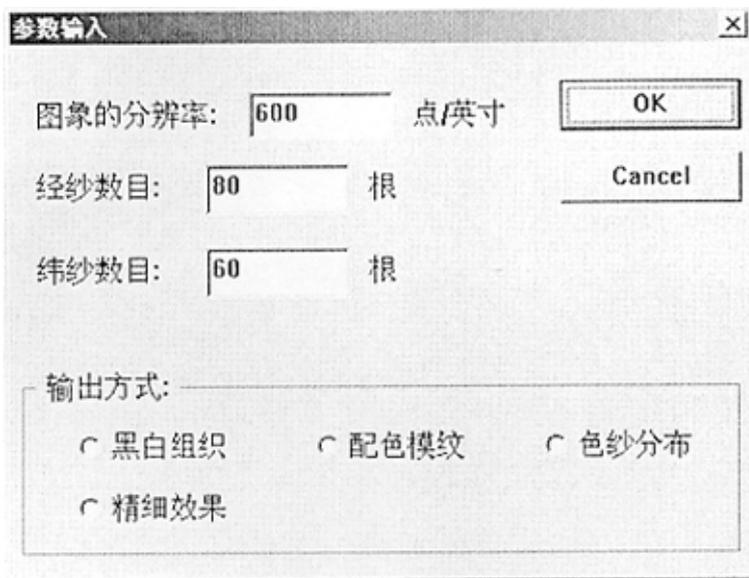
纹版文件的位置和文件名

纹版数据有多少列



### 3.4.2.3 织物输出参数定制

#### 3.4.2.3.1 输出参数设置



输出的分辨率：显示、打印、输出图像的分辨率，与打印的尺寸无关（缺省值 600），整型

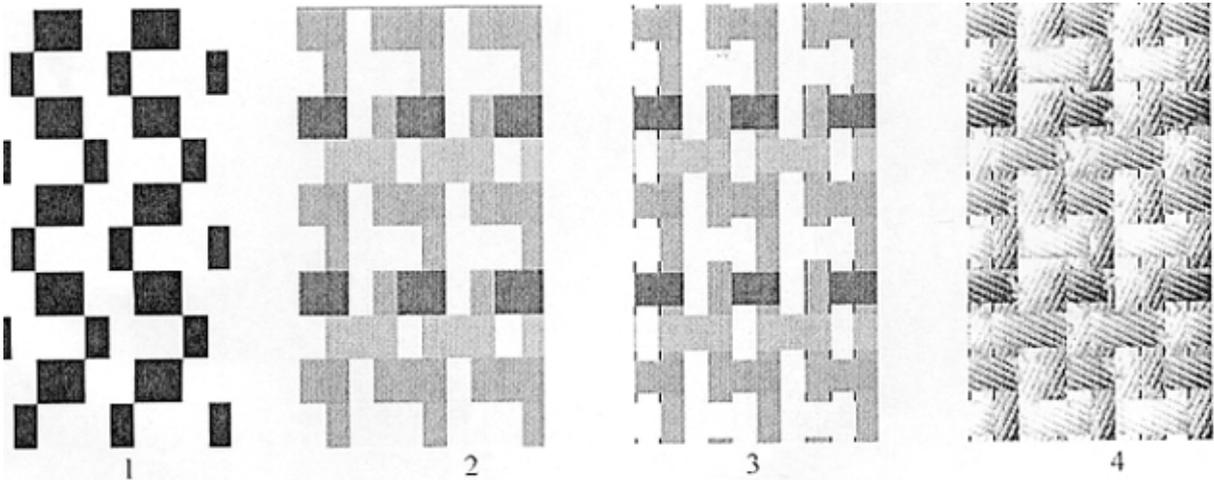
经纱数目：模拟织物中经纱的数量，和输出分辨率一起影响输出图像的大小（缺省值 80），整型

纬纱数目：模拟织物中纬纱的数量，和输出分辨率一起影响输出图像的大小（缺省值 60），整型

#### 3.4.2.3.2 输出方式设定

黑白组织：（下图 1）用黑白两色反映了织物经纬纱交织的组织结构：黑

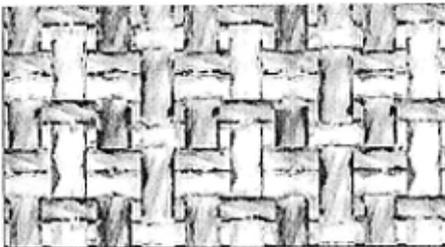
色的区域表示经纱在上，白色的区域表示纬纱在上。



配色模纹：（上图 2）用实际的纱线颜色代替黑白组织中的各个组织点  
 色纱分布：（上图 3）在图 2 的基础上描绘出纱线实际的粗细和排列密度  
 精细效果：（上图 4）加入了捻度、阴影、毛羽等各种效果的最终输出结果

### 3.4.2.3.3 选择贴图方式模拟纱线

另一种可选的输出效果是用贴图方式模拟纱线的形态，程序中使用的纱线图像是通过数码相机拍摄的实物图像。操作方法是：在定制输出参数之前按下工具栏中的 P 按钮，然后选择精细效果的输出方式，效果如下：

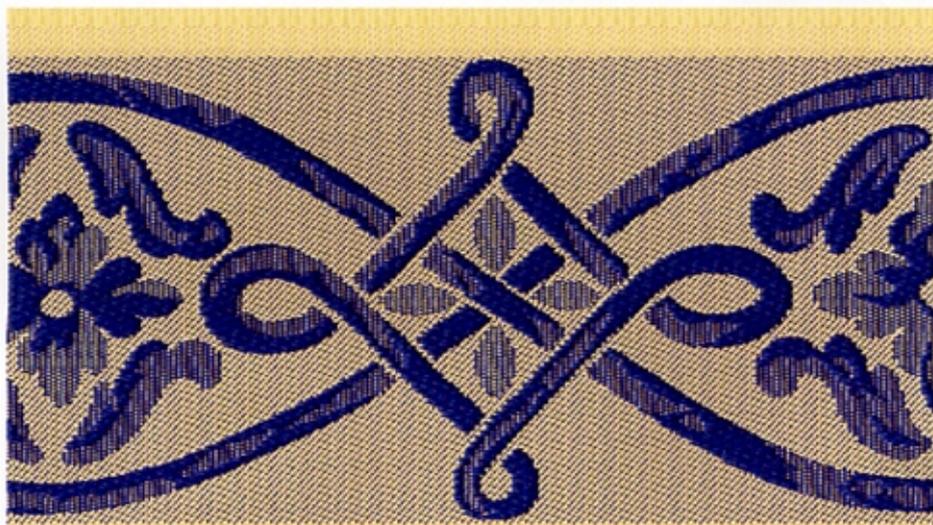
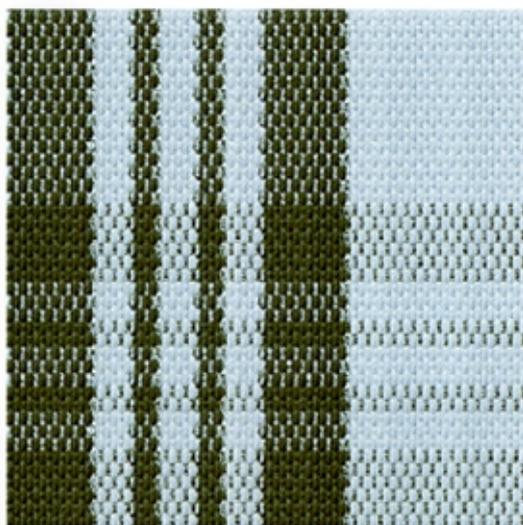
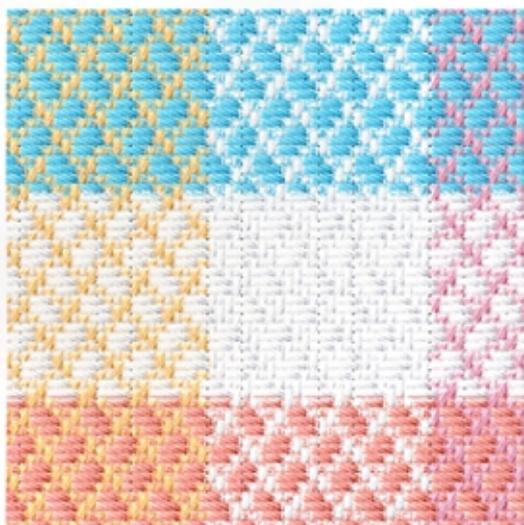
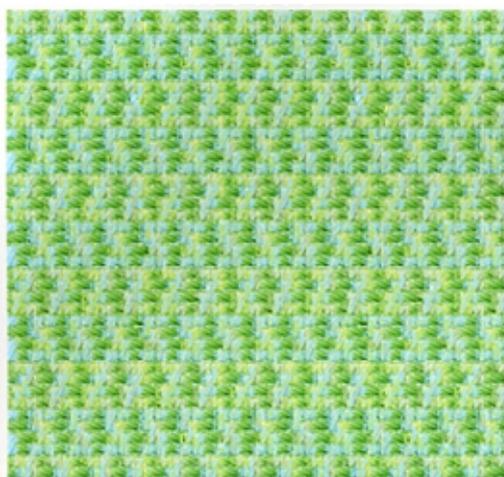


使用这种方法时，纱线的一些属性（如捻度）并不是根据实际的输入值设置的，也不能调整这些属性。增加了这一功能的目的是体现纱线边缘弯曲的效果，并且加速了模拟的速度；当使用目的是非工艺生产时，这也是一个可供的选择。

#### 3.4.2.3.4 保存输出文件

在菜单中选择保存织物输出图像或者按下工具栏中的存档按键，可以保存输出的图像文件 (\*.bmp 格式)

### 3.5 模拟效果展示



## 第4章 三维织物仿真

### 4.1 设计思路

#### 三维模拟织物

在二维环境下对织物模拟已经实现，其效果存在以下有待改进之处：

1. 只能模拟出以正视图观察的织物面料效果
2. 对纱线的模拟缺乏立体感
3. 不能体现纱线交织的立体结构
4. 纱线交织的阴影只是在图像层次依据视觉效果的模拟，而非精确计算的结果
5. 对织物的光照效果没有精确的计算

由此，，提出了三维环境下模拟织物的设计思路，以 OpenGL 作为绘制的工具，重用二维环境中用到的模块与思想方法，主要包括以下几个方面的工作：

1. 单根纱线的基本形态模拟；
2. 单根纱线的精细效果；
3. 提出一种经纬纱交织时，纱线走向的模型
4. 在纱线的光照效果中引入 BRDF 光照模型，模拟其各向异性的织物属性
5. 用光线跟踪算法计算生成纱线交织的阴影效果

在三维织物仿真设计中,我的设计思路是先用一些 Bezier 曲面拼接成单根纱线的表面模型，是纱线轴心位置的函数；然后通过给定的纱线密度、直径参数分别计算经纱和纬纱的弯曲形态,即计算 Bezier 曲面上每个控制点对应的纱线轴心位置；其它的纱线可以通过该模型的旋转、放缩得到；最后逐条绘制纱线。

### 4.2 设计步骤

目前已经实现的部分：

1. 单根纱线的基本形态模拟；先简单的把纱线看作细长的圆柱体，考虑以逐个绘制三维曲面的方式来描绘纱线的外形
2. 提出一种经纬纱交织时，纱线走向的模型：先以平纹织物为例，观察模型产生的效果及可行性

#### 1. 单根纱线的基本形态模拟

用 4 片 Bezier 曲面连接成形似圆柱体的曲面，模拟一小段纱线。具体的做法是：假设纱线与 Z 轴平行

- a. 用圆模拟纱线段的横截面，横截面是平行于 xoy 平面的
- b. 取 1/4 圆弧 (4 个控制节点)，利用控制顶点的平移，在 z 周方向作延伸可得 16 个控制节点，做成一个 Bezier 曲面，如下图

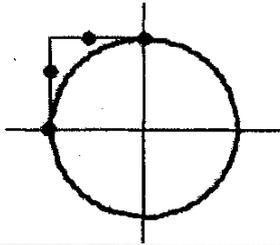
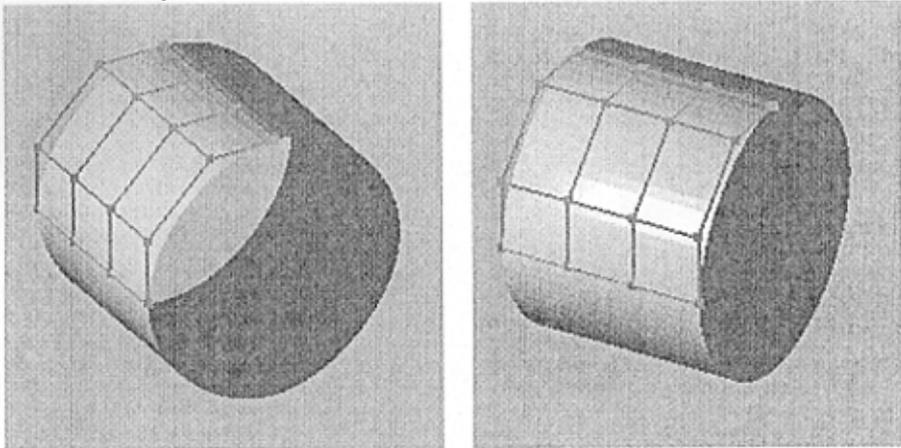
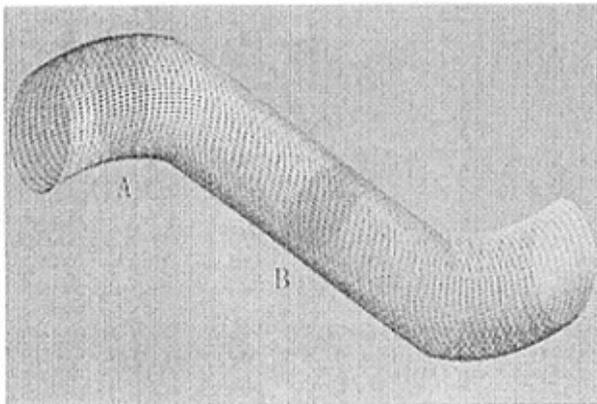


图 1



- c. 事实上，控制顶点相对于圆柱体轴心线的相对位置是不变的  
完成其它三个曲面

使用这种方式时，由于控制定点都在与纱线走向垂直的方向上。这将导致随着纱线交织时走势的变化，纱线段的横截面积的大小可能会变化。



如图，A 处的横截面积 > B 处的横截面积。

因为  $S$  (横截面积) = 图 1 的圆面积  $\times \cos \theta$

$\theta$  是该处纱线轴心切向与水平轴 (这里是 z 轴) 的夹角

幸运的是，B 点的位置也正是经纬纱交织时纱线被挤压的位置，所以从实际

上看来，B 处的横截面积的确应该比 A 处的横截面积小一些。要指出的是，利用了这个巧合是定性分析的结果，现实的织物中，B 处的面积和 A 处的面积的比例受到织物各种属性的影响，是不确定的。

## 2. 以平纹织物为例，提出纱线走向的模型

根据上面提出的纱线表面曲面模型，我们可以把计算纱线的走向简化为计算纱线轴心的走向。下面以平纹织物为例，讨论决定各纱线轴心的位置函数

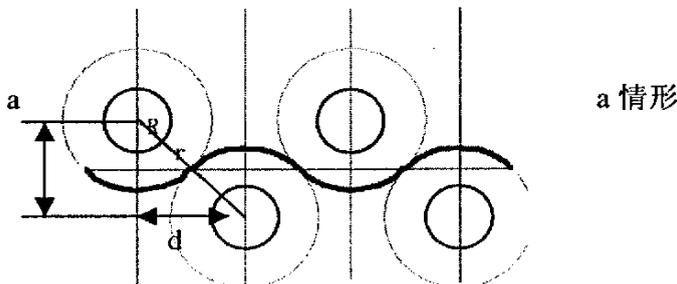
影响位置函数的因素：经纱的直径、密度；纬纱的直径，密度。为了简化起见，所有的经纱采用同一种直径和密度；所有的纬纱采用同一种直径和密度。这种简化是可行的，因为如果经、纬纱线之间的直径和密度不同，我们可以针对每一种直径和密度的纱线计算各自的纱线轴心函数；必要时，还可以采用分段函数来连接同一纱线的不同纱线走向模型。

已知一种纱线的排列（如纬纱），求另一种纱线的轴心位置（如经纱）

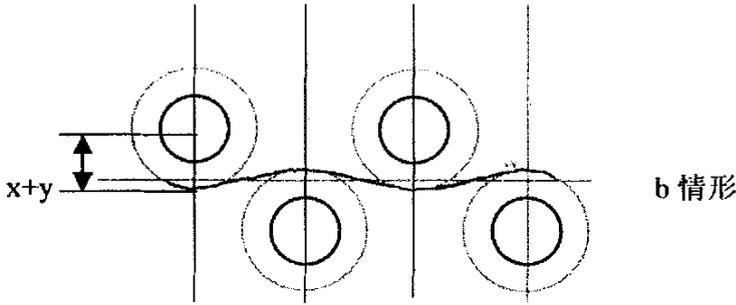
下图中，圆表示纬纱的横截面。在本文介绍的模型中，纱线轴是由分段曲线构成的，这些段要么是直线，要么是圆弧。对于经纱，当经纱的位置函数是圆弧时，圆弧中心和纬纱横截面的圆心重合，圆弧的半径是经纱的半径加上纬纱的半径。

如何确定圆和直线的分界位置呢？

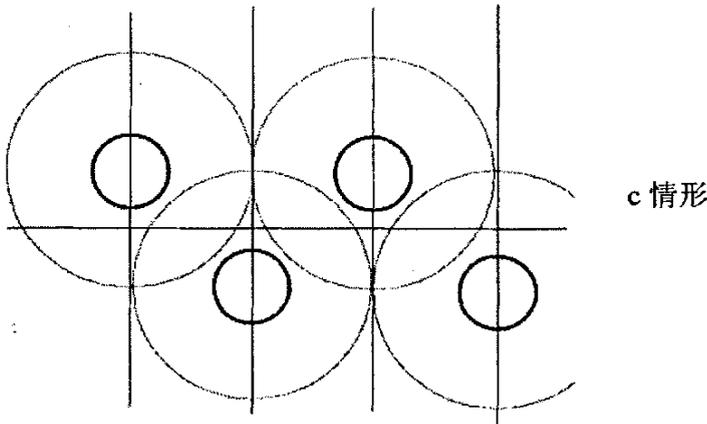
1. 以纬纱的横截圆面的圆心为圆心， $r+R$ （经纱半径+纬纱半径）为半径画圆，暂且称它为大圆
2. 然后在相邻的大圆之间作内切线，这里又分三种情况
  - a) 大圆之间本身就相切，这时经纱轴心的位置函数就是各段圆弧的连接（红色的粗线条）



- b) 当相邻的大圆相离时，圆弧和内切线连接成了经纱的轴心



c) 大圆相交，这时纬纱排列过于紧密，这种排列无法织出平纹织物。



我们已经有了已知纬纱排列求经纱排列的模型，那么，纬纱的排列是如何确定的呢，稍微思考以下就会想到，纬纱的排列似乎是由经纱的排列决定的。这样，我们好像进入了类似于先有鸡还是先有蛋的死循环。

在解决这个问题时，我是这样考虑的：

我们可以换个角度来看，在一块纺织成的面料上，理论上纱线的排列总是使得总体能量的最小。而纱线的能量主要是由纱线弯曲产生的势能贡献的。由此，我们可以得出这样的结论：纱线的排列遵循总的弯曲程度最小的原则（也就是离中心轴最近的原则），当然前提是纱线的排列符合设计的要求。

下面，我们来计算纱线究竟应该偏离中心轴多少才是弯曲程度最小。

那么，应该先计算经纱还是先计算纬纱呢？（我们似乎又回到上面的死循环中了）我的想法是：先排列密度大、直径大的纱线。因为如果先排列疏松的纱线的话，在计算完成其轴心位置后，可能存在在疏松纱线的弯曲程度下，紧密的纱线排不下的问题。

然后，当较紧密的纱线（假设为纬纱）已经排列完毕，这时也存在两种情形，即上文中提到的 a 情形和 b 情形。

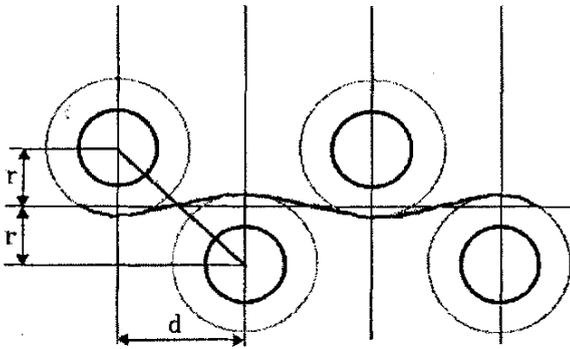
对 a 情形来讲, 大圆的圆心距 =  $2 \times (R+r)$ , 由纬纱的密度又可以计算得纬纱得的水平间距  $d$ , 所以相邻纬纱的垂直距离  $a$  是确定的。由此, 经纬纱的弯曲程度也可以计算出来了。

对于 b 情形, 假设纬纱横截面的圆心距水平轴的距离为  $x$ , 则纬纱的势能正比于  $x^2R^2$ , 再假设经纱轴心距水平轴的距离为  $y$ , 则经纱的势能正比于  $y^2r^2$ , 且有  $x+y = R+r$ 。从而, 这块织物的势能正比于  $nx^2R^2+my^2r^2$ 。其中  $n$  为纬纱的总数目,  $m$  为经纱的总数目。有因为是平纹织物, 所以  $m = n$ 。织物的势能正比于  $x^2R^2+y^2r^2$ 。显然, 当  $xR = yr$  时, 势能最小。又  $x+y = R+r$ , 所以  $x = r, y = R$  时势能最小。经纱位移  $R$ , 纬纱位移  $r$ 。

计算 a、b 模型的分界

既然我们采用了 a、b 两种不同的模型来计算纱线轴心的位置, 那么, 对于一定属性的纱线, 我们如何分辨究竟应该属于哪种情形呢?

b 情形: 大圆的圆心距  $> 2 \times (R+r)$

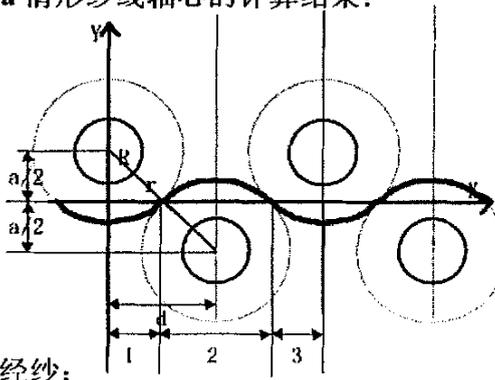


圆心距<sup>2</sup> =  $d^2+(2r)^2 > 4 \times (R+r)^2$ ;

a 情形: 大圆的圆心距 =  $2 \times (R+r)$

也就是除了 b 情形之外的情形  $\rightarrow d \leq (4R^2+8Rr)^{1/2}$

a 情形纱线轴心的计算结果:



经纱:

从图可得, 经纱的轴心函数是周期为  $2d$  的分段函数 ( $d$  是纬纱的水平间距)。分

为3段:

其中:  $a = \sqrt{(2R+2r)^2 - d^2}$

分段 1:  $0 < x \leq \sqrt{(R+r)^2 - (\frac{a}{2})^2}$

$$y = \frac{a}{2} - \sqrt{(R+r)^2 - x^2}$$

分段 2:  $\sqrt{(R+r)^2 - (\frac{a}{2})^2} < x \leq d + \sqrt{(R+r)^2 - (\frac{a}{2})^2}$

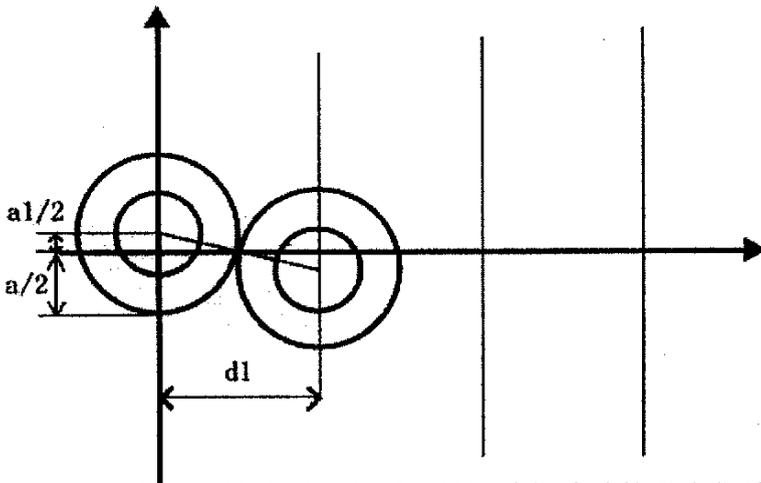
$$y = \sqrt{(R+r)^2 - (x-d)^2} - \frac{a}{2}$$

分段 3:  $d + \sqrt{(R+r)^2 - (\frac{a}{2})^2} < x \leq 2d - \sqrt{(R+r)^2 - (\frac{a}{2})^2}$

$$y = \frac{a}{2} - \sqrt{(R+r)^2 - (x-2d)^2}$$

纬纱:

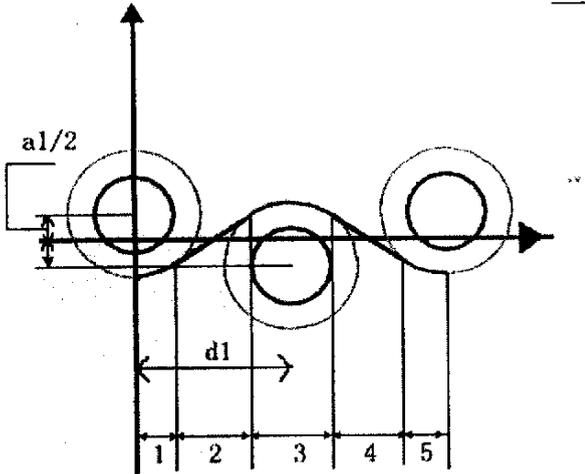
让我们再次考虑: 经纱的密度要符合怎样的条件, 才能使纬纱排得下



图中, 小圆表示经纱截面, 上面已经计算得纬纱的最大位移是  $a/2$ ; 则经纱的最大位移是:  $a/2 = R+r-a/2$

大圆相切时是经纱能构排列得最紧密得时候;

所以, 经纱的间距  $d1^2 \geq (2R+2r)-a1^2$



从图可得，纬纱的轴心函数是周期为  $2d1$  的分段函数（ $d1$  是经纱的水平间距）。分为 5 段：

其中： $a1 = 2(R+r) - a$

$$x_0 = \frac{2(R+r)^2 d1 - \sqrt{a1^2 (R+r)^2 (d1^2 - 4(R+r)^2 + a1^2)}}{a1^2 + d1^2}$$

$$k = \frac{2a1d - \sqrt{a1^2 (R+r)^2 (d1^2 - 4(R+r)^2 + a1^2)}}{2(4(R+r)^2 - d1^2)}$$

分段 1:  $0 < x \leq x_0$

$$y = \frac{a1}{2} - \sqrt{(R+r)^2 - x^2}$$

分段 2:  $x_0 < x \leq d1 - x_0$

$$y = k(x - \frac{d1}{2})$$

分段 3:  $d1 - x_0 < x \leq d1 + x_0$

$$y = \sqrt{(R+r)^2 - (x - d1)^2} - \frac{a1}{2}$$

分段 4:  $d1 + x_0 < x \leq 2d1 - x_0$

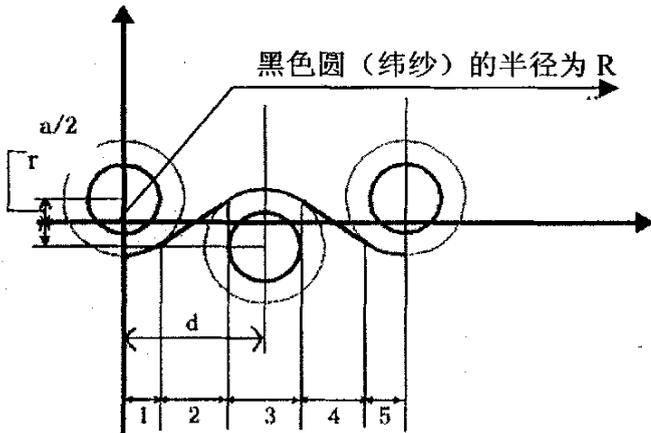
$$y = -k(x - \frac{3d1}{2})$$

分段 5:  $2d1 - x_0 < x \leq 2d1$

$$y = \frac{a1}{2} - \sqrt{(R+r)^2 - (x - 2d1)^2}$$

经纱的间距满足  $d1^2 \geq (2R+2r) - a1^2$  时，就可保证上面公式中根号下的内容为正。

b 情形纱线轴心的计算结果:



经纱:

从图可得, 经纱的轴心函数也是周期为  $2d$  的分段函数 ( $d$  是纬纱的水平间距)。分为 5 段

$$x_0 = \frac{(R+r)^2 d - \sqrt{r^2 (R+r)^2 (d^2 - (4R^2 + 8Rr))}}{\frac{d^2}{2} + 2r^2}$$

$$k = \frac{dr - \sqrt{(R+r)^2 (d^2 - (4R^2 + 8Rr))}}{2((R+r)^2 - \frac{d^2}{4})}$$

分段 1:  $0 < x \leq x_0$

$$y = r - \sqrt{(R+r)^2 - x^2}$$

分段 2:  $x_0 < x \leq d - x_0$

$$y = k(x - \frac{d}{2})$$

分段 3:  $d - x_0 < x \leq d + x_0$

$$y = \sqrt{(R+r)^2 - (x-d)^2} - r$$

分段 4:  $d + x_0 < x \leq 2d - x_0$

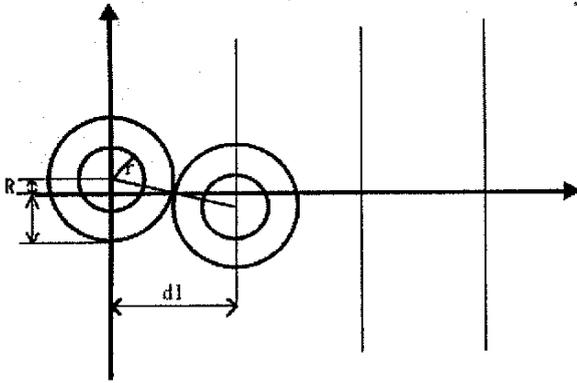
$$y = -k(x - \frac{3d}{2})$$

分段 5:  $2d - x_0 < x \leq 2d$

$$y = r - \sqrt{(R+r)^2 - (x-2d)^2}$$

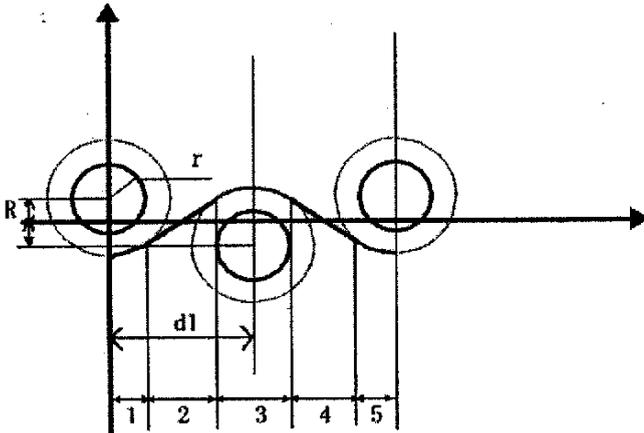
纬纱:

经纱的密度要符合:



图中，小圆表示经纱截面，上面已经计算得纬纱的最大位移是  $r$ ；则经纱的最大位移是  $R$

大圆相切时是经纱能构排列得最紧密得时候；所以，经纱的间距  $d1^2 \geq (2R+2r)^2 - (2R)^2 = 8rR+4r^2$



从图可得，纬纱的轴心函数是周期为  $2d1$  的分段函数（ $d1$  是经纱的水平间距）。

分为 5 段：

其中：

$$x_0 = \frac{(R+r)^2 d1 - \sqrt{R^2 (R+r)^2 (d1^2 - 4(2rR+r^2))}}{2(R^2 + \frac{d1^2}{4})}$$

$$k = \frac{kd1 - \sqrt{(R+r)^2 (d1^2 - 4(2rR+r^2))}}{2((R+r)^2 - \frac{d1^2}{4})}$$

分段 1:  $0 < x \leq x_0$   

$$y = R - \sqrt{(R+r)^2 - x^2}$$

分段 2:  $x_0 < x \leq d_1 - x_0$   

$$y = k(x - \frac{d_1}{2})$$

分段 3:  $d_1 - x_0 < x \leq d_1 + x_0$   

$$y = \sqrt{(R+r)^2 - (x - d_1)^2} - R$$

分段 4:  $d_1 + x_0 < x \leq 2d_1 - x_0$   

$$y = -k(x - \frac{3d_1}{2})$$

分段 5:  $2d_1 - x_0 < x \leq 2d$   

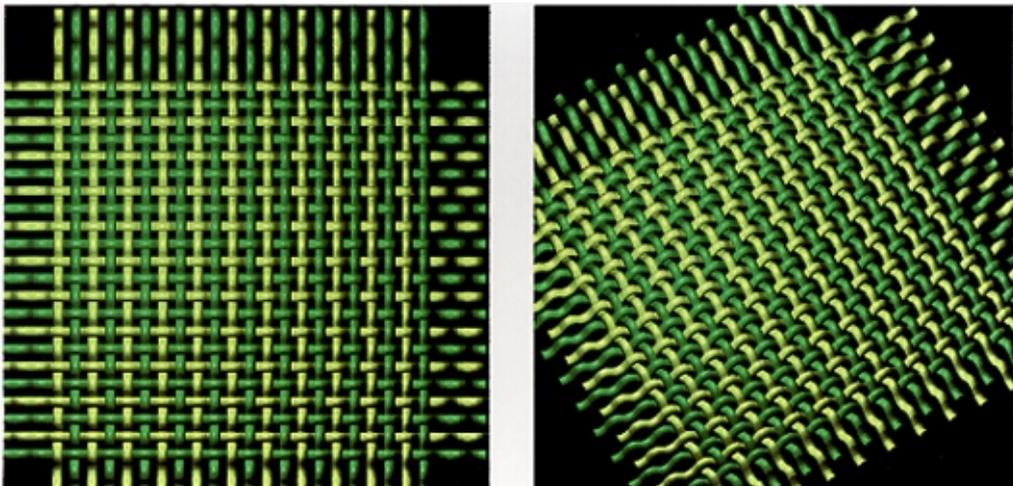
$$y = R - \sqrt{(R+r)^2 - (x - 2d_1)^2}$$

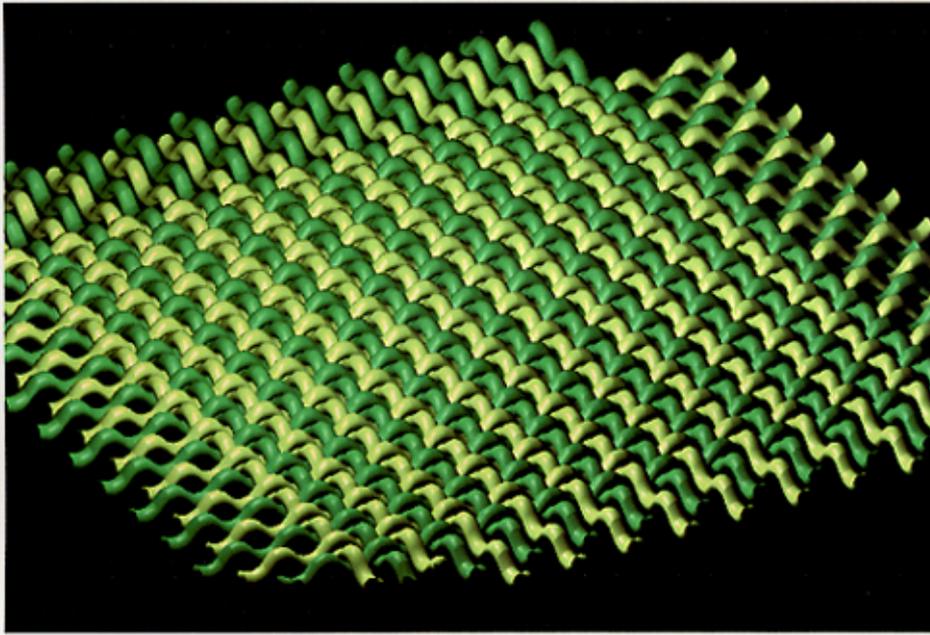
这种模型的主要特性是能够在微观层次上对纱线的的交织形态有比较精确的表示。通过模拟视图的旋转、平移、放缩，纺织工艺人员能够全面的掌握关于织物组织结构的信息，提高织物设计的精度，同时为将来细致的模拟三维织物结构打下了基础。

存在的问题：在计算纱线交织形态时，没有考虑纱线被挤压的情形，所以不能模拟经纱和纬纱排列都很紧密的织物。

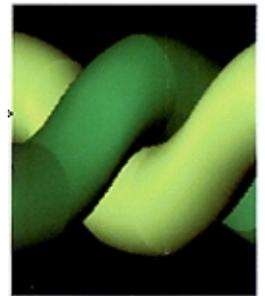
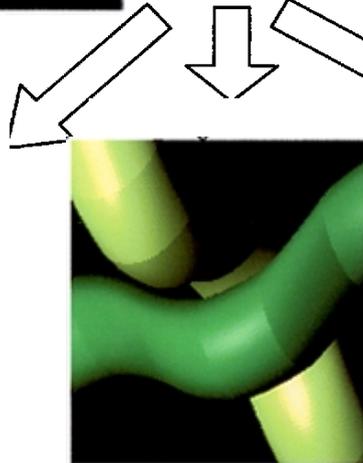
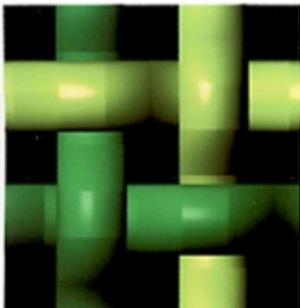
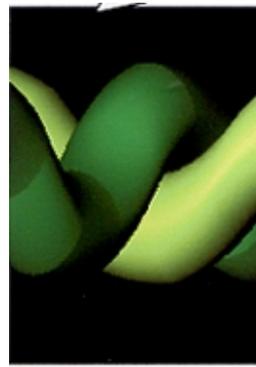
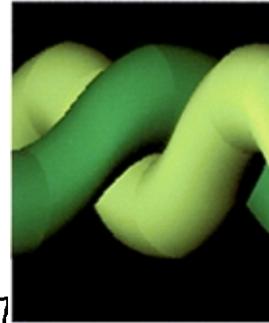
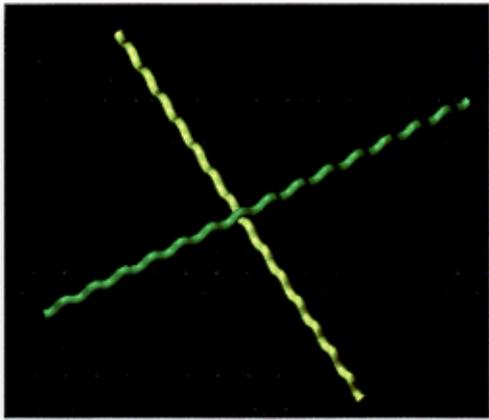
可能的解决方法：对交织模型作修改，使得纱线的半径能够随着纱线的密度自动调整

### 4.3 效果展示





利用缩放旋转等变化，检查织物的组织结构



## 第5章 总结

织物仿真是一个比较宽的课题，在研究过程中，它实际牵涉到很多项技术，而且需要理论与实践的结合；同时，又是一个比较深的课题，需要将各项技术进行筛选，调整，结合，并应用到该课题中。

能够参加这一课题的研究，对我来说是宝贵的经历。整个课题的研究过程大致分为以下几个阶段：

需求分析：清楚地理解将纺织厂的现实需求，了解需求的背景，同时将需求合理的转化为数据模型，明确其输入和输出；

整体构思：将一个问题分解成几个模块，明确各个模块的功能，输入及输出；

研究方法探索：收集材料，筛选出各个模块的解决方案；

编程实现与测试；

与需求方沟通，调整程序并反馈。

在本课题研究过程中，我获得了许多的经验，特别是在解决问题的思路方面。我认为本研究课题将来的主要工作侧重在两个方面：

- 简化操作方法，同时增加程序应用的灵活性；
- 改善或引入新算法，提供更好的模拟效果。

具体地说，在平面织物仿真方面

1. 加入纱线数据库，色卡数据库等插件，完善界面设计，使得软件的应用起来更加简单、友好
2. 优化织物的模拟效果，对于纱线的模拟在于增加纱线的生动感，例如纱线在交织过程中或挤压或蓬松的效果，纱线的弯曲，偏移纱线轴的现象模拟；在宏观方面，可以增加织物的褶皱，变形效果

3. 拓宽模拟纱线的种类，添加各种异形纱(雪尼尔纱，大肚纱等等)的模拟

在三维织物仿真方面

1. 用 BRDF 方法模拟织物各向异性的光照效果
2. 用 Perlin 函数或者其他方法生成三维织物的毛羽
3. 将纱线形态的模拟拓宽到其他组织结构（目前只有平纹组织）
4. 增加三维织物的褶皱，变形效果，还可以做成动画的效果
5. 调整纱线模型和交织模型，使之能模拟紧密排列的纱线

## 第6章 名词解释

**经纱和纬纱:**在纺织织物中,所有的纱线被安排成沿着互相垂直的两个方向排列:

纵向的称为经纱,横向的称为纬纱

**色纱排列:**经纱和纬纱都是由各种纱线排列而成的,色纱排列描述了排列位置与纱线的对应关系,分为经纱的色纱排列和纬纱的色纱排列

**纱线密度:**单位是(根/cm),同一种纱线排列可以有多种密度,例如表达式 25.5(100) 42.8(300) 表示前 100 根纱线的密度是 25.5 根/cm,接下来 300 的根密度为 42.8 根/cm

**组织点:**经纱和纬纱的交点,在该交点上若经纱覆盖纬纱,则称为经组织点,否则纬纱覆盖经纱,成为纬组织点

**组织和纹版:**组织和纹版数据本质上都是由 0, 1 组成的矩阵。它描述了织物上的每一个经纬纱交织点的交织状况,是经纱覆盖纬纱(对应于数值 1),还是纬纱覆盖经纱(对应于数值 0)。所不同的是,组织数据规律性很强,矩阵的行列数较小,对应于一般的纺织织物;纹版数据规律性较弱,数据量较大,对应于提花织物。

**浮长:**一段纱线中连续露在织物表面的组织点的数量

**回头数:**纱线在排列时往往具有一定的规律性,是某些重复单元的组合,把最小重复单元中的纱线根数称为回头数,即组织矩阵的行数和列数。

## 第7章 参考文献

1. [http://freespace.virgin.net/hugo.elias/models/m\\_perlin.htm](http://freespace.virgin.net/hugo.elias/models/m_perlin.htm)
2. S. H. Westin, J. R. Arvo, K. E. Torrance. Predicting Reflectance Functions from Complex Surfaces [J]. *SIGGRAPH'92*: 255~264. July 1992
3. An Introduction to BRDF-Based Lighting, NVIDIA Corporation, Chris Wynn
4. Creation of Virtual Fabrics by Wavelet Analysis of Spun Yarn Density Signals, Moon W. Suh and Jooyong Kim
5. I00-A06 FABRIC DESIGN AND ANALYSIS SYSTEM IN 3D VIRTUAL REALITY, Annual Report September 30, 2000, PIs: Dr. Sabit Adanur – Auburn University, Dr. Yehia El Mogahzy – Auburn University, Dr. Faissal Hady – Auburn University
6. An Approach to Cloth Synthesis and Visualization, Vladimir L., Edward A.
7. Predicting Reflectance Functions from Complex Surfaces
8. Y. Q. Xu, Y. Chen, S. Lin, etc. Photorealistic Rendering of Knitwear Using The Lumislice [J]. *SIGGRAPH'2001*: 391~398. Aug 2001.
9. 顾尔丹, 马凌洲等。提花织物外观的计算机模拟生成 [J]. *系统仿真学报*, 2001, 13(3):400-403.
10. K. Perlin. An Image Synthesizer [J]. *SIGGRAPH'85*: 287~296
11. Characterization and Control of Fabric Properties in Textile and Apparel Manufacturing. PIs: P. Banks-Lee, N. C. State; etc.
12. Realistic Rendering and Animation of Knitwear. Yanyun Chen, Stephen Lin, Hua Zhong, Ying-Qing Xu, Baining Guo, and Heung-Yeung Shum
13. Three Dimensional Model-Based Tracking Using Texture Learning and Matching. Philippe Gerard and Andrzej Gagalowicz
14. Perlin Noise and Turbulence Written by Paul Bourke January 2000