

二维线框图重建三维形体

李琦 王宇翔 彭艺 林建立

一、 背景介绍

基于工程视图的三维形体重建技术是计算机学科一个非常重要的研究领域。从六十年代初发展到现在，工程视图的三维重建研究已经到达了一个较为成熟的阶段，从最初多面体重建，到曲面体重建；从简单形体的重建，到复杂装配体的重建，国内外的专家与研究学者在这一领域中已经获得了丰硕的研究成果。但是基于单个线框图的三维形体重建的研究目前还处在一个刚刚起步的阶段。

线框图(Line Drawing)区别于草图(Sketch)和一般的工程图(Orthographic View)，是工程领域中一种重要的形体表现形式。线框图由具体的制图工具绘制而成，用二维的几何元素如直线段和圆弧等，来描述一个三维形体。它不同于草图：草图往往是由设计者直接手绘而成，可能具有重复的线段、物体的投影甚至错误的几何信息，草图更注重设计，是一个粗糙的形体表现形式；但线框图是精确的，它要求具有精确地点、线、弧等二维元素来精确表示物体的构成，没有任何描述模糊或者错误的地方(除非某些具有多解情况的线框图)。线框图同样区别于工程图：工程图对形体的描述也要求精确，但三视图或者二视图表示下的工程图都要求具有正交投影的性质，而线框图一般则需要偏离一定的角度对形体进行投影，以便尽可能地得到形体的完整描述。

根据目前已有的初步研究成果，单个线框图的三维形体重建算法的具体流程主要可以归纳成两个方面，如图 1 所示：

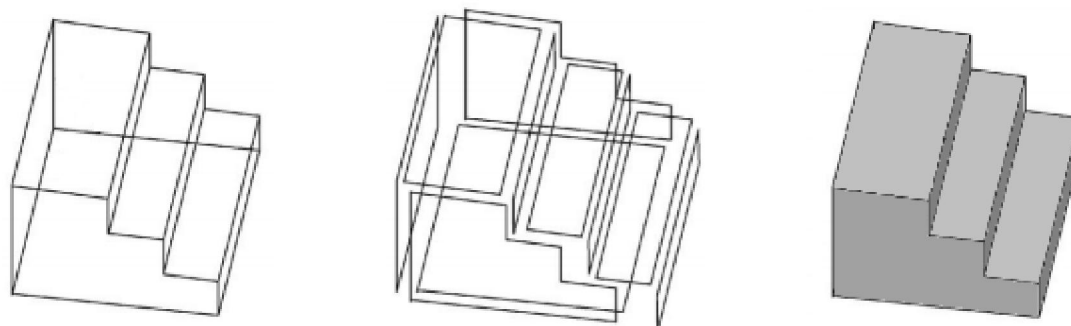


图 1: (a)单个的二维线框图；(b)恢复而得的二维平面；(c)重建而得的三维形体

◇ 拓扑信息恢复，即形体二维面的查找。

对于一个只有二维信息的单个线框图，如何根据查找得到形体各个面的构成(二维点和二维边序列)是线框图重建中的一个重要问题。根据一般二维形体的基本特征，我们可以对二维

面的构造加以约束，从而推测得到形体的拓扑信息。

◇ 几何信息恢复，即形体三维坐标重建。

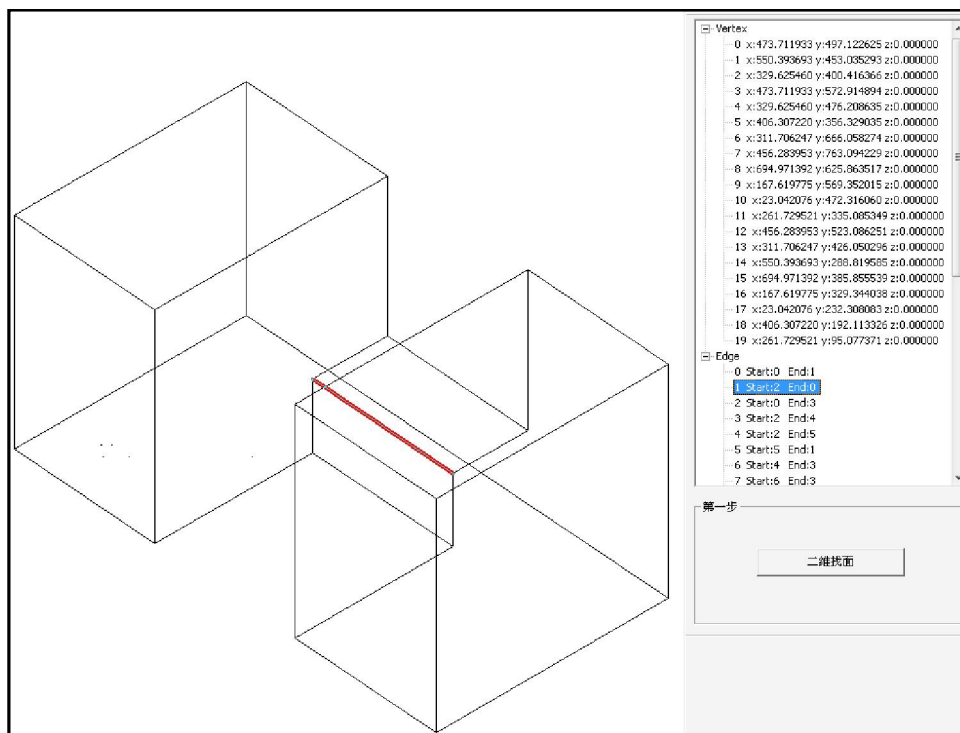
得到形体的拓扑信息之后，在没有其他视图的情况下，我们没有任何方法能够精确推导出物体的深度信息。但是基于一般人类的读图思维，我们可以利用物体本身的对称性、相似性等约束，在已知二维坐标的情况下来推测形体的三维坐标的深度值，从而恢复形体的三维模型。

二、 实验结果及程序演示

我们小组的实验平台为 Visual Studio 2005，编程语言采用 C++。测试数据汇集了以往的单个二维线框图重建论文中涉及各个测试样例，主要来自于论文作者 Peter A.C. Verley 的个人网站 <http://pacvarley.110mb.com/Sketch/Wireframes/index.html> 中的测试数据集，共有 66 个。最后的实验结果及程序演示如下：

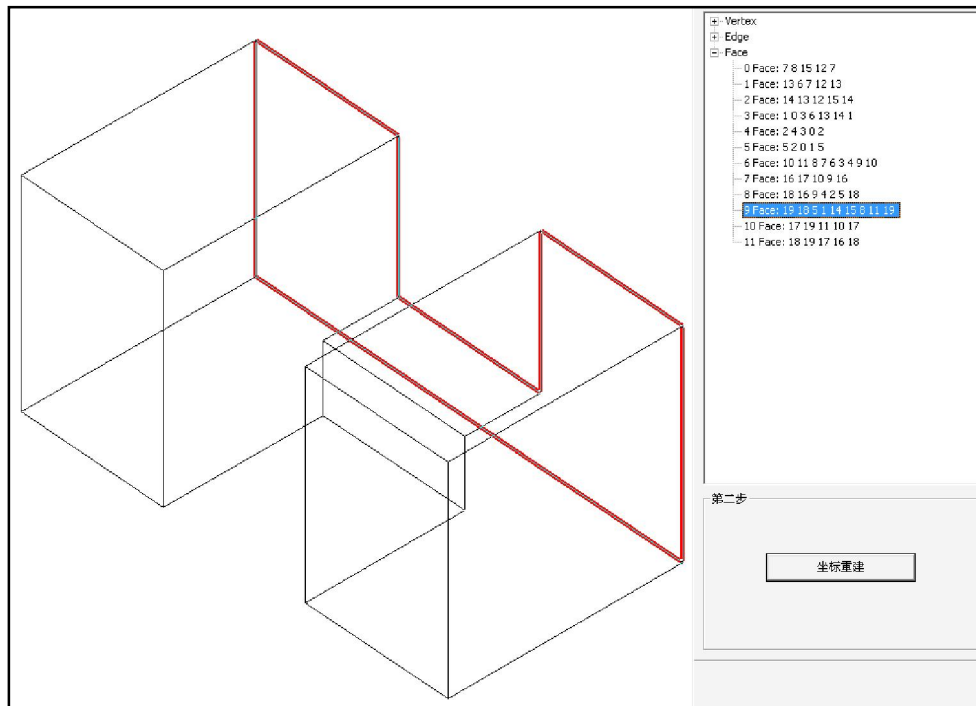
1. 基本形体读取和显示

点击“打开”按钮从测试数据集中选择 ldr 二维模型格式文件，即可显示具体的二维线框图以及相关的点和边的信息；按住 ctrl 移动鼠标可以移动模型在视图中的位置；双击界面右方控制树中具体的项可以查看对应的二维集合结构。具体下图所示：



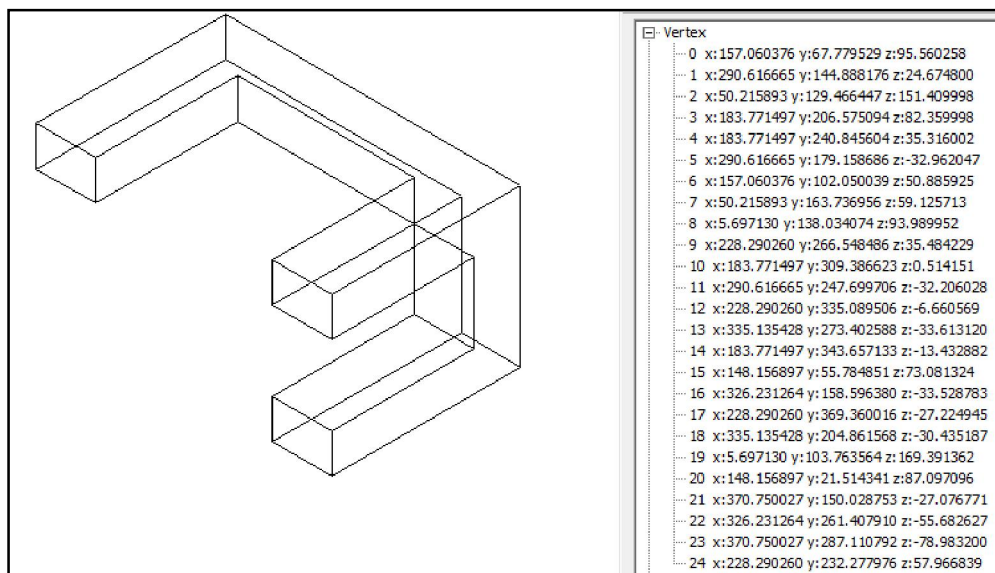
2. 二维平面查找

点击界面右方的“第一步：二维找面”按钮即可搜索得到所有的二维平面；双击右方控制树中的面对应的项，可以查看对应的二维平面。如下图中红色所示即为一个完整的二维平面，控制树中所存的面信息为点的索引集合。



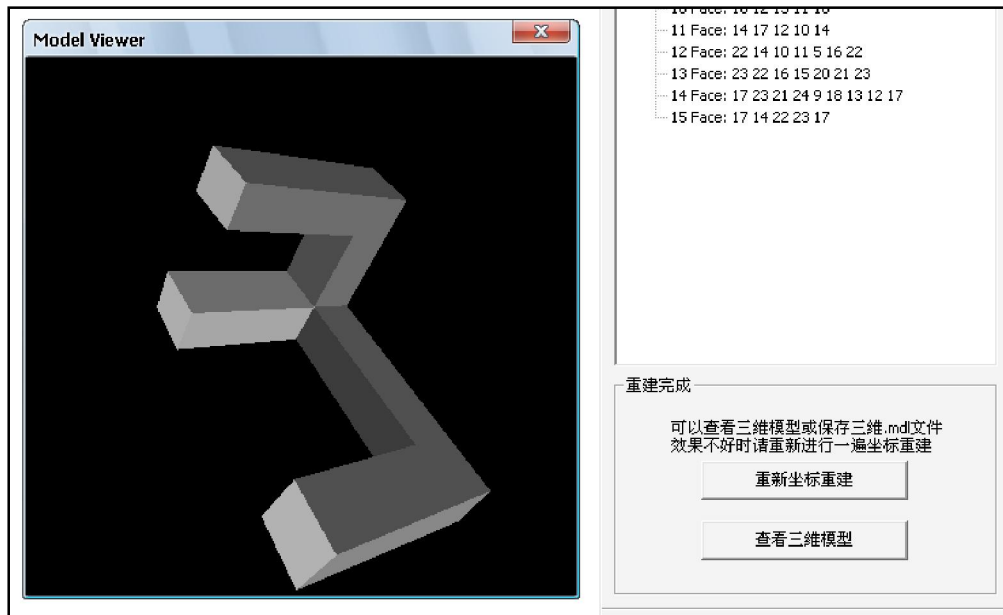
3. 三维坐标重建

点击界面右方的“第二步：坐标重建”按钮即可得到整个形体的三维坐标值；查看右方控制树中的顶点对应的项，可以看到每个顶点的三维坐标，如下图所示。重构完成后，还可以通过下面的三维形体显示来直观观察重构后的物体。



4. 三维形体显示

在上步的坐标重建之后会自动弹出显示窗口，还可以进行基本操作如下：按鼠标左键拖动可以进行旋转；ctrl+鼠标左键可以进行平移；shift+鼠标左键进行缩放；F2 切换线框图显示模式和实体图显示模式。

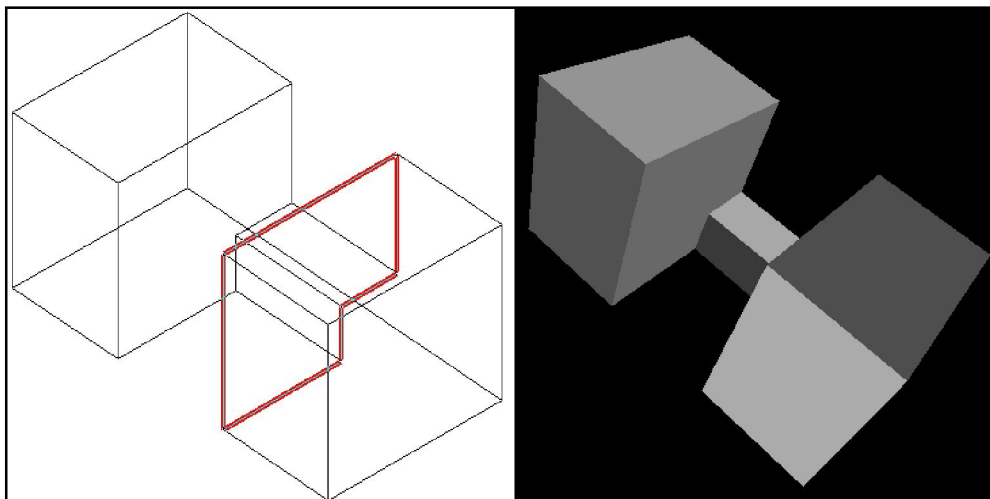


由于坐标重建会陷入局部最优解，可以对形体进行多次的坐标重建，来达到最好的三维模型效果，最后可以将文件保存为三维形体的 mdl 格式文件，可以在我们的程序中直接打开观察。

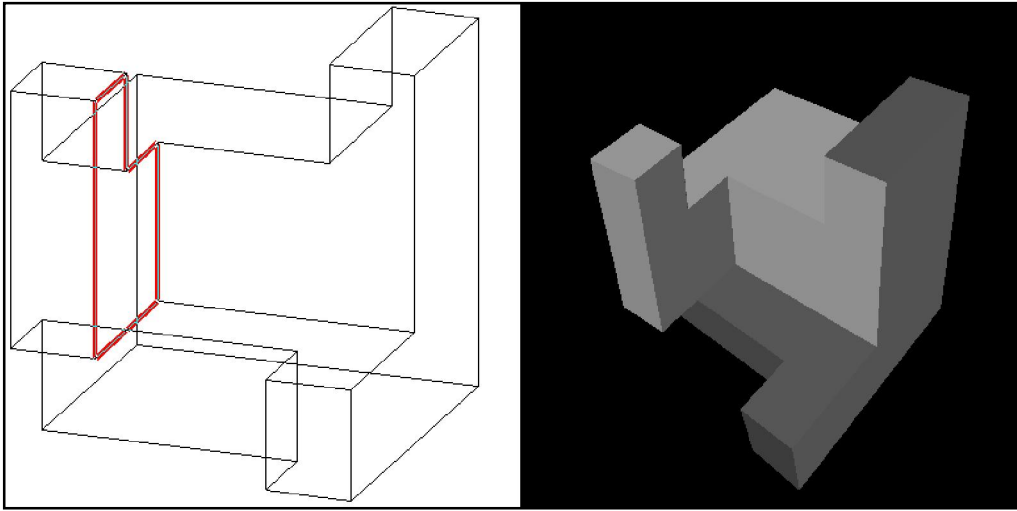
三、 实验结果示例

程序运行成功的实验示例如下：

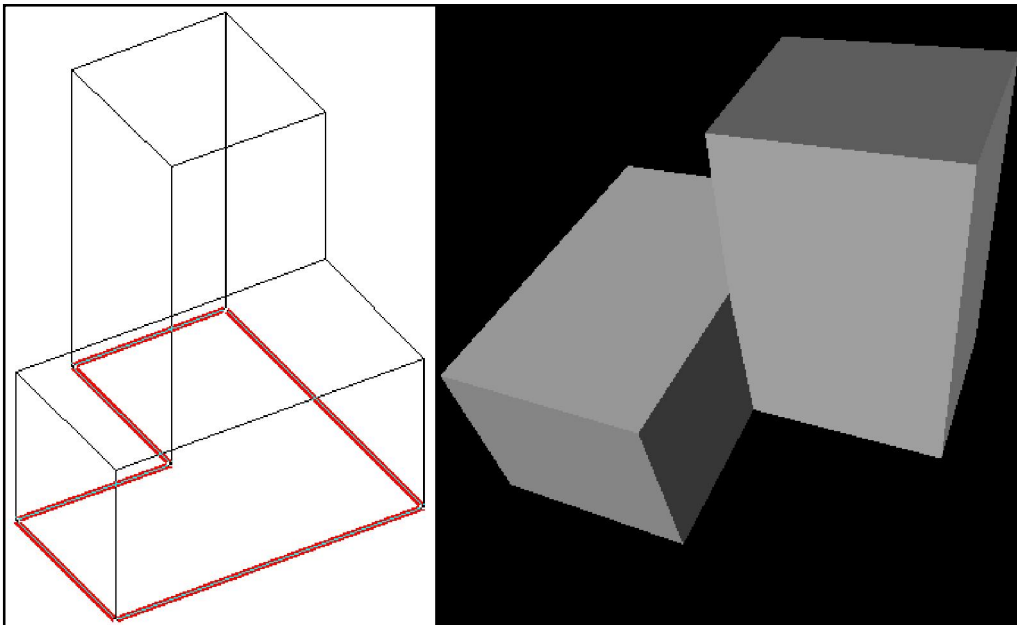
1. Cubes: 左图为二维平面查找之后的结果，其中红色面为查找而得的二维平面；右图为三维坐标重建之后的显示结果



2. **Block4**: 左图为二维平面查找之后的结果，其中红色面为查找而得的二维平面；右图为三维坐标重建之后的显示结果

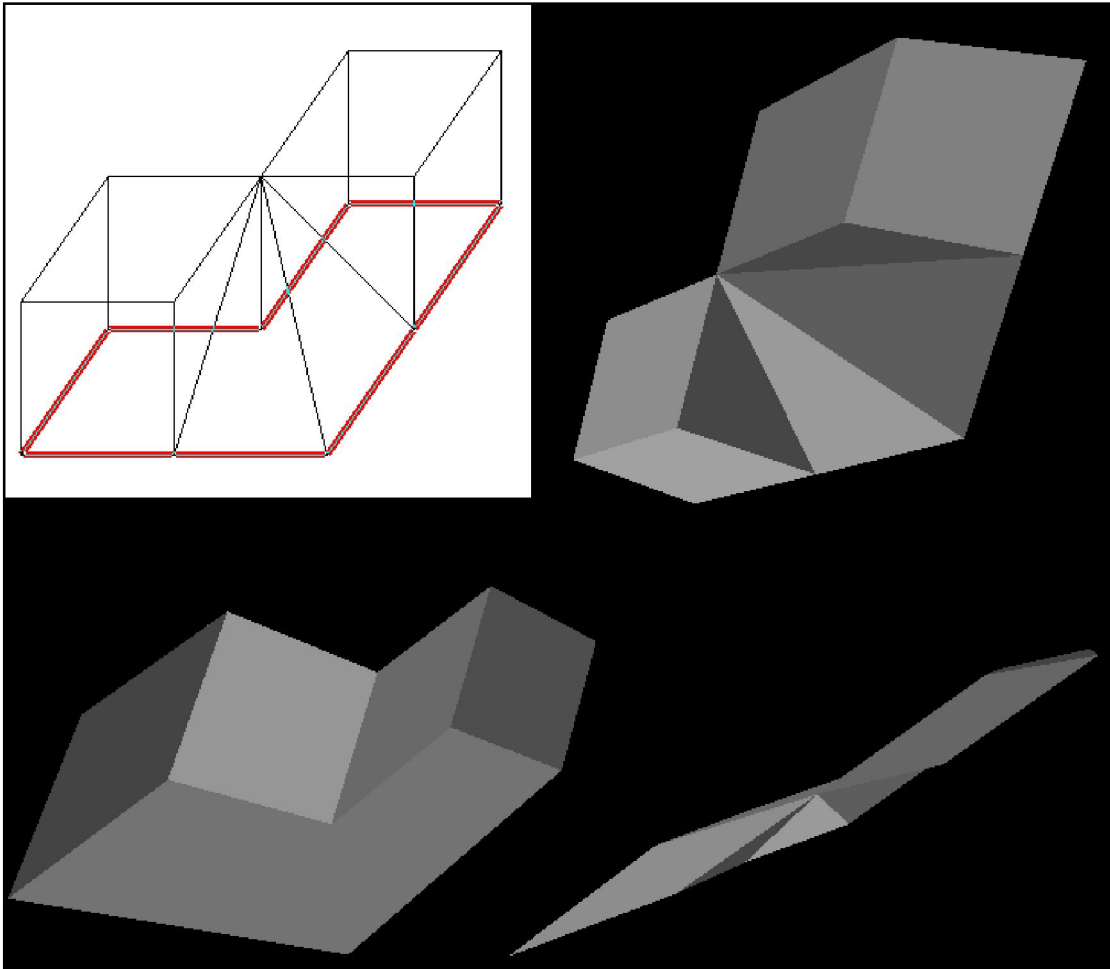


3. **Tdac**: 左图为二维平面查找之后的结果，其中红色面为查找而得的二维平面；右图为三维坐标重建之后的显示结果

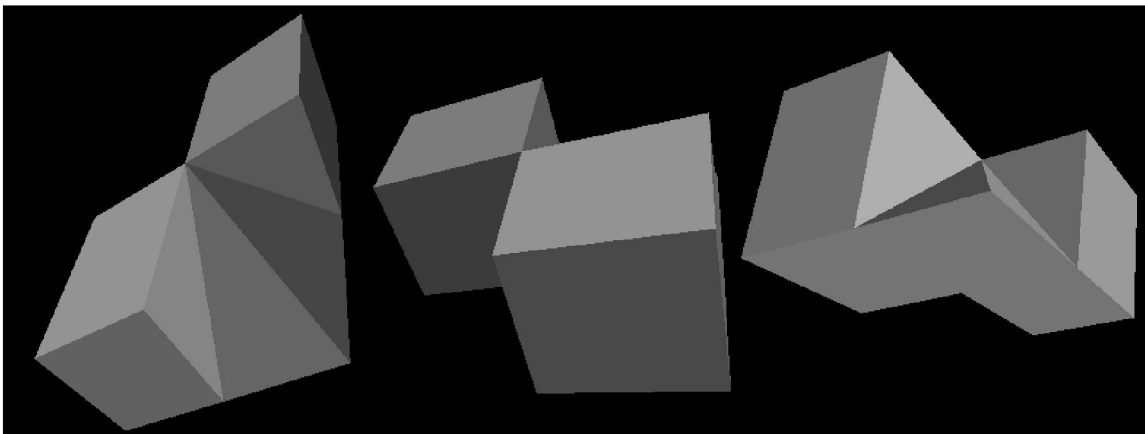


对于我们数据集中的例子，二维平面查找能够搜索得到所有正确的二维平面；而三维坐标重建算法则有时会陷入局部最优解，我们可以通过重复运行三位坐标重建算法，得到较好的三维结果进行保存。下面是一个失败的例子，之后进过重复运行，得到了新的正确三维模型：

1、Extended K-Vertex1: 上图为三维重建失败之后的形体示例，右下方的三个图分别从不同角度显示了当前模型的结果



Extended K-Vertex1: 下图为重新运行三维重建算法之后得到的新的结果，三个图分别从不同角度显示了当前模型的结果

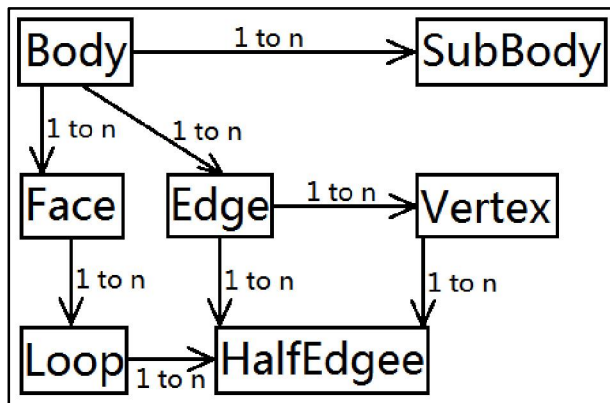


4. 实验内容及算法实现

我们小组主要根据目前在基于单个线框图重建的领域中一些最新的研究成果，实现一个基本的二维线框图重建三维形体的系统。系统主要实现的基本模块如下：

1. 基本数据结构

我们的系统参考了我们实验室目前开发的用于工程视图重建的基本数据结构并进行了适合我们线框图重建算法的修改。主要类图如下：



对于一个基本形体(body)而言，由面(face)、边(edge)、点(vertex)三类基本几何元素信息组成，我们在初始读入二维线框图的实验数据的时候，能够生成一个基本的二维形体，含有二维顶点信息和边的连接关系，然而二维面的信息是需要我们算法进行查找实现的，点的z坐标也需要我们计算得到。在算法具体的实现的过程中，我们还引入了几个基本的数据结构包括：半边(halfedge)，是指具有方向信息的边，每条边都可以生成具有两个方向的半边；面环(loop)，是指有半边组成的面的边界信息，对于一些存在空洞的面，通常由多个分离的面环组成；子实体(subbody)，是指原本的形体进行切割之后形成的多个孩子，拥有和基本形体一样的数据结构，因为在我们的切割算法中子实体还可以继续进行形体的切割。

2. 二维平面查找

我们的系统基本实现了2010年CAD期刊上Peter A.C. Verley的最新文章“A new algorithm for finding faces in wireframes” [1] 中介绍的算法，相较于以往的各种算法在二维找面的准确度和时间复杂度上都有明显地提高。算法主要思想流程如下：

- ◇ 初始化每条半边为一个初始面环，根据边的邻接关系为每个面环设置优先级；
- ◇ 根据二流形体的拓扑约束，按优先级对面环进行合并；
- ◇ 当一个闭合的面环形成的时候输出二维平面。

我们小组实现的算法与以往的二维搜面算法相比在准确率和算法复杂度上都有明显的提高。

之前常用于二维搜面的算法主要有：Dijkstra 算法，同样由 Peter A.C. Verley 在早期研究得到，能够较快搜索得到二维平面，然而算法准确率上还有所欠缺；Hybrid GA 找面算法，是由香港中文大学的 J.Liu 提出，主要是借鉴染色体杂交遗传下一代的生物学方法，来迭代得到最有可能的面环组合，算法的准确性依赖迭代次数以及每代染色体的筛选，而普遍迭代次数约为 20~50 次，算法效率不高。

当然小组系统实现的算法本身也存在一定的缺陷，首先是对于某些复杂二维线框图算法目前尚不能搜索到所有正确的二维平面；其次算法针对的基本都是平面形体，对一些基本曲面形体还不适用，这将是今后继续探索的方向。

3. 三维坐标重建

重建所有点的 3D 坐标的算法对于整个流程来说也非常重要，大多数研究人员将其视作根据目标函数求最优解的等价问题，因此目标函数的标准选择就非常重要。这里采用了 L.Cao, J.Liu 和 X.Tang[2]的论文中的目标函数的构造方式，其目的主要是为了优化取不同坐标时所表现出的对称性、平面性。

$$3.1 \text{ 对称性计算: } WS = \sum_i^n \frac{A_i}{P_i^2}$$

其中 A 表示某个平面的面积，P 表示其周长。WS 越大表示该形体越对称。

具体实验过程中，因为当前代入目标函数的三维坐标不能保证每个面在三维空间内确实是平面，除非该面只有三个点。所以在真正计算时，要先将每个面进行三角剖分，剖分成一个个三角面，然后用三角面面积计算公式算出每个三角面的面积，再求和以表示整个面的面积。

$$3.2 \text{ 角度的方差计算: } SDA = \text{Var}(\theta_1, \theta_2, \dots, \theta_k)$$

其中，k 表示线框图中所有相邻边所形成的角的总数目， θ_i 表示第 i 个相邻边形成的夹角的角度，Var 表示标准差。SDA 越小表示整个形体的角度平均度越高，在很多文章中，都曾经用 MSDA (minimum standard deviation of angles) 表示形体的对称性。

3.3 平面性计算

每一个面都可以用 $f = (a, b, c)$ 来表示，使得对于任意在面上点 $V_i (x, y, z)$ 满足：

$$\begin{cases} ax + by + cz - 1 = 0 \\ V_i^T f = 1 \Rightarrow Vf = 1 \end{cases}$$

上式对平面上的 m 个点都成立，其中 $V = (V_1, V_2, \dots, V_m)^T \Rightarrow f = (V^T V)^{-1} V^T 1$

在得到 f 后，每个面的平面性计算如下：

$$DP = \frac{1}{a^2 + b^2 + c^2} \sum_{j=1}^m (ax_j + by_j + cz_j - 1)^2$$

最终的三维坐标重建的目标函数为：

$$f(x) = \lambda \left(\frac{1}{WS} + SDA \right) + (1 - \lambda)DP$$

这里 λ 的取值依次为 $1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}$ ，基本意义是先主要关注整个形体的对称性，再微调使得每个面保持平面性。最后采用爬山（hill-climbing）算法来求的目标函数求最优解。这个算法可以高效的重构形体的z坐标，但是顶点数目较多的情况下，容易出现局部最优解。

4 三维形体显示

显示部分采用 OpenGL+MFC 实现。主要构成为一个 MFC 对话框类，在其中包括了 OpenGL 的显示和刷新过程代码。MFC 对话框的显示部分被替换为 OpenGL 的显示模式，在每次重绘的时候进行 OpenGL 的方式进行绘制，并在绘制函数的最后再次触发重绘事件，以保证对话框内的内容实在不断的刷新。

显示部分接受重建之后得到的三角网格化后的模型信息。将这些数据以网格信息的形式存入一个显示列表以及边的信息存入另一个显示列表，并在显示过程进行读取和绘制，并可以进行线框模型和实体模型的切换。在显示的过程中，响应鼠标事件，利用一个轨迹球进行模型的旋转变换。轨迹球利用四元数进行旋转角度的计算，避免了欧拉角所造成的“万向锁”问题。在显示中，由于面片重构的时候并不能保证所有的面都是以逆时针方向排列顶点，且在处理过程中不能指定一个正方向，这样利用普通的光照方程无法正确的计算每个面的颜色值，所以我们采用了一种伪光照的方法来进行显示：对每个面随机赋一个灰度值做为该面的颜色，这样在应用中就产生了一种类似光照的效果。

5. 实验总结

最后我们小组的工作实现了开题时所有的基本目标，能够较为准确完整地将一个单个的二维线框图重建成为一个三维的实体。

- ◇ 系统能够实现简单形体的单视图重建。包括二维面查找算法以及三维坐标重建算法的实现。其中形体只包括多面体，不包括曲面形体的重建，也不考虑存在孔洞的复杂形体。
- ◇ 系统需要具有良好的用户界面。能够通过基本的用户交互操作，来逐步实现形体的重建。每一步都能给用户提供良好的图形结果以及信息查询。例如，初始的二维形体的显示以及

最后三维形体在光照模型下的动态显示效果等。

- ◇ 系统需要附带足够以及能够体现系统重建功能的重建模型样例。同时为用户提供良好的样例输入以及输出界面。

我们小组的前期研究遍布了目前重建领域中的各个最新的研究成果，在其中挑选了目前算法最优的论文进行了很好的实现。同时我们对论文中的一些不足也进行了一定的修改，增加了算法的运行速度。

然而为了达到较好的三维坐标重建效果，我们的算法会进行多次微调和迭代，程序的运行速度不能得到很大的提升，对与一些点和边比较多的复杂模型，算法运行时间不可估计，这是我们需要进一步修改和研究的内容。同时坐标重建的算法有时会陷入局部最优解而导致三维模型生成效果较差，此时需要我们重复进行三维坐标重建过程，得到较好的效果。

我们小组还对开题时的一些扩展目标进行了研究：

- ◇ 算法能够处理复杂形体的单视图重建。包括一些孔洞，拼接的复杂形体结构，主要可以采用切割形体的方法进行实现。
- ◇ 算法能够处理一些基本二次曲面形体的重建。这个目前尚没有很好的解决方案，可以进行深入的后续研究

对于第一个扩展目标我们进行了基本的实现，最后已经能够根据复杂形体成功找到合适的切割面。但由于组员的时间协调关系，最后的分割体合并尚未实现，我们可以在后续进行补充。对于第二个扩展目标，我们研究了一些目前国际上最新的论文和算法，但是很多算法在曲面的识别率和曲面表示上都存在一定的缺陷，重建效率也不高，目前尚没有比较公认的算法来对单个二维线框图的曲面形体进行重建。

6. 小组成员及分工

| 姓名 | 学号 | 工作内容 |
|-----|------------|---|
| 李琦 | 2010210804 | 1. 前期准备，实验选题，小组论文查找； 2. 基本数据结构，基本实验框架搭建； 3. 二维平面查找算法实现； 4. 实验总结报告撰写。 |
| 王宇翔 | 2010210802 | 1. 前期准备，实验选题，小组论文查找； 2. 基本数据结构，基本实验框架搭建； |

| | | |
|-----|------------|--|
| | | 3. 三维形体显示算法实现; 4. 实验总结报告撰写。 |
| 林建立 | 2010210777 | 1. 三维坐标重建; 2. 平面三角剖分; 3. 实验总结报告撰写。 |
| 彭艺 | 2010310569 | 1. 切割部分算法部分实现; 2. 实验总结报告撰写; 3. 实验结果网页制作。 |

7. 参考论文列表

- [1] Peter A.C. Verley*, Pedro P. Company. A new algorithm for finding faces in wireframes. *Computer-Aided Design*, 2010, 42:279-309
- [2] J Liu, X Tang. Evolutionary Search for Faces from Line Drawings. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2005, 27(6):861-872
- [3] Tianfan Xue, Jianzhuang Liu, Xiaoou Tang. Object Cut: Complex 3D Object Reconstruction Through Line Drawing Separation. *CVPR 2010*: 1149-1156
- [4] Y. Chen, J. Liu, X. Tang. A Divide-and-Conquer Approach to 3D Object Reconstruction from Line Drawings. *Computer Vision*, 2007. *ICCV 2007*: 1-8
- [5] Marill, T. Emulating the Human Interpretation of Line-Drawings as Three-Dimensional Objects. *International Journal of Computer Vision*. Vol. 6, No. 2, 1991, pp. 147-161.
- [6] Leclerc, Y. and Fischler M. An Optimization-Based Approach to the Interpretation of Single Line Drawings as 3D Wire Frames. *International Journal of Computer Vision*. Vol. 9, No. 2, 1992, pp. 113-136.
- [7] Lipson H. and Shpitalni M. Optimization-Based Reconstruction of a 3D Object from a Single Freehand Line Drawing. *Computer Aided Design*. Vol. 28, No. 8, 1996, pp. 651-663.