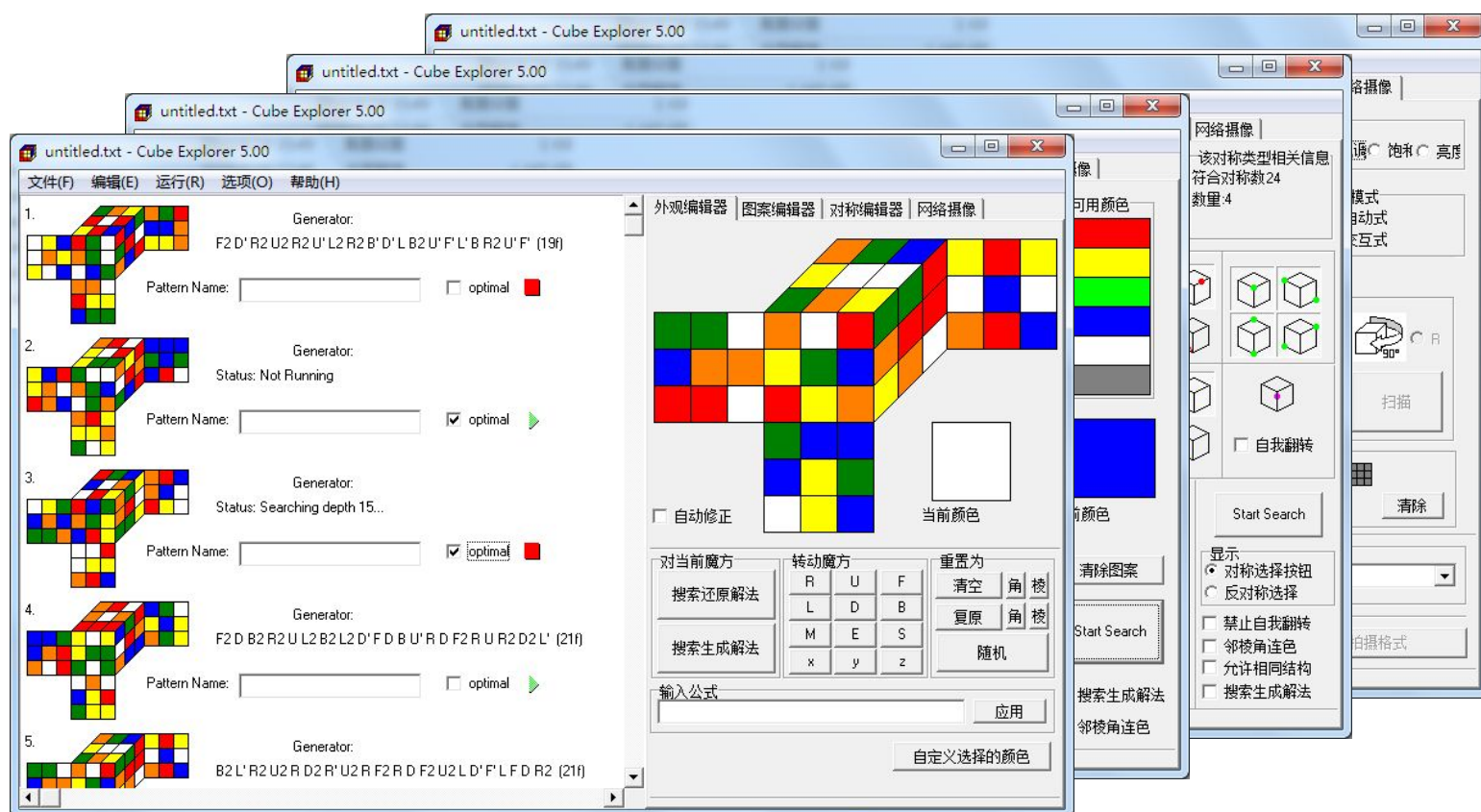


Cube Explorer 5.00

中文操作手册



作者：Herbert Kociemba

汉化：zbyxzh

前言

本手册的文字内容主要来源于 Cube Explorer 5.00 版附带的帮助文档，原作者为 Herbert Kociemba。考虑到实用性和普及率，本手册只对说明文档的“Overview”和“Exploring the Cube”部分进行了翻译。“The Mathematics behind Cube Explorer”等章节的内容请参考后续的汉化资料。

本手册推荐配合 Cube Explorer 5.00 汉化版进行使用。该汉化版下载地址为：<http://bbs.mf8.com.cn/viewthread.php?tid=50160>

本手册汉化作者为 zbyxzh，本汉化只为个人兴趣和技术交流所作。欢迎大家就 Cube Explorer 的使用心得展开讨论和交流，也欢迎在本手册的发布帖上进行提问，我会尽己所能提供帮助！

由于时间较紧，加上本人才疏学浅，翻译过程中难免出现错误和遗漏。请各位朋友不吝赐教！我的 E-mail: zbyxzh@126.com

zbyxzh

2010 年 4 月

目录

第一章 概述	4
基本定义	5
求解魔方的算法	6
二阶段算法简述	8
第二章 魔方探索	10
了解 Cube Explorer 的设计	10
安装 Cube Explorer	11
外观编辑器	12
图案编辑器	14
对称编辑器	16
网络摄像页面	19
主窗口	23
编辑菜单	25
运行菜单	27
选项菜单	28
求解不完整魔方	31
Cube Explorer 接口	35
使用帮助系统	38
第三章 附录	39

概述

Cube Explorer，并不像其他许多软件一样，仅能对魔方进行模拟，或者采取冗长的策略对打乱的魔方进行还原。Cube Explorer 能够运行一种精妙而又强大的算法（二阶段算法）。这种算法通常能在短时间里提供出一组平均步数在 19 步之内的解决方案，仅比实际最少的步数多了一步到两步。Cube Explorer 同时还能执行另外一个算法，用于寻找最小步解法——无法进一步优化的解法。借助一台 2010 年硬件水平的个人电脑，寻找这样一个最佳解法通常仅仅需要花费几分钟的时间。

Cube Explorer 也可以帮助你来寻找漂亮的魔方图案（同时提供一个简洁的步骤来实现它）。Cube Explorer 会在全部可能的魔方图案中进行搜索，并将所有符合要求的魔方列举出来。

请注意：Cube Explorer并非源于“Rubik’s Cube”的商标持有者，也没有从商标持有者处得到协助、认同或是赞助。该商标的持有者是Seven Towns 有限责任公司，它是Rubik’s Cube三维智力玩具的制造商和全球销售商，也是其官方网站<http://www.rubiks.com>上电子版本的提供者。

Cube Explorer 是一个非营利性的教育产品，它是免费软件，也是科学研究的成果。

基本定义

一个魔方是由 8 个角块、12 个棱块以及 6 个中心块构成的。棱块可以被翻转，角块也可以被旋转。那就意味着这些块的颜色朝向是可以在空间中改变的。

魔方的六个面被称为 **U** (上)、**D** (下)、**R** (右)、**L** (左)、**F** (前) 和 **B** (后)。在描述解法时，**U** 表示魔方的上层顺时针旋转 90 度，**U2** 表示旋转 180 度，**U'** 表示逆时针旋转 90 度。

从 Cube Explorer 5.00 版开始，中层转动 **E**、**M**、**S** 以及整体转动 **x**、**y**、**z** 也可以被软件识别。你可以点击外观编辑器中的转动按钮来理解他们的含义。**E'**、**M'**、**S'**、**x'**、**y'** 和 **z'** 就是它们的反方向转动。

一个由多次转动组成的序列，例如 **U D R' D2**，被称作一个**解法**。转动的步数被称作解法的长度。一个解法可以是一个**还原解法**或是一个**生成解法**。使用还原解法可以将一个被打乱的魔方还原成未打乱的状态；而生成解法则是还原解法的逆执行——它将一个未打乱的魔方进行打乱。当你想要玩出漂亮的魔方图案时，生成解法就会发挥功能。

求解魔方的算法

最简单的魔方还原策略需要经过多个阶段完成对魔方的还原。每个阶段对一定数量的块的位置或朝向进行修正。随着对还原策略的执行，你会发现每一面都变成了统一的颜色。这种策略非常适合手动还原，但是由于它的解法长度较大，因此不是最佳方法，原因是它没有考虑到魔方在数学上的群结构。实现这类算法通常需要超过 70 步的转动。

第一个考虑到魔方群结构因素的算法，是由数学家 **Morwen Thistlethwaite** 在 1980 年开发出来的。他将魔方的状态划入 4 个子群，而进行降群时所允许转动也被限制在四个小组中。计算机是实现这种算法的最佳工具，而且还原一个魔方最多不超过 52 步。

在 1990 年，**Hans Kloosterman** 通过构造与之前不同的转动限制组，将这种算法的还原步数上限降低至 42 步。

在 1992 年，我在一台内存为 1MB 的计算机上开发出了二阶段算法。这种还原策略的步数上限无法精确给出，因为它单独取决于魔方的不同状态，而检查所有的魔方状态是不可能完成的。但是在一般情况下，你可以在短时间内得到一个平均不超过 19 步的解法。

1997 年 5 月，借助人工智能，**Richard E. Korf** 开始运用启发式搜索理论的概念来解决特定的魔方，并且找出了 10 个随机生成魔方的最少步解法。这个程序在一台 **Sun Ultra-Sparc Model 1** 工作站上运行，需要 82MB 的内存。每寻找一个魔方的最少步解法平均需要花费 5 天的时间。在此之前，寻找一个最少步解法需要耗时数月甚至数年。之后发现，他所使用的单阶段算法和我在二阶段算法中每一阶段所使用的算法是十分接近的。

1997 年 7 月，借助于魔方的对称性，**Micheal Reid** 做到了高效地存储更多的数据。他所使用的数据库可以将二阶段算法的第一阶段储存起来，使算法可以像单阶段算法一样快速运行。他在一台有着 128MB 内存和 200MHz 奔腾处理器的计算机上运行了他的程序，结果该程序搜寻最少步解法的速度是 **Korf** 的程序的

20 倍。

在 2001 年，我改进了二阶段算法，使它能够与个人计算机硬件水平的提升相匹配（使用 128MB 内存）。第一阶段同样直接被储存起来，所以它能运行地非常快。程序的最少步解法部分会将第一阶段当做单阶段算法来使用，进而可以采用与 Micheal Reid 接近地方法寻找最少步解法。

像 Windows XP 这样的 32 位系统，4GB 是它的内存上限，于是我将最少步解法部分设计成需要使用这么大的内存。它的运算速度是 2001 年时第一阶段最少步解法的 15 倍。使用 Cube Explorer 特别版并在设置中开启无限最少步求解器便可体验这一算法。在 Windows Vista 64 位操作系统下，并使用 Intel Core i7 四核处理器，最少步求解器可以在 1 小时内搜索出大约 290 个随机打乱的魔方的最少步解法。

二阶段算法简述

下面的介绍是为了能让你大致了解到这一算法是怎样工作的。它有一定的技术要求，而且理解起来可能会有困难。如果你仅是想要使用 Cube Explorer，下面的信息不是必须的，也没有必要为了使用程序而去理解这一算法。

对于一个未打乱的魔方，如果你使用除 R, R', L, L', F, F', B 和 B' 以外的转法来转动它，你能生成的状态仅是魔方所有可能状态群中的一个子群。这个子群表示为 $G1 = \langle U, D, R^2, L^2, F^2, B^2 \rangle$ 。在这个子群中，角块和棱块的朝向是不能被改变的。也就是说，当一个棱块或是角块处在一个特定位置时，它的色向总是一样的。同时，UD 夹层（U 层和 D 层中间的那一层，即中层 E）上的棱块始终位于该夹层上。

在第一阶段，算法是在寻找这样一个解法：它能将被打乱魔方的状态转动成为 $G1$ 群内状态。即：角块和棱块的色向会被限制，且原本属于 UD 夹层的棱块必须被转至该夹层上。在抽象空间内，一次转动仅仅将魔方的状态坐标 (x, y, z) 转换为 (x', y', z') 。 $G1$ 内的所有魔方都有着相同的状态坐标 $(x0, y0, z0)$ ，这就是第一阶段的目标状态。

为了搜寻这一目标状态，本程序所使用的搜索算法，用当前的研究术语，被称为迭代加深启发式搜索算法（IDA*）。对于魔方，这就表示要对增长的解法步数进行迭代。启发式函数 $h1(x, y, z)$ 会估算出对于一个状态为 (x, y, z) 的魔方，其要达到目标状态所必需的步数，而避免对步数的高估是十分必要的。从 Cube Explorer 2 起，程序可以准确提供魔方在第一阶段中达到目标状态所需的步数。该函数在生成解法的过程中允许进行剪枝。如果你不想在达到目标状态前花费太多时间的话，这是十分必要的。启发式函数 $h1$ 是一个基于内存中数据的查询表，并且可以预先剪枝掉 12 步转动。

在第二阶段，算法通过执行被允许的转动，将已属于 $G1$ 子群的魔方还原。解法将还原位于 U 面和 D 面的 8 个角块、8 个棱块的位置，以及 UD 夹层中四

个棱块的位置。启发式函数 $h2(x, y, z)$ 仅估计到达还原态所需的步数，因为 $G1$ 群中的元素实在是太多了。

当算法找到第一个完整解法后并不会立即停止运算，而是对第一阶段其他非最佳（最少步）解法进行第二阶段运算。例如，若第一个完整解法包括第一阶段的 10 步，及第二阶段的 12 步；第二个完整解法可能包括第一阶段的 11 步，及第二阶段的 5 步。第一阶段的解法步数在增加，第二阶段在减小。当第二阶段的步数达到 0 时，完整解法便是最少步解法，算法至此停止运行。

当前的二阶段算法不会主动搜索那些需要多次进出第二阶段才能得出的最少步解法。这样的设计会极大的提高速度。如果想要得到最少步解法，你可以使用最少步求解器。

魔方探索

了解 Cube Explorer 的设计

Cube Explorer 的程序界面包括主窗口和多个分开的页面：

这些页面包括：

- 1、外观编辑器，你可以编辑魔方的着色、对魔方进行旋转并进行还原；
- 2、图案编辑器，帮助你创造漂亮的魔方图案；
- 3、对称编辑器，帮助你寻找那些具有对称性和反对称的魔方；
- 4、网络摄像，可以借助网络摄像头对魔方的着色进行输入。

各页面的输出结果将会在主窗口中显示。

Cube Explorer 的主要功能是对所给的或是计算所得的魔方，进行步数较少的生成解法和还原解法的搜索。生成解法可以让一个未打乱的魔方经过转动，变成主窗口所显示的状态。相对的，借助还原解法可以将主窗口所显示的魔方还原。

Cube Explorer 同样可以用来搜索两个特定状态间的转换解法，这对我们手动一步步还原魔方是十分必要的。

安装 Cube Explorer

Cube Explorer 的运行需要三个文件: 主程序 `cube5xx.exe`, 说明文件 `cube.chm` 和可供选择的 Java 应用程序 `AnimCube.jar`。`cube.ini` 文件中则包含了一些有用的默认参数。你应当把所有文件放在同一文件夹内。第一次运行时, 程序会计算并生成一些表格文件, 并储存在你的 Cube Explorer 文件夹内, 生成文件所需的时间与你的计算机配置相关, 最长可能会耗时 30 分钟。下次运行时, 这些表格会直接从硬盘读取到内存之中, 速度非常快。运行基本功能所需的表格文件会占用 70MB 的内存。为了求解魔方, Cube Explorer 使用了一个非常强大的算法, 而这些表格文件是运行算法所必需的。

Cube Explorer 不会在自己所在的文件夹之外替换或生成文件。所以当你想要卸载 Cube Explorer 时, 删除整个文件夹即可。

外观编辑器



编辑外观是十分简单的，但是有些特色功能却并不明显。现总结如下：

左键单击中心块色片以选择颜色；

左键单击任意其他色片，用当前颜色进行着色。配合 **Shift 键**和 **Ctrl 键**，可以对不完整的魔方进行额外设置（详见求解不完整魔方）；

右键单击中心色片可以调整中心块的朝向（需在**选项菜单**中开启**求解时考虑中心块朝向**）；

右键单击任意其他色片可以删除当前着色。

当你想要输入一个给定的魔方并求解时，先选择**复原**会比较方便。

单击**搜索还原解法**，当前魔方会被添加进入主窗口，并使用二阶段算法对魔方求解。单击**搜索生成解法**，则生成器会开始计算。你可以通过点击主窗口中的**Solver:/Generator:**按钮在两种解法之间进行切换。(Solver 为还原解法，Generator 为生成解法)。

如果你想要验证算法的表现，不妨使用**随机**按钮。所有产生的随机魔方的出现几率都是一样的。

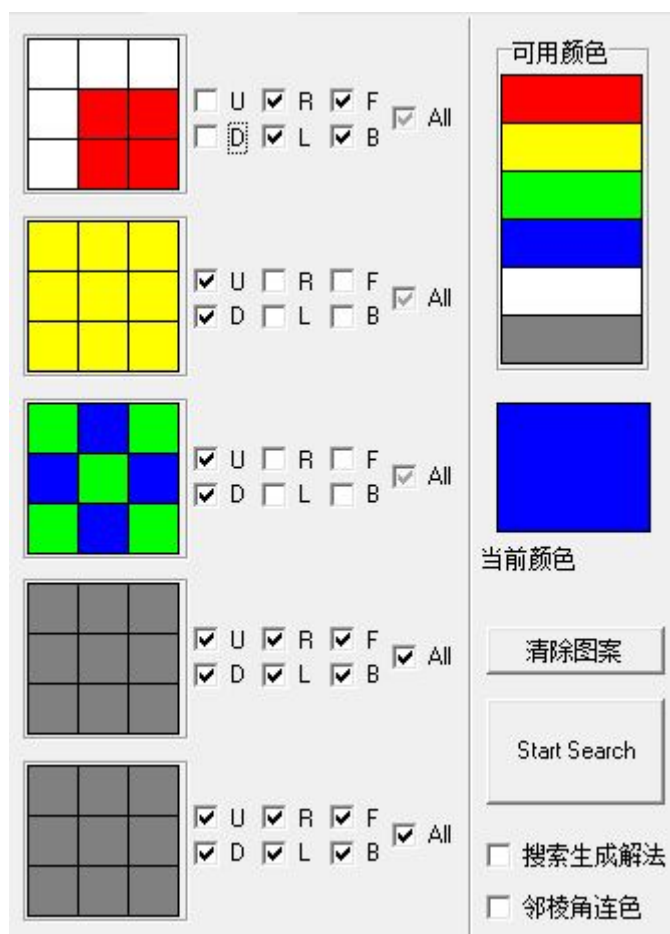
你可以在**输入公式**框内输入或是粘贴公式。

左键单击转动魔方内的按钮，可以对编辑器内的魔方执行所对应的转动。**右键单击**则执行相应的反方向转动。

不同的魔方配色方案也是不同的。你可以选中一个中心块，单击**自定义选择的颜色**对当前颜色进行修改。

如果你开启了**自动修正**，那么在对色片进行着色过程中，程序会自动纠正颜色错误。有时它会非常好用，有些情况下也很烦人。

图案编辑器



如果你想要转出一个有着漂亮图案的魔方，你可以使用图案编辑器。

你必需要理解，你在**可用颜色**中选择的颜色，与魔方的实际颜色无关，它是用来表示颜色分布结构的。上方所展示的例子，是用来寻找这样的魔方：它的 R、L、F 和 B 四面都有一个 2×2 的方块（在 1 号图案模板中设定）；而它的 U 面和 D 面，则仅有一种颜色（在 2 号图案模板中设定），或者是有一个棋盘的图案（在 3 号图案模板中设定）。如果在一个模版中选中 **ALL**，且所有的色片都是灰色的（例如上图最后两个模板），那么程序会自动忽略这一模板。

如果你选中了**搜索生成解法**，程序会调用二阶段算法对符合图案要求的魔方求解。如果结果中有**同构魔方**，主窗口只会显示其中一个。

如果两个面在相交的边上相互匹配，这种图案被称为**邻棱角连色**。即：两个相邻的棱块和角块，当且仅当它们在其中一个公共面上的着色相同时，它们在另一个公共面上的着色也相同。邻棱角连色的图案通常都是你所期待的漂亮图案中最好的选择。

对称编辑器

选择对称类型 (共33类)



D3

该对称类型相关信息
 符合对称数6
 数量:432

当前对称类型





























☐ 自我翻转

对称选项

☒ 严格对称
 ☐ 允许其它对称

每面允许出现的颜色数

☒ 1
 ☒ 2
 ☐ 3
 ☐ 4
 ☐ 5
 ☐ 6

对称参与范围

☒ 整个魔方参与
 ☐ 仅棱块参与
 ☐ 仅角块参与
 ☐ 忽略角块
 ☐ 忽略棱块

Start Search

显示

☒ 对称选择按钮
 ☐ 反对称选择

☐ 禁止自我翻转
 ☐ 邻棱角连色
 ☐ 允许相同结构
 ☐ 搜索生成解法

通过对称编辑器，你可以寻找那些具有对称性和反对称性的魔方。这些魔方在进行相应的对称操作时是不变的。

当你将一个魔方映射到另一个魔方上时，一共有 48 种几何旋转、反射等。这些映射的子集定义了魔方对称性的子群，其中一共有 33 个基本子群。你可以通过点击对称选择按钮，或是在**对称类型选择**的下拉菜单中进行选择。你可以选择主窗口中的魔方，并且对它的对称性和反对称性进行分析。如果这个魔方具有某一对称性，那么相对应的对称选择按钮会被按下；如果具有反对称性，相应的按钮则会升起。

点击 **Start Search**, 程序将会计算出所有遵守当前所选对称规则的魔方。

选中**允许其它对称**，那么被找出的魔方也允许拥有没有选中的对称性。



选择**每面允许出现的颜色数**，对魔方的颜色进行设定。

如果你仅想分析棱块或是角块，你可以在**对称参与范围**内进行选择。你可以仅考虑一类色块的对称，或是完全忽略它。

禁止自我翻转可以去除那些是由主窗口内已输出魔方通过自我翻转而形成的魔方。

对**邻棱角连色**和**搜索生成解法**的解释，请参考图案编辑器一章。

若想对反对称性进行探索，你可以右键单击一个对称选择按钮，选择**应用为反对称**；或者在**显示**中选中**反对称选择**。之后的介绍如下：

选择对称类型 (共33类)		该对称类型相关信息
 D3		符合对称数6 数量:432
Subgroup of Index 2		
 C3		
<p>在第一个下拉菜单中选择点群G。 在下方的下拉菜单中选择一个G的子群H。 程序会搜索出满足下面属性的魔方c： 1. H中的所有元素都在c的对称群中； 2. 对魔方c执行G中任意一个元素的操作会得到其反向魔方。</p>		
对称选项		Start Search
<input checked="" type="radio"/> 严格对称 <input type="radio"/> 允许其它对称		
每面允许出现的颜色数		
<input checked="" type="checkbox"/> 1 <input checked="" type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5 <input type="checkbox"/> 6		显示
对称参与范围		<input checked="" type="radio"/> 对称选择按钮
<input checked="" type="radio"/> 整个魔方参与		<input checked="" type="radio"/> 反对称选择
<input type="radio"/> 仅棱块参与 <input type="radio"/> 仅角块参与		<input type="checkbox"/> 禁止自我翻转
<input type="radio"/> 忽略角块 <input type="radio"/> 忽略棱块		<input type="checkbox"/> 邻棱角连色
		<input type="checkbox"/> 允许相同结构
		<input type="checkbox"/> 搜索生成解法

如果经过一个对称操作 S ，魔方被转换成了它的反向状态，那么我们就称这个魔方对于 S 是反对称的。对一个魔方执行对称操作 S 和其逆操作 T ，便可再次得到原来的状态。

对于上方下拉菜单中的每一个对称类型，下方的下拉菜单中通常都有许多个它的子群。这两个选项共同定义了一个确定的反对称类型。

网络摄像页面

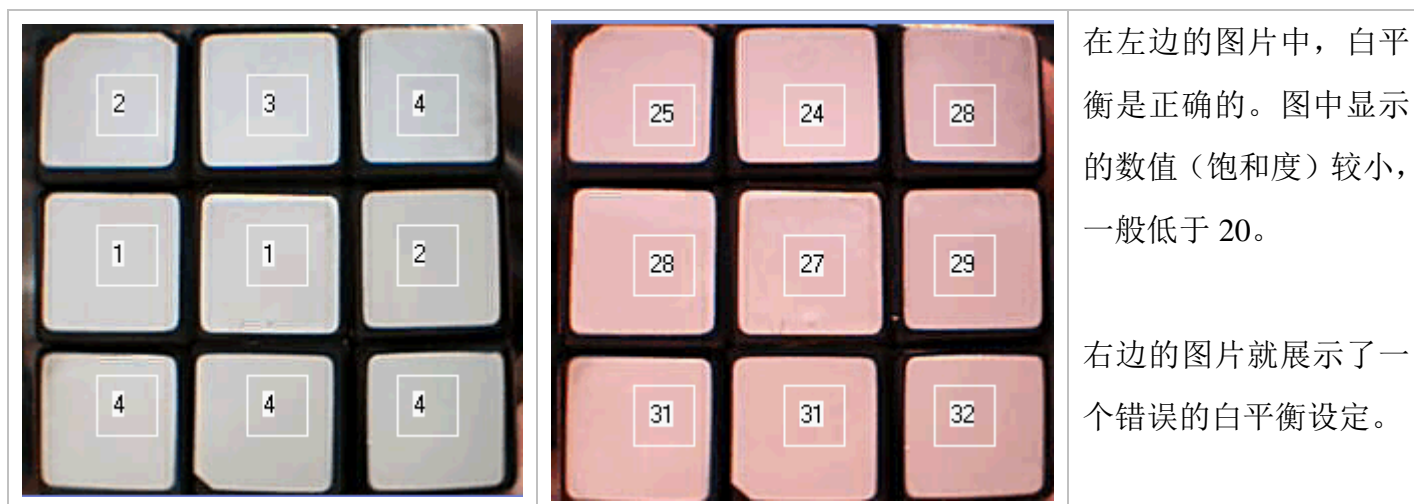
除了使用外观编辑器，你还可以借助网络摄像头来读取魔方的各面颜色。这种方法非常快捷，但是如果想要使用成功，还有一些非常重要的注意事项。

1. 设置网络摄像头

在**摄像设备**下拉菜单中选择摄像头。请在启动 Cube Explorer 之前确保你的摄像头已经连接到电脑，否则它将不会出现在列表中。

单击**摄像设备设置**按钮来调节白平衡、曝光、增益和饱和度。

将一个未打乱的魔方的白色一面或是一张白纸放在摄像头前面，直到它设置正确。必要时可以开启自动白平衡。标准如下：



之后**关闭**自动白平衡。如果你不关闭自动白平衡，魔方的颜色就无法正确识别。所以不要忘记这一步！

设置曝光及增益，使颜色不要过亮。标准如下：



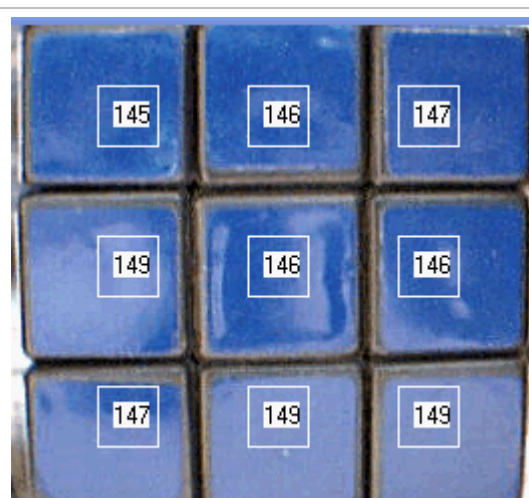
如果你在摄像窗口的左上角看到这样一行信息："Decrease expose or gain for optimal results!", 说明你的设置存在问题。**关闭**自动曝光和增益, 否则就会出问题了。

有时, 适当提高饱和度会起到一定效果, 但这一步不是必需的。

2. 设置识别参数

将**识别模式**设为**自动式**, 并在**显示**中选择**色调**。

将不同的颜色放在摄像头前



橙色或红色色片的色调值必须低于 20 或者高于 220, 黄色的色调低于 60, 绿色低于 125, 蓝色低于 220。如果和你的魔方不吻合, 你可以改变颜色的**色调上限**。一般情况下, 默认值都是通用的。左边的例子中, 蓝色的色调就是正确的。

下面, 我将说明如何分辨红色和橙色, 因为它的方法取决于你所选择的识别模式: 自动式和交互式。

如果你选择了**交互识别模式** (在这种模式下, 扫描过程中反馈的信息最多), 除了红色和橙色色片 (有在窗口左下角尚有信息提示时), 其他颜色都能立即被识别。右键点击识别错误的色片, 你就可以选择正确的色片颜色了。只要你对橙色和红色各进行了 2 个以上的正确采样, 左下角的信息就会消失。在合适的光照下, 如果橙色色片被识别成为红色, 你可以一直进行手动取样 (最多 9 个)。

参数识别

色调上限

橙色: 20 黄色: 60

绿色: 125 蓝色: 220

样本数

红: 橙: 清除

显示

☒ 色调 ☐ 饱和 ☐ 亮度

识别模式

☐ 自动式 ☒ 交互式

当前扫描面

☐ B ☒ L ☐ F ☐ R

☐ U ☐ D

扫描

还原被扫描魔方

扫描预览

清除

摄像设备

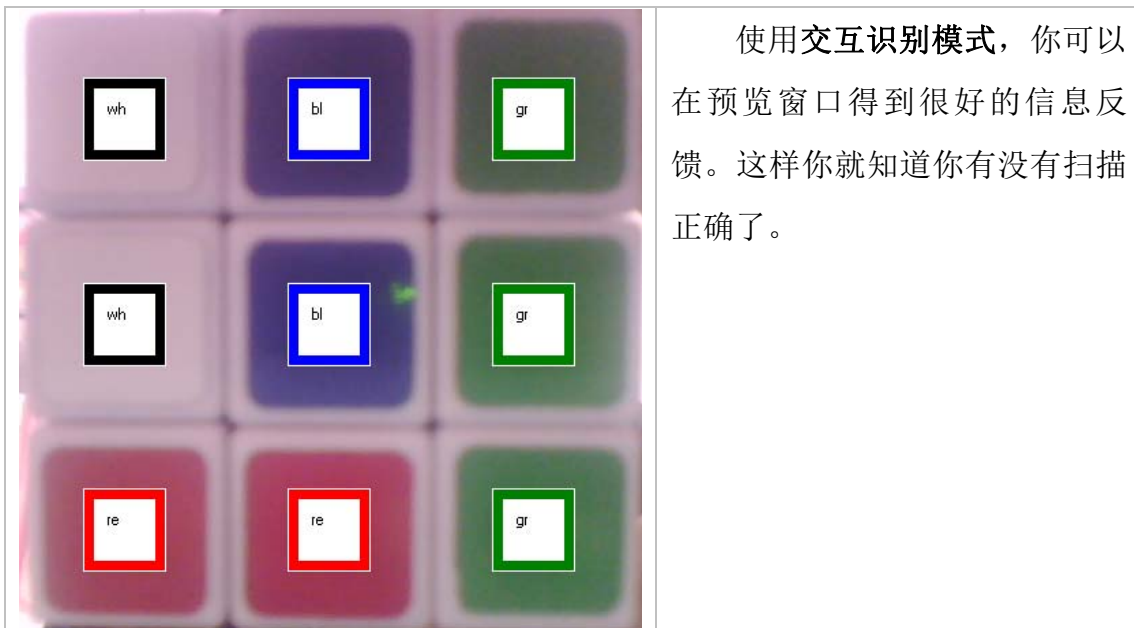
Integrated Camera

摄像设备设置 拍摄格式

如果你选择了**自动识别模式**，在点击**还原被扫描魔方**之前，程序不会对魔方的颜色进行分析。红色和橙色的区别取决于它们的色调和亮度。如果红色和橙色的色调过于接近（你可以通过开启**显示**中的**色调**来查看），那就选择**亮度**来替代。

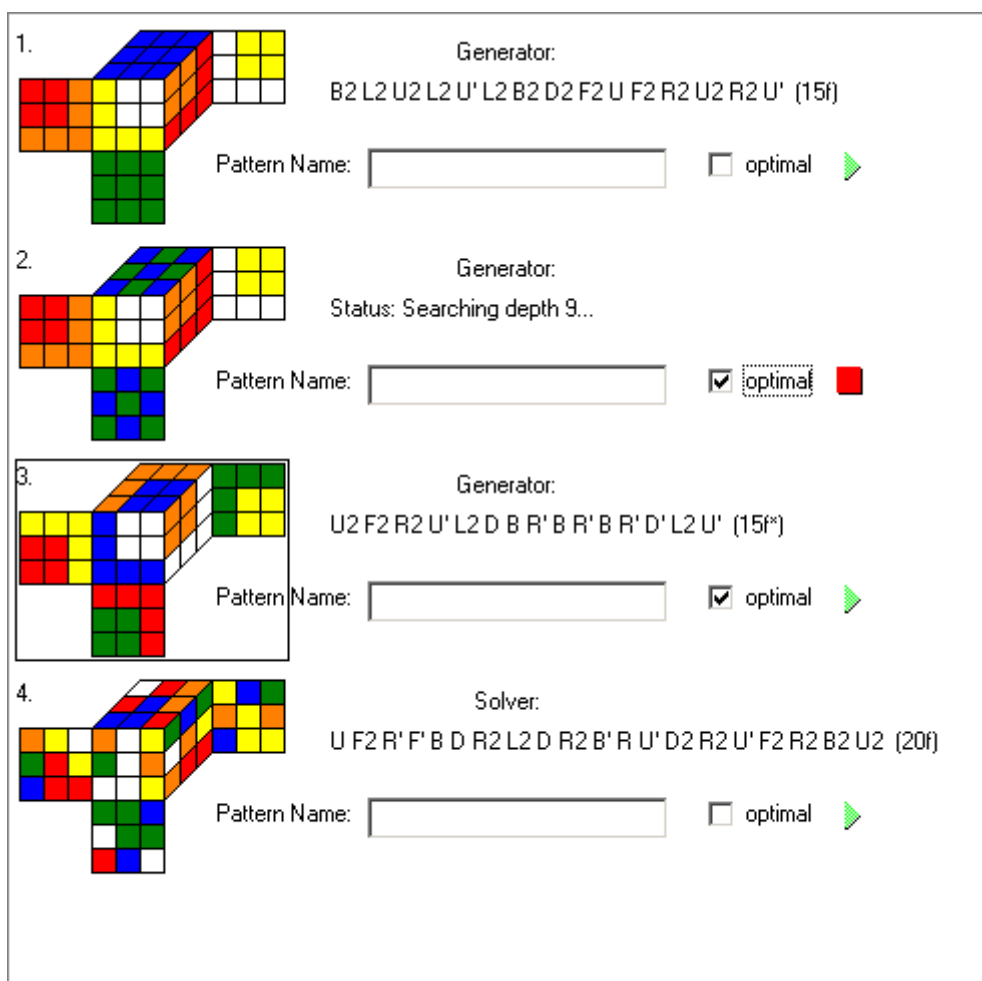


如果你**扫描**一个魔方，那么程序会从**B**面开始扫描，如左图所示，在点击**扫描**按钮开始下一面之前，你看到的应当是魔方的**U**面和**F**面。若要扫描**L**面，你需要按照**当前扫描面**中**B**和**L**之间的图示旋转魔方，然后再按**扫描**按钮继续，直到整个魔方扫描完毕。





最后，点击**还原被扫描魔方**。如果所有的色片都是有效的，那么程序就会开始计算解法。

主窗口



在主窗口中，有下面几种操作方式：

- 点击  键，可以开始或者继续魔方求解算法。
- 点击  键，可以停止一个正在运行的计算。Cube Explorer 是一个多任务程序，你可以在同一时间内进行多个计算。在关闭 Cube Explorer 之前，你应当停止所有正在运行的计算。
- 如果你选中 **optimal** 选项，程序会计算该魔方的最少步解法。这可能会花费一段时间，同时当前的搜索深度也会显示出来。（如上图第二个魔方）
- 如果最少步求解器找到了一个解法，那么这个解法步数处会有一个“*”标记。（如上图第三个魔方，解法步数为 15f*）

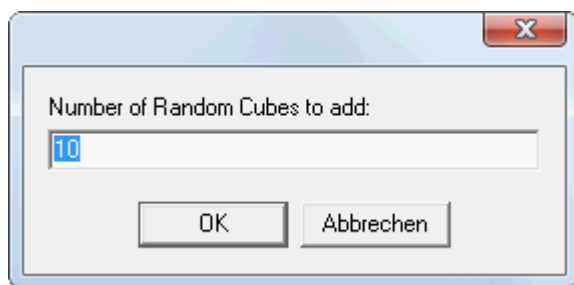
- 点击 **Generator** 或者 **Solver**，可以在生成解法和还原解法之间进行切换。
- 若要删除主窗口中的一个魔方，你可以左键单击选择它，然后按 **Delete** 键删除；或是菜单栏**编辑——删除选中的魔方**（也可以在主窗口无魔方处单击右键，在弹出的菜单中选择**删除选中的魔方**）。如果想要一次删除多个魔方，那先要选择它们。当你单击选中一个魔方时，该魔方周围会出现一个黑框。你可以像日常的系统操作一样，借助 **Shift**、**Control** 和 **Shift-Control** 键来扩大选择范围。
- 当最少步求解正在运行时，将鼠标移动到 **optimal** 选项上，你可以看到 **Cube Explorer** 在搜索树不同层面计算上所用的时间和结点数。

<div>删除选中的魔方(D)</div> <div>开始搜索选中的魔方(S)...</div> <div>停止搜索选中的魔方(T)</div> <div>整理选中的对称魔方(A)</div> <div>为选中的魔方添加对称信息(Y)</div> <div>清空主窗口(C)</div>	<p>右键魔方以外的区域，会弹出如左图所示的菜单。相关解释请参考编辑菜单一章。</p>
<div>绕U-D轴旋转1/4周后使用公式(U)</div> <div>绕URF角旋转1/3周后使用公式(R)</div> <div>绕F-B轴旋转1/2周后使用公式(F)</div> <div>使用当前公式的左-右镜像公式(Q)</div> <div>使用当前公式的逆公式(S)</div> <div>将还原方法复制到剪贴板(T)</div> <div>将打乱方法复制到剪贴板(V)</div> <div>移除当前魔方(W)</div> <div>将该魔方转移至外观编辑器(X)</div> <div>转移对称至对称性编辑器(Y)</div> <div>运行傍系搜索器求解该魔方(Z)</div>	<p>右键单击一个魔方，会弹出另外一个菜单。你可以对其进行移除、旋转、镜像操作，或者将它移至外观编辑器作进一步修改。如果你对一个魔方进行旋转或是镜像操作，可能并没有得到你预期的效果。原因是该魔方在进行这两种操作的同时，也重新进行了着色，所以中心块的颜色能够保持不变。使用当前公式的逆公式将会给出当前魔方 A 的逆状态魔方 B，即 A 魔方的生成解法是 B 魔方的还原解法，反之亦然。将解法复制到剪贴板可以方便地在不同应用程序间进行粘贴。</p> <p>运行傍系搜索器求解该魔方是 3GB 特别版 Cube Explorer 的特有功能。被选中的魔方所定义的傍系中的 278691840 个魔方，程序用了不到两天时间便完成了对它们的最少步解法的搜索。也就是说每秒钟能搜索完成大约 1600 个魔方！</p>

编辑菜单



1. 删除选中的魔方	
2. 整理选中的对称魔方	选中的魔方会被旋转，从而在对称编辑器中的 33 种对称类型会用对称象形图显示色向。
3. 为选中的魔方添加对称信息	<p>选中的魔方的对称信息将会添加到图案名称一栏。</p> <p>例如：C2(b) {D2(edge)} 表示该魔方的对称类型为 C2(b)，同时还具有反对称 D2(edge)。</p> <p>{1} 表示自我翻转的魔方，所以 C2(b) {1} 表示该魔方的对称类型为 C2b，而且是自我翻转魔方。</p>
4. 添加随机打乱的魔方	<p>你可以最多向主窗口添加 100000 个随机打乱的魔方。如果你想一次求解多个魔方，那么不要忘记使用选项—立即释放线程这一功能。</p>
5. 加入所有反向同构魔方	主窗口中的所有魔方将被反转，并重新添加到主窗口中。



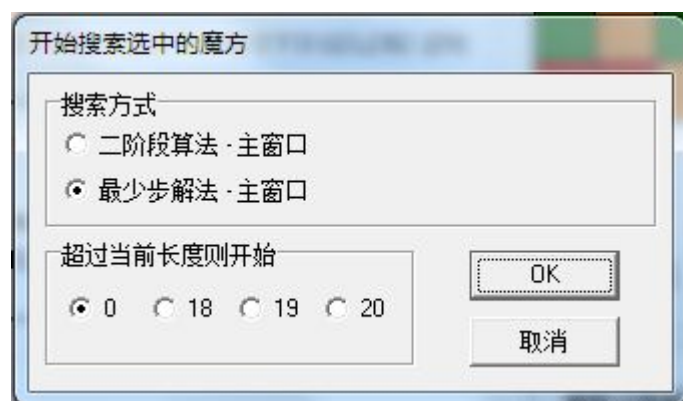
6. 生成 WCA 打乱公式	按照 World Cube Association 的比赛规则生成 5 个打乱的魔方。 如果你想要生成更多的打乱，你可以使用 添加随机打乱的魔方 ，并开启运行——自动开始二阶段算法。
7. 按图案名称排列魔方	主窗口中的魔方将按照你的选择进行排序。
8. 按解法长度排列魔方	
9. 清空主窗口	

运行菜单

通常情况下，搜索一个魔方的解法可以通过单击主窗口右侧的绿色箭头来完成。如果你想同时或者依次对多个魔方进行搜索，你可以用到这个子菜单



。1.开始搜索选中的魔方



你可以使用 **Ctrl** 和 **Shift** 键配合鼠标选择一个或者多个魔方。当这些魔方的当前解法步数超过你所选定的值，程序就会使用二阶段算法或最少步算法同时对它们求解。

2.停止搜索选中的魔方

停止当前搜索。

3.设置最大自启动线程数

Cube Explorer 支持多核处理器在同一时间内计算多个解法。可以在这里设置线程数。

4.自动开始二阶段算法

5.自动开始最少步算法

如果没有魔方被选中，开启这个选项会从第一个魔方开始自动运行二阶段算法或是最少步算法。在**选项——二阶段算法设置**中设置的数值会影响到算法的表现。

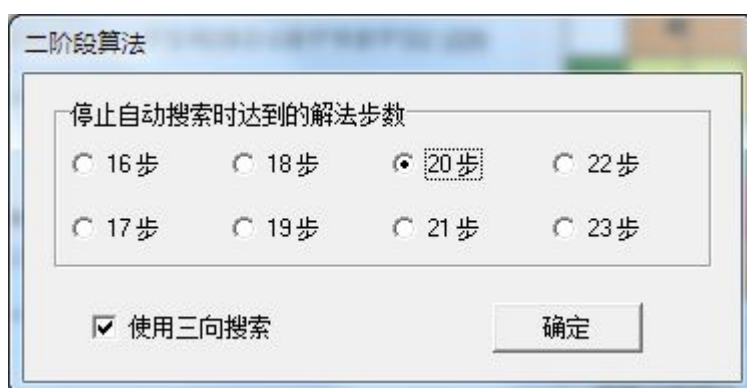
选项菜单



1. 允许中层转动

这一特色功能非常强大，它可以用来寻找包含中层转动的简短解法。使用这一模式需要重新计算一个庞大的表格文件。

2. 二阶段算法设置



二阶段算法能够计算出一组步数逐渐减少的解法。在得出一个步数少于**停止自动搜索时达到的步数**的解法之前，二阶段算法不会停止运算。

在不同朝向的情况下，二阶段算法会给出不同的解法。开启**使用三向搜索**后，算法会同时计算三个不同朝向下的解法，并选取其中最短的一个。在大部分情况下，这会极大地提高算法的表现。

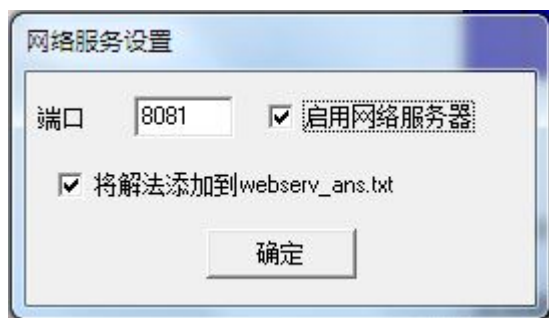
3. 无限最少步求解器



想要运行无限最少步求解器，你的电脑必须拥有 1G 的内存。任意一个魔方平均花费不到半个小时，便可找出最少步解法（使用奔腾 3 550MHz 处理器）。如果你没有足够的内存或是硬盘空间，或者主窗口内仍有魔方，下面的复选框会被禁用。

此外，如果你的电脑有着 3GB 以上的内存，操作系统为 Windows XP Professional 或其他 64 位系统，你可以使用特别版的 Cube Explorer。这个版本的运行速度是原版的 15 倍。
注：本窗口翻译见附录。

4. 网络服务



只有当你想要使用 Cube Explorer 来驱动其它程序或机器人时，你才会用到这一功能。选择一个服务器支持的端口并启动它。这一功能的具体说明请参看这里。为了安全，当你不使用网络服务器时，请关闭这一功能。

5. 立即释放线程

通常你可以暂停一个魔方的求解，并在之后继续进行计算，因为计算的线程只是被暂停了而没有被删除。如果你同时对成千上百个魔方求解，那么成千上百个线程可能会引发故障。遇到这种情况，你可以立即释放线程。缺点是释放后计算将重新开始。

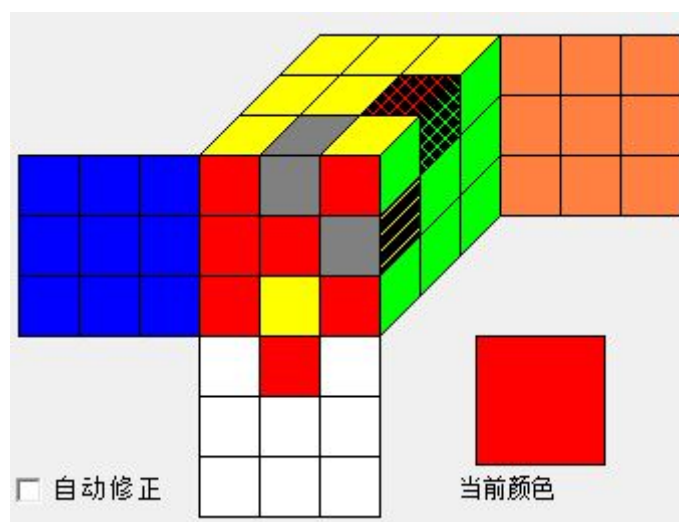
6.加载文件时跳过同构魔方	当你从文件向主窗口加载魔方时，可以忽略那些主窗口中已经存在了的及其及其同构魔方。
7.同构魔方包括倒置魔方	你可以将倒置的魔方算入同构魔方的范围。这项功能将影响所有将魔方从文件向主窗口添加的操作，比如图案编辑器和对称编辑器。
8.求解时考虑中心块朝向	<p>如果在中心色片上有数字（例如数独魔方）或是图片，那么这时魔方中心块的朝向就非常重要。这将魔方的状态数提高了 2048 倍。如果想要使二阶段算法按正常速度发挥，算法需要在第一次运行时生成一个庞大的剪枝表。</p> <p>想要使用这一功能，你必须要有 1GB 的内存。</p> <p>提示：在许多情况下，如果考虑中心块朝向，最少步求解通常十分缓慢。所以，无限最少步求解器对解决这一类问题没有提速效果。</p>
9.编辑中心块朝向	<p>会弹出一个对话框，供你对中心块朝向进行调整。如果中心块朝向是无效的，OK 按钮将显示为不可用。</p>
10.生成统计表	将会弹出一个独立窗口，显示主窗口中的魔方的解法步数、解出的最少步解法以及反对称魔方数。



求解不完整魔方

从 Cube Explorer 4.0 版开始，程序可以对不完整的魔方进行求解。在外观编辑器中，色片颜色为灰色，代表它的着色尚未被定义。你可以使用鼠标右键来取消一个色片的着色。如果一个棱块的两个色片或是一个角块的三个色片的着色都没有被定义，那么它们在求解后的魔方中可以处在任意位置，拥有任意色向。使用鼠标左键并配合 **Ctrl** 和 **Shift** 键，可以对色片进行另外两种设定，具体见下文。

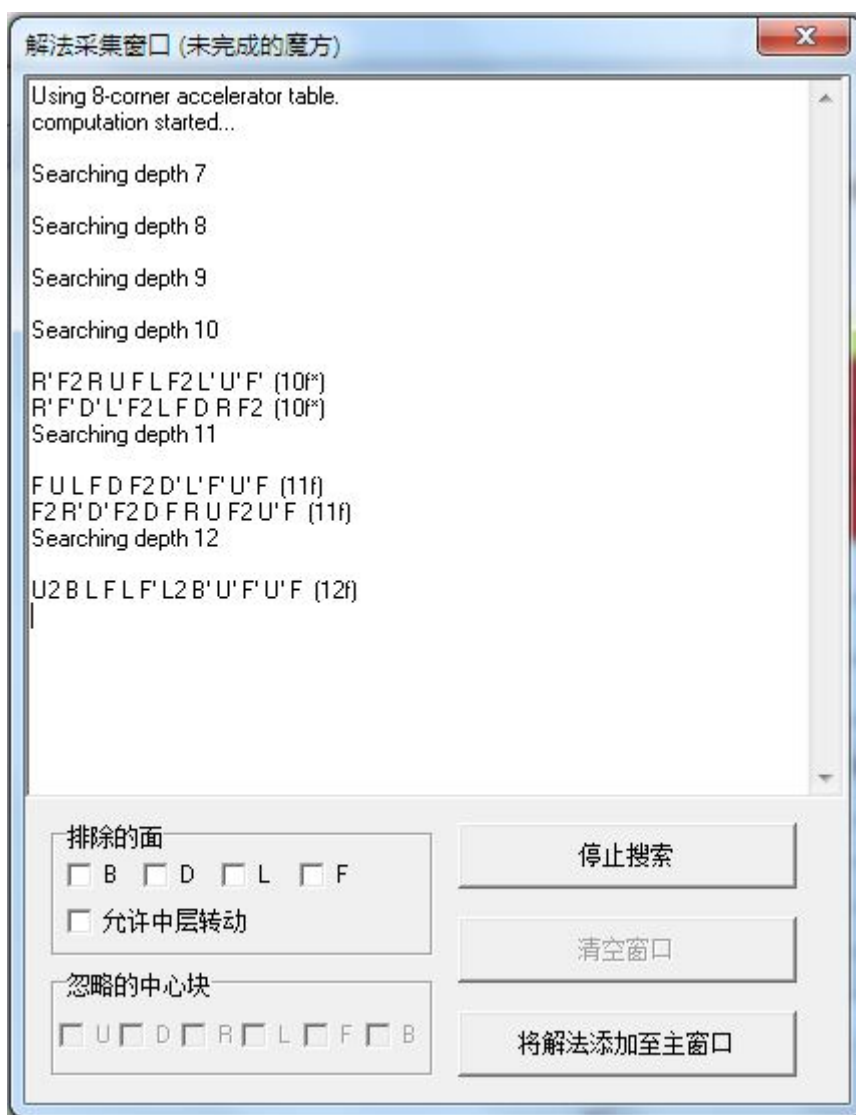
如果你对一个不完整的魔方进行求解，通常会出现一个独立的**解法采集窗口**，而且解法通常有很多个。你可以编辑，并且复制、粘贴你自己选择的解法。对于每一个棱块或是角块的颜色，总共可以有 4 种不同的设定。如下例所示：



1. 当前魔方的 FD 棱块具有规则的黄色和红色。在还原之后，这个棱块会处在 UF 位置，并且具有正确的色向（黄上，红前）。
2. 当前魔方的 UF 棱块两个色片都是灰色的。在还原之后，这个棱块可以处在任意位置，拥有任意色向。通过这一功能，你可以在求解中忽略一些色块。
3. 当前魔方的 UR 棱块两个色片分别是红色阴影和绿色阴影。在还原之后，这个棱块会处在 FR 位置，但是色向是任意的。按住 **Shift** 键再点击鼠标左键，可以使用阴影对色片着色。
4. 当前魔方的 FR 棱块拥有一个灰色的色片和一个具有黄色斜线的色片。在还

原之后，该棱块的色向由具有斜线的色片决定，即黄色片朝上，但是棱块的位置是任意的。按住 **Ctrl** 键再点击鼠标左键，可以使用斜线对色片着色。当你想要搜索 OLL 公式（速拧用公式）时，这一功能是非常有用的。

你可以参考最下方的几个实例。



如果你点击**搜索还原解法**，程序会在一个独立窗口中列出可能的解法。这个窗口有一个**将解法添加至主窗口**按钮。你会发现主窗口的所有魔方都满足你在外观编辑器中对不完整魔方的设定。

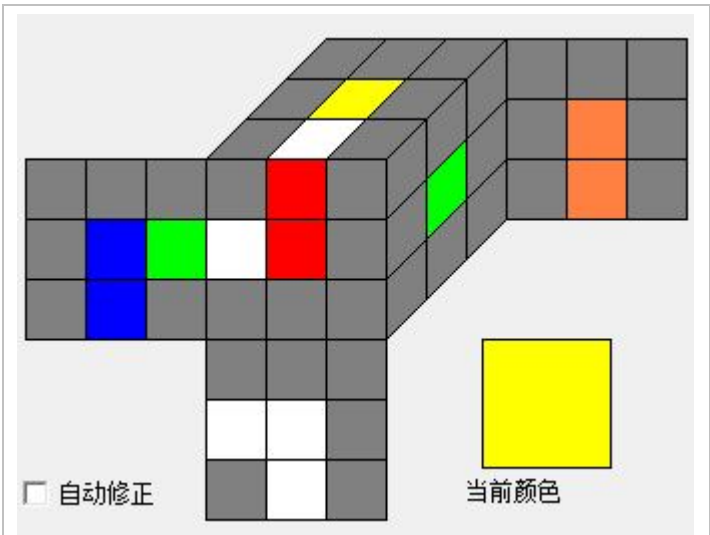
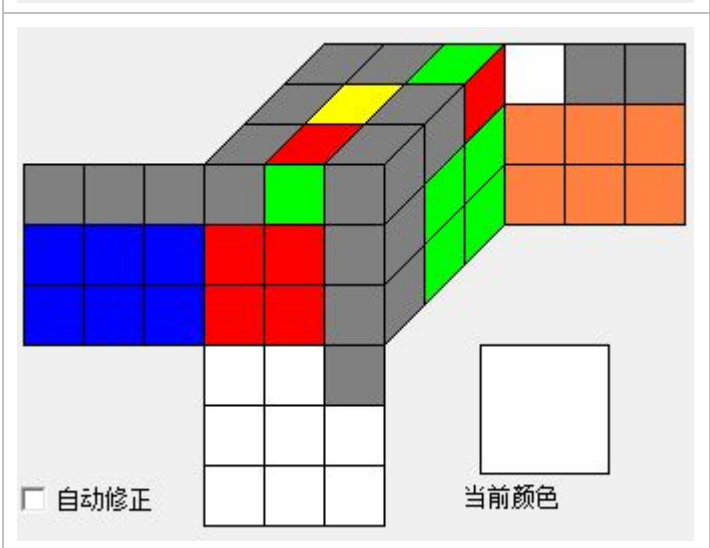
求解不完整魔方所用的算法，比求解完整魔方所使用的二阶段算法慢很多。无论如何，对于寻找步数不超过 15 步（取决于位置）的解法还是十分有效的。

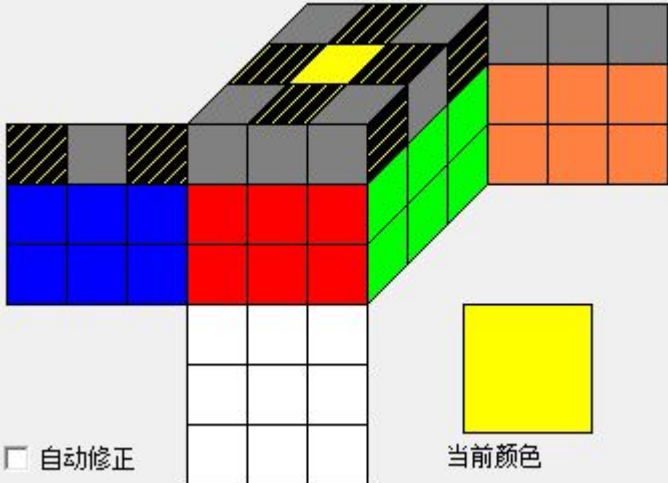
如果你不想旋转魔方的某一个面，你可以在解法采集窗口中对它进行排除。要注意的是，如果你排除了过多的面，你可能一个解法也找不到。

你可以开启**允许中层转动**，这样程序就会计算那些包含中层转动的解法。这一功能的开闭与**选项——允许中层转动**（对像为主窗口中的魔方）这一操作是相互独立的。

在 Cube Explorer 的标准模式下，**忽略朝向的中心块**这一功能是不可用的。如果你已经开启了**选项——求解时考虑中心块朝向**，借助这一功能，你可以选择在求解中忽略某些中心块的朝向。

使用实例：

	搜索十字解法
	搜索 F2L 解法，将棱块和角块插入空槽



☐ 自动修正
当前颜色

搜索 OLL 解法（调整U层色向）

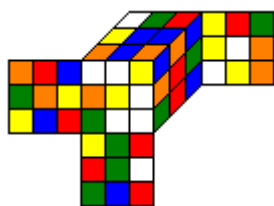
Cube Explorer 接口

你是否想要制造一台使用 Cube Explorer 来搜索解法的解魔方机器人？我决定开发一个使用自选端口的网络服务。

案例一：你只希望 Cube Explorer 替你计算解法

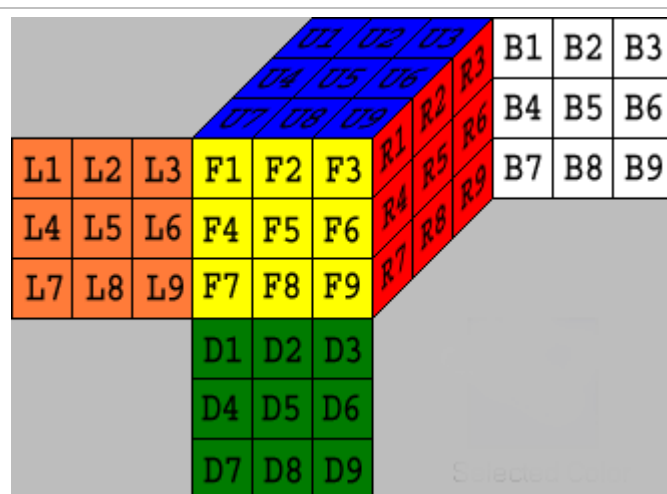
例如，如果你通过选项——网络服务设置，在 8081 端口上开启了网络服务，你可以向这个端口传输一个经过加密的包含魔方 54 个色片位置信息的字符串。你可以通过自己编写的软件或是一般的浏览器进行字符串的传输。下面就是一个使用 8081 端口向本地传输的例子：

<http://127.0.0.1:8081/?bdrfuululululrddrubbflfbdbbfdrdbdurlrudlffurfrdfblbfl>



这就是上方编码所表示的魔方。

在“？”之前的内容都会被程序忽略。在上图所示的例子中，由于 U 面的中心块是蓝色的，因此所有的蓝色色片所在的位置都应用“u”来表示。表示颜色所用的位置顺序如下：



左边的图表示了每一个位置编号所代表的实际位置。你可以忽略图中的配色。

上面例子中的字符串以“bdr”开头，因为对于例子中的魔方，U1 位置色片的颜色是 B 面的颜色，U2 位置色片的颜色是 D 面的颜色，U3 位置色片的颜色是 R 面的颜色。这个设定过程一直持续到 B9 色片设定完成。

返回的解法的质量取决于你在**选项——二阶段算法设置**中设定的参数。

在上述例子中，程序会返回字符串：

```
U2 B U2 L2 D L' B' D' L' B2 D' B U D' F2 U' L2 U' F2 U F2 (21f)
```

在**选项——网络服务设置**中，你可以决定是否将解法记录在文件 `websrv_ans.txt` 中。这也是查看结果的另一种方法。

案例二：你还想使用网络摄像头对魔方进行扫描。

如果你的机器人没有眼睛，就让 Cube Explorer 来完成这一任务吧。

你可以通过向网络服务器传输字符来控制摄像头扫描的进程。

首先，你要设置好网络摄像头的所有参数，使它在**交互式**模式下能够可靠地工作。

现在，你和你的机器人可以将魔方的 B 面对准摄像头（通常会手动操作），然后向服务器发送字符串 `"http://127.0.0.1:8081/?scanB"`，摄像头就会对 B 面进行扫描。然后对准 L 面，发送 `"http://127.0.0.1:8081/?scanL"`。以此类推，使用字符串 `"?scanF"`，`"?scanR"`，`"?scanU"`和 `"?scanD"`进行控制。在每一种情况下，网络服务器都会反馈一个 `"Done!"`。

想要对魔方求解，你需要发送字符串：`"http://127.0.0.1:8081/?transfer"`，而不是单击**还原被扫描魔方**按钮。如果所有的色片都已完成扫描，该魔方将被移至主窗口求解。这时，网络服务器会反馈一个 `"Done!"`，而不是计算出来的解法。

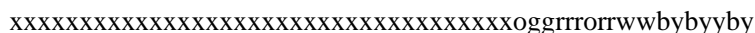
最后，你可以发送字符串`"http://127.0.0.1:8081/?getLast"`来获取 Cube Explorer 主窗口中最后计算出来的解法。

如果你手动输入了魔方，并仅希望你的机器人完成转动，字符串 `"?getLast"` 是十分有用的。

字符串 `"?clea"` 会清空 Cube Explorer 的主窗口。

如果你不想在 Cube Explorer 中手动开启摄像头，你也可以使用字符串 `"?connect0"` 连接网络摄像头。如果你有多个连接到计算机的扫描设备，`"?connect1"` 会连接第二个设备，`"?connect2"` 会连接第三个设备，以此类推。

例如：



让我知道。

行交流

样可能会比将网络功能加入到自己的程序里更加简单。

- 用你的程序读取这一文件，进行想要的处理。

在接受命令时，主窗口（不是扫描预览窗口）必须被选中、前置。

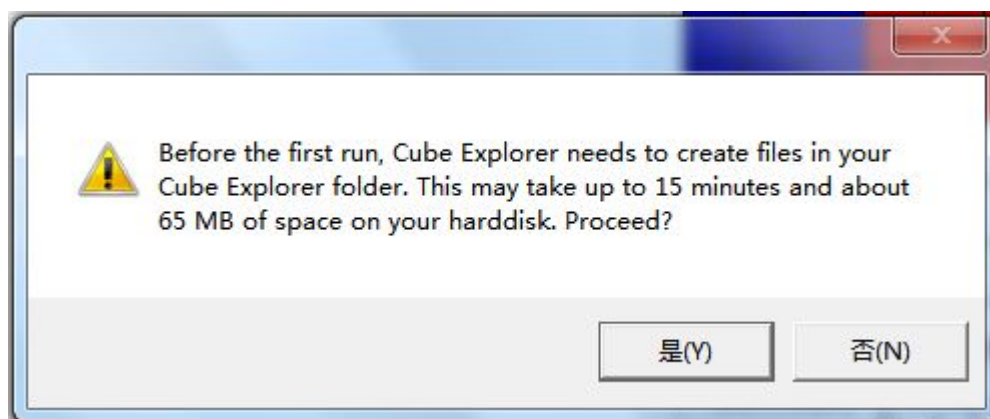
使用帮助系统

你可以按下 F1 键,或是选择主菜单中的**帮助——目录索引**来进入帮助系统。
F1 键会直接打开与当前界面相关的帮助内容。

附录

部分未汉化的界面翻译：

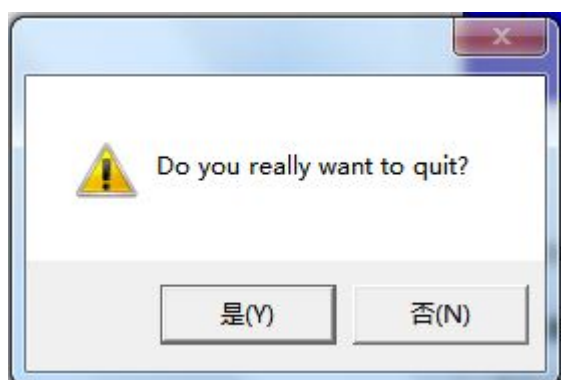
1. 第一次运行 Cube Explorer 时会弹出的窗口



译文：

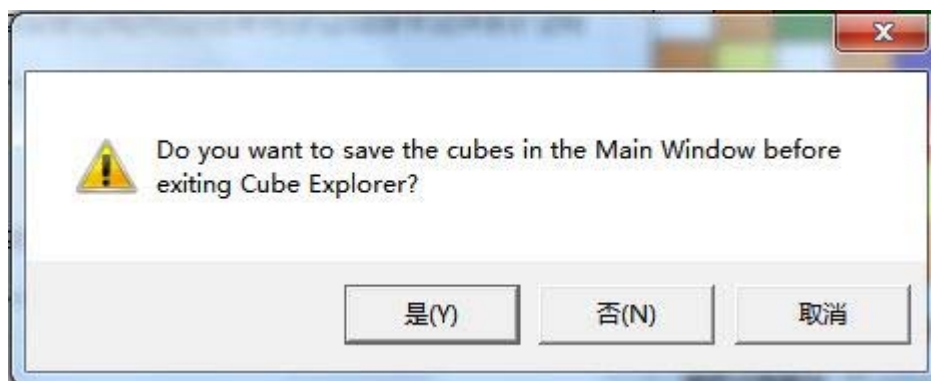
在第一次运行 Cube Explorer 之前，程序需要在你的 Cube Explorer 文件夹内生成一些文件。这可能会花费 15 分钟，并占用大约 65 MB 的硬盘空间。是否继续？

2. 退出时可能出现的窗口之一



译文：你确定要退出？

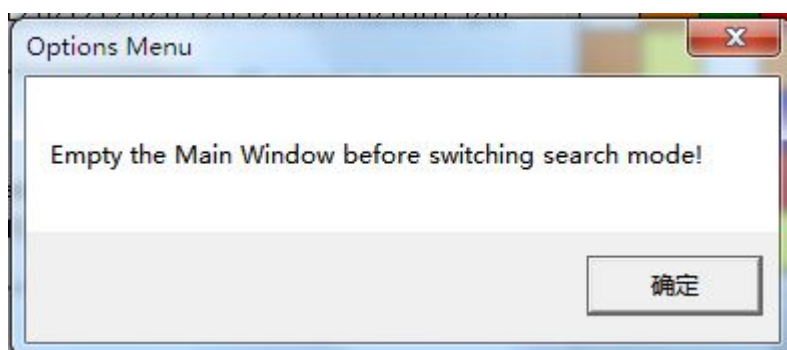
3. 退出时可能出现的窗口之二



译文:

在退出 Cube Explorer 之前是否保存主窗口中的魔方?

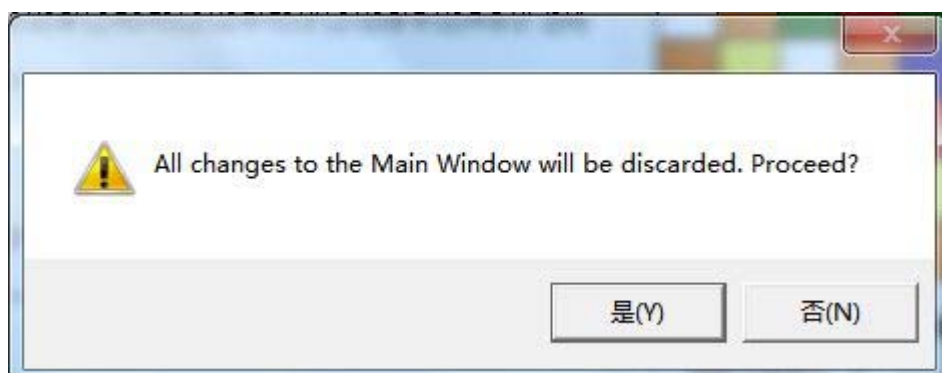
4. 在切换搜索模式时出现



译文:

在切换搜索模式前请清空主窗口!

5. 在清空主窗口时出现



译文:

主窗口的所有改动都将被放弃。是否继续?

6. 开启无限最少步求解器时出现（标准版）



译文：

想要使用无限最少步求解器，你需要：至少 1 GB 的内存；Cube Explorer 文件夹下有 673 MB 硬盘空间用来存放表格文件；清空的主窗口。

建议只在整洁的系统下（重启后）开启这一功能，否则即便是 1 GB 的内存也难以应付大型文件的交换。

与本就很快的标准最少步求解器相比，无限最少步求解器的运行速度是它的 15 倍。

7. 开启无限最少步求解器时出现（特别版）

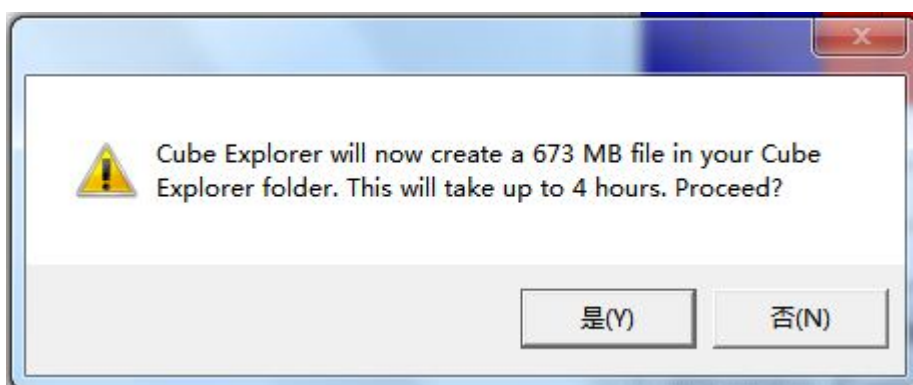


译文：

想要使用无限最少步求解器，你需要：Windows XP Professional 或 Windows Vista 操作系统。（32 位或 64 位）。如果你是用的是 32 位的操作系统，你需要开启/3GB 开关；至少 3GB 内存；Cube Explorer 文件夹下有 1883 MB 硬盘空间；清空的主窗口。

如果发生大型文件的交换，请在在整洁的系统下（重启后）运行。

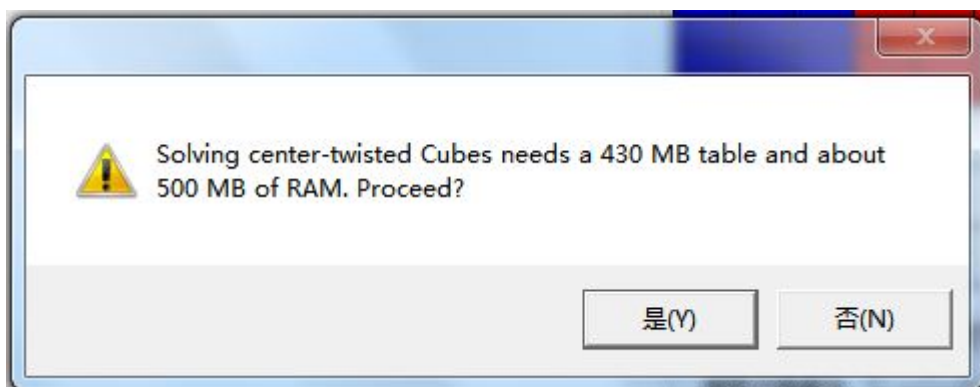
8. 初次开启无限最少步求解器中出现（标准版）



译文：

Cube Explorer 现在将会在你的程序文件夹下生成一个 673 MB 的文件。这将会花去至多 4 个小时的时间。是否继续？

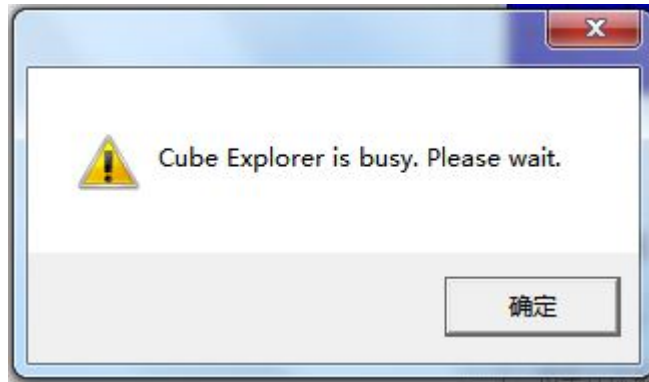
9. 初次开启求解时考虑中心块朝向功能时出现



译文：

求解时考虑中心块朝向需要使用一个 430 MB 的表格文件和 500 MB 的内存。是否继续？

10. 载入或生成文件时试图退出 Cube Explorer



译文：

Cube Explorer 正忙，请稍后。