

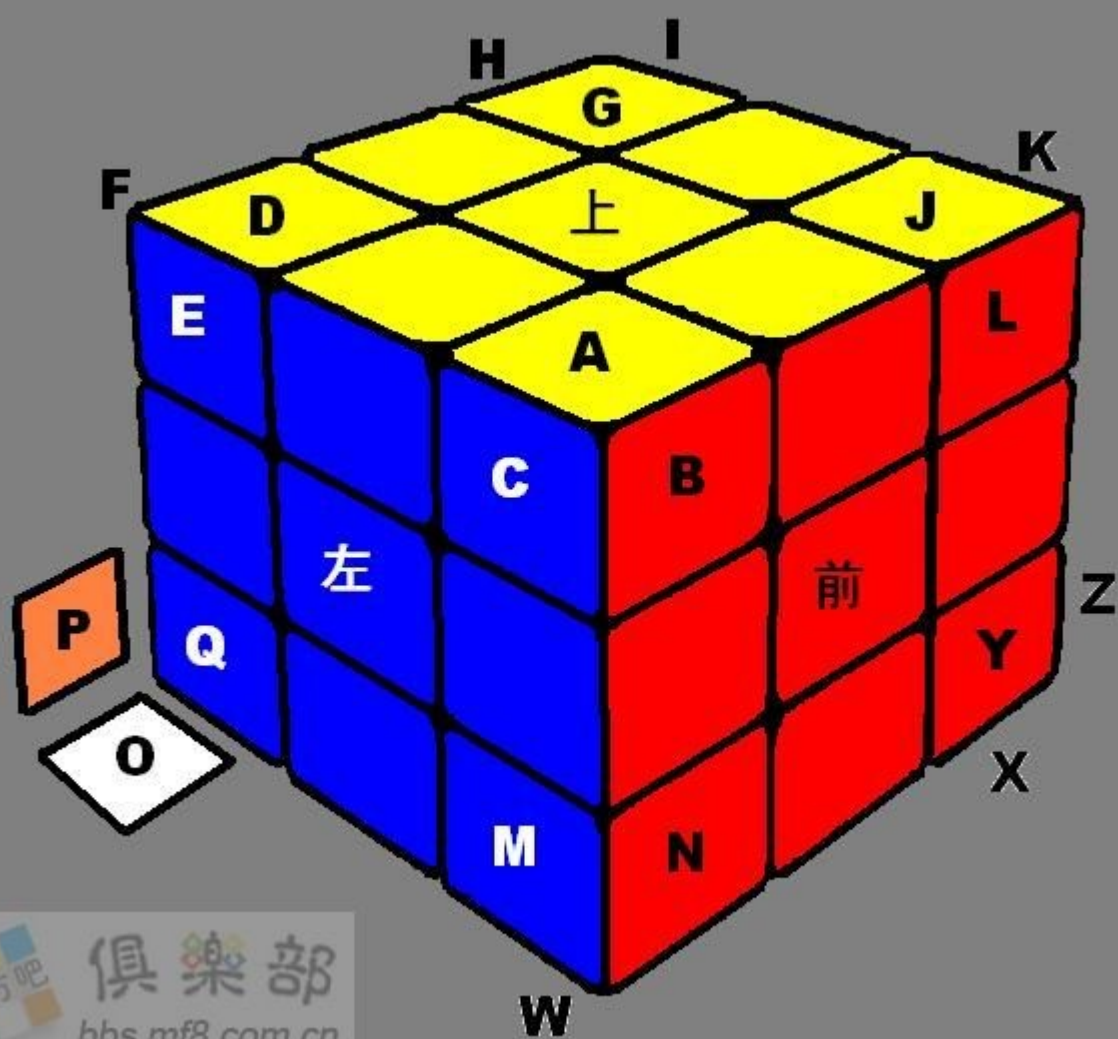
ㄣ 于法初级补充教程(转)

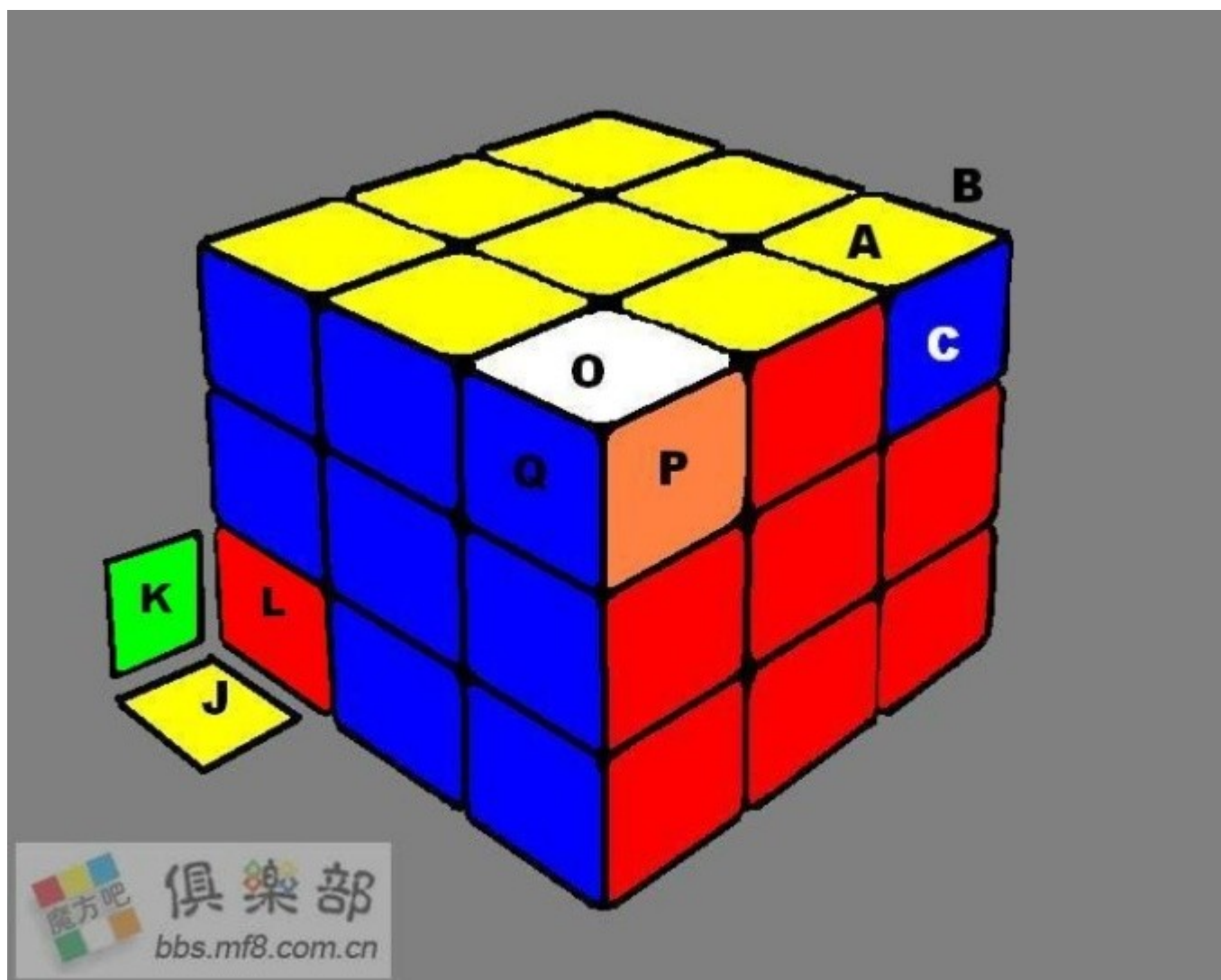
首先，我想阐述下ㄣ 于法的基本原理

ㄣ 于法是应用角块三循环和棱块三循环公式来复原的一种方法，每次解决两个角块或两个棱块的方向和位置，每次三循环公式都会涉及三个角块或三个棱块的方向及位置变化，如果说四步法是单纯块到块位置的移动的话，那么ㄣ 于法就是面到面位置的移动，或者说，是颜色到颜色的移动，学过四步法的人可能知道，四步法是先解决块 的方向，后解决位置，而ㄣ 于法是位置和方向同时解决。看到这里，很多没学过四步法的人可能糊涂了，没关系，下面我给大家举一个角块移动的例子，看了之后就明白了

看过原教程的人一定知道这个公式" $AJ \times L2' (U R U') L2' (U R' U') x'$ "，这是角块公式中的第一个公式，意思是做了这个公式之后 A 面会移动到 J 面的位置，那我现在要在这个公式的名称前面加 一个字母"**O**"，就是 **OAJ**，也就是说，做完这个公式之后，O 面到了 A 面，A 面到了 J 面，J 面回到 O 面，这样就形成了一个完整的循环。

下面 2 个图就是使用 AJ 公式前后的变化，



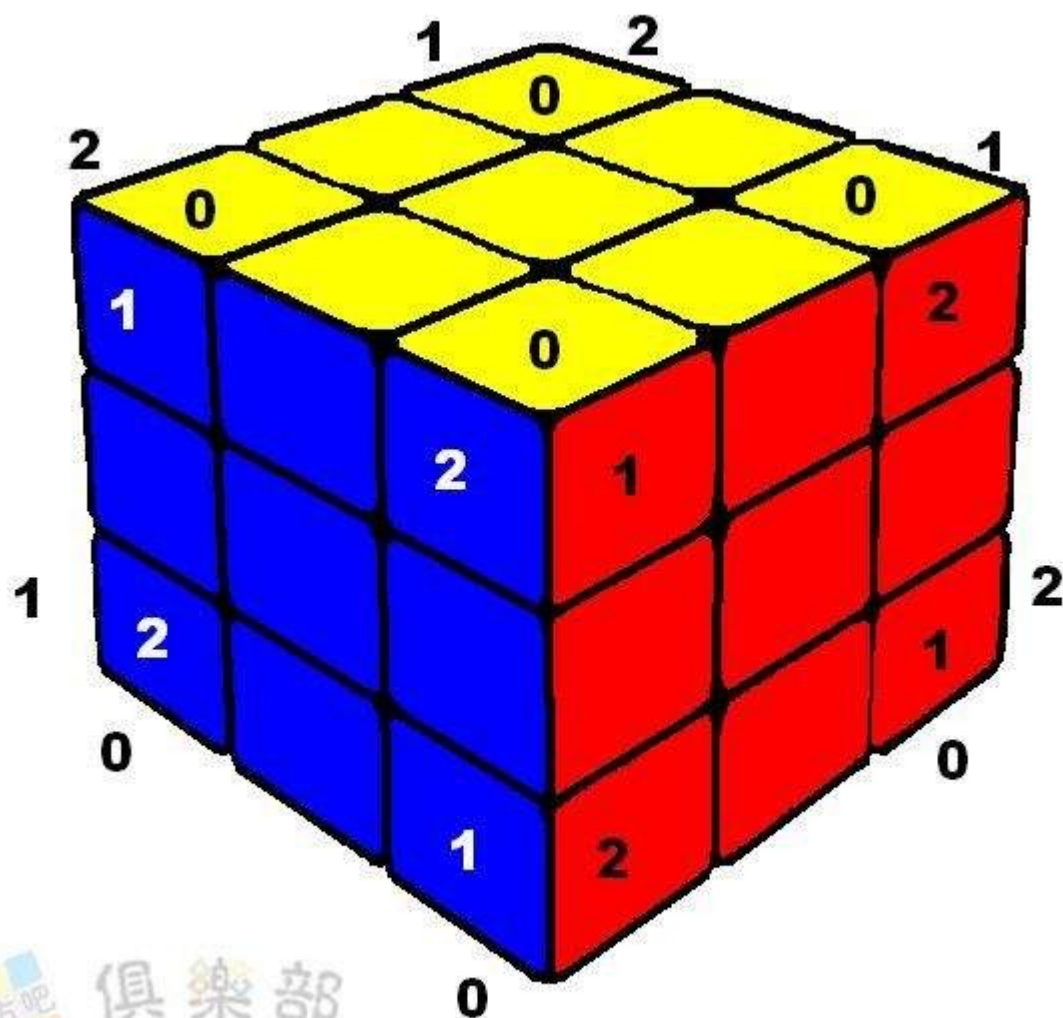


可以看出，原来O位置的白色面移动到了位置A,A位置的黄色面移动到了J位置，以此类推，J位置的黄色面移动到了O位置，这就是拧法的基本循环原理，看到这里，你应该明白我为什么说拧法是面到面位置的移动了吧？其它公式以此类推，角块公式也好，棱块公式也好，都是这个原理。

下面我们来说一下原教程的角块“0 1 2”的含义

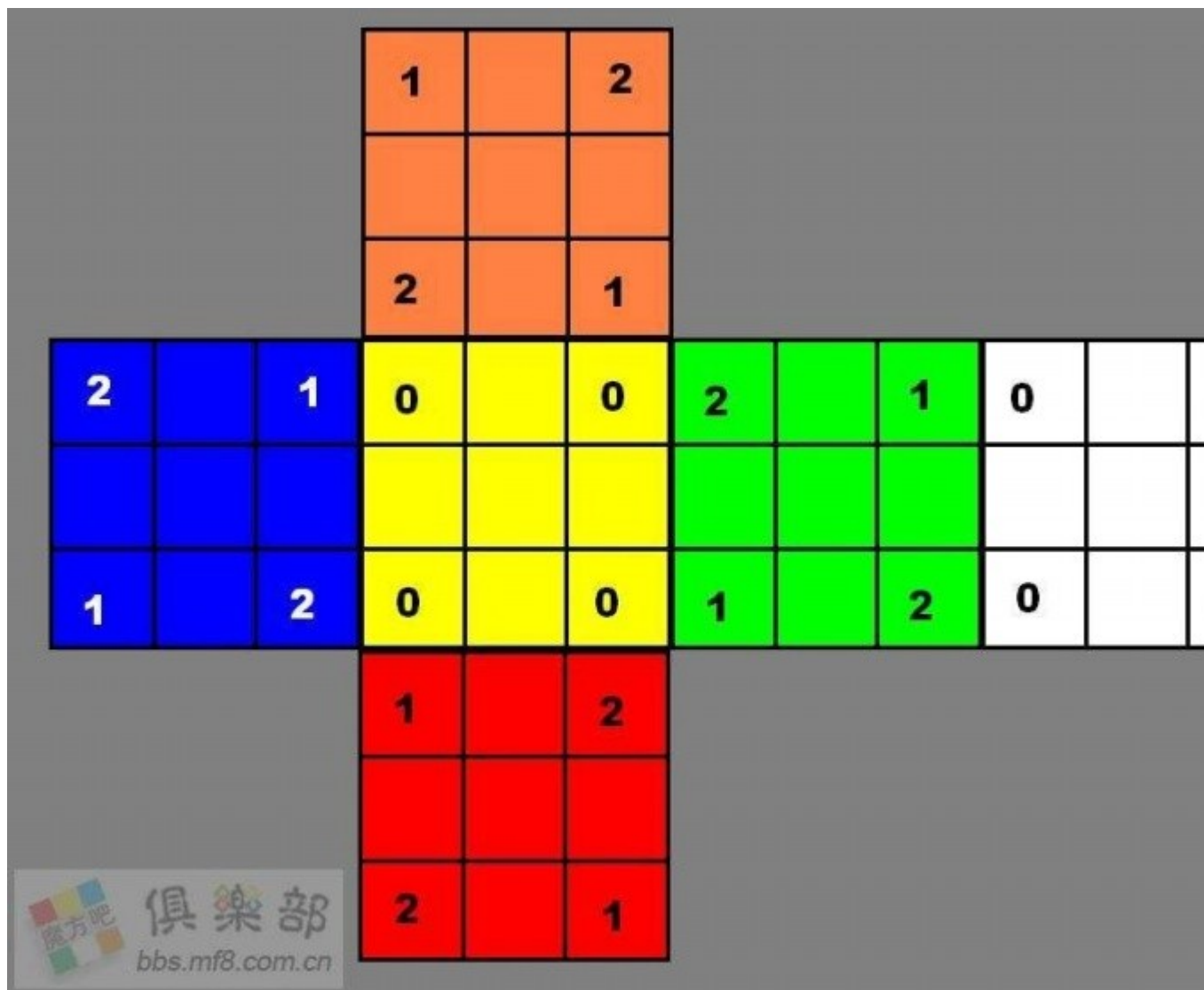
原教程的角块提出了“0 1 2”这个概念，很多人会奇怪，明明都已经用字母编码了，为什么还用数字再编一遍？我想说，“0 1 2”这三个数字的概念很重要，它揭示了角块在移动过程中方向的变化，在小循环过程中起到很大作用，在最后需要翻角的过程中也能用到，可以预知最后缓冲块归位后的方向变化。原教程中只给

出了 ULF URF 这两个角块的 “0 1 2”位置，其实，8 个角块都有 “0 1 2”的位置，下面我给出了两张直观图



俱樂部

bbs.mf8.com.cn



看图可知，U层和D层的“0 1 2”循环位置是正好相反的，也可以理解为U层做一次X2的移动之后正好成为D层的“0 1 2”循环位置。

下面我们来看一下“0 1 2”的应用实例，因为我不会弄Java，所以辛苦大家，只能看图了。

打乱顺序 (M'2 U' M'2 U2 M'2 U' M'2) U2

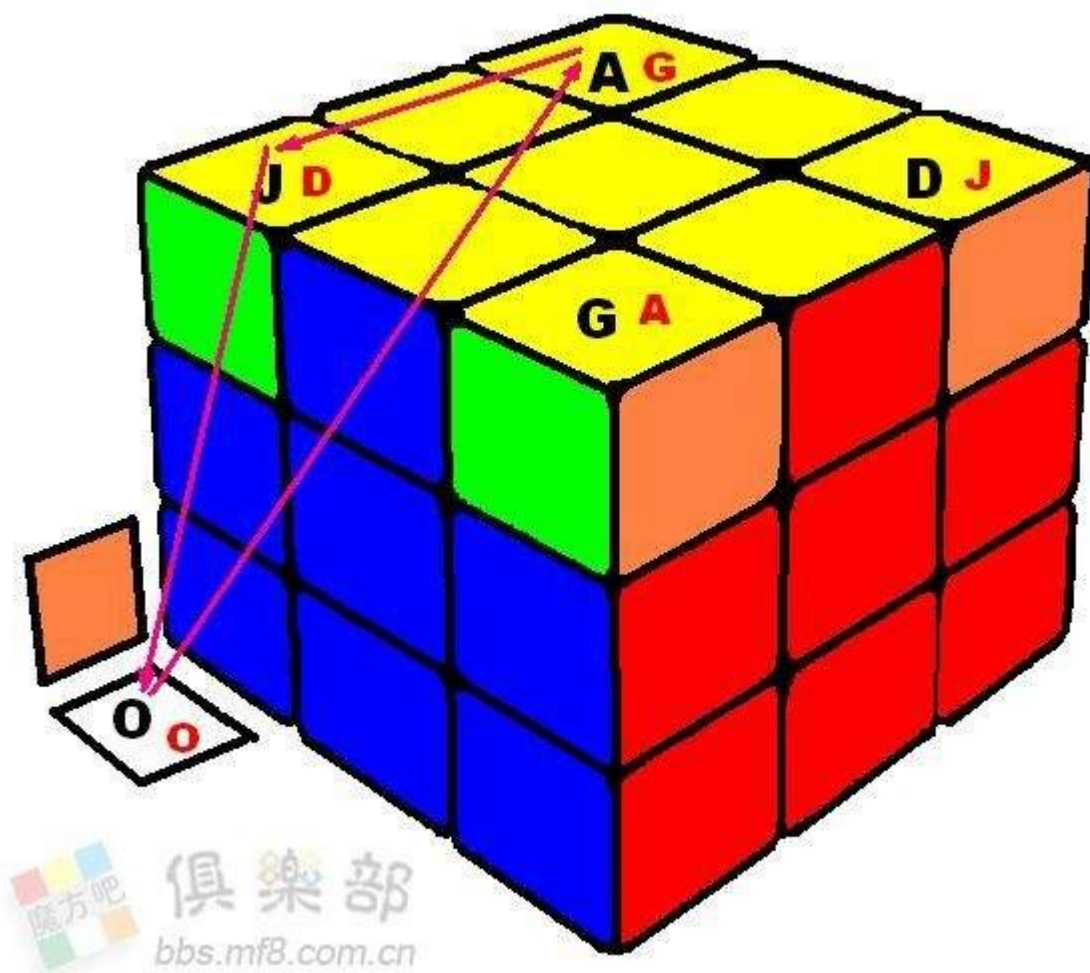
学过速拧的人可能会说，这不就是PLL21的公式加一个U2吗？

没错，经过这个公式打乱后你发现4个角块交叉互换就可以了，但为了法中我们可以用小循环解决这个问题。

DBL已经归位，而且DBL的“0 1 2”位置也都摆放正确，也就是说DBL是正确归位的，那我们就找其它的角块做缓冲块，随便找一个就可以了，比如用BUR，因为缓冲块起始位置的字母编码“O”这个位置的位置数字是0，因此下个编码也要用0，那么我们就用G，然后看G的位置是AA的位置是G，编码“GA G”，这就形成了1个小循环，还有另两个角块要互换，那就再加入到小循环中，第二次编码用第一个小循环之外没有复原的角块，可以随便选，但要记住上一个位置的小循环结尾是G，这个G的位置是“0 1 2”位置当中的0，因

此下个 编码也要选和 0 有关的，比如用 D，看 D 的位置是 J,J 的位置是 D,编码是 D J D，这就形成了第 2 个小循环，把两组编码合在一起，最后编码就是 GA GD JD

下图中给出了棱块 DBL 在这次编码复原中移动的轨迹，黑色字母是角块本身的编码，红色字母是实际复原后的位置编码。



可以看出，字母编码“O”只是在 G 和 D 的位置简单“溜达”了一圈就原封不动的回到了原位。

再来看一个例子打乱顺序 $(M'2 U' M'2 U2 M'2 U' M'2) U2 B2(R U'2 R'2 U'R2 U'R'2 U'2 R)U2 B2$

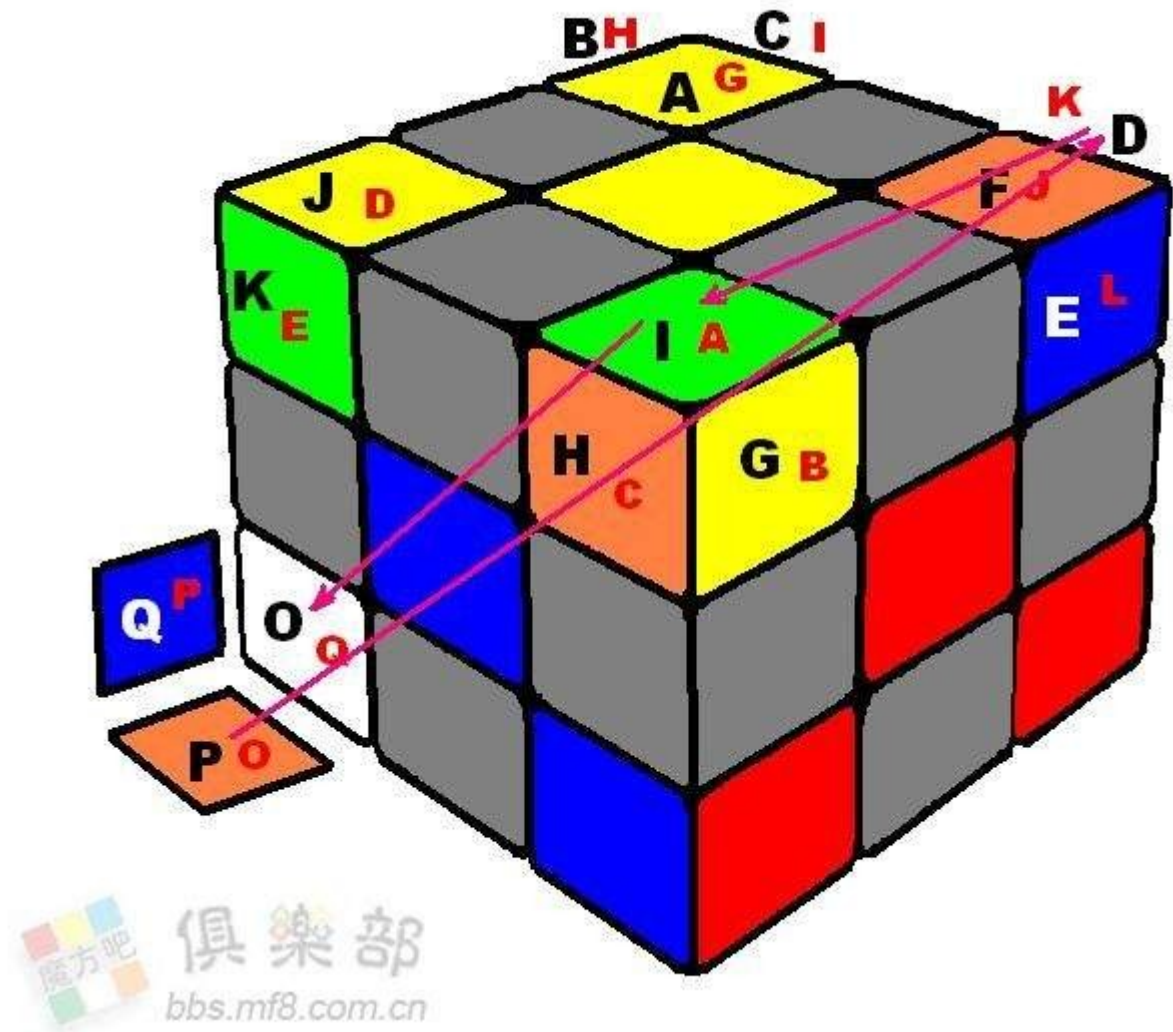
学过速拧的人一看就知道这个就是上个例子加了个 OLL 公式，首先看 DBR 上面的角块已经归位了，但是位置不正确，那我们就记住它的方向，然后看 DBL 上面的缓冲块也已经归位了，同样方向不正确，编码的起始位置还是字母 O,不同的是字母 O 位置上的色面是 P，而 P 在“0 1 2”是位置 1，因此小循环开始的时候就要选另一个角块的位置 1，可以随便选，比如选 K 吧，看 K 的位置是 D,D 的位置是 J，然后看第二个小循环，上一个循环的结尾是 J，属于位置 0 的色面，因此下一个也要用 0 的，这个也可以随便选，就选 A 吧，看 A 的位置是 I，I 的位置是 C。最后组合下编码，

就是 KD JA IC

这里要注意了，最后这个位置 C，在“0 1 2”位置中属于位置 2，也就是说最后色面 P 跑到了 DBL 块的位置

2，这样的话我们就可以判断 DBL 最终归位后的方向，和刚开始的那个 DBR 放在最后一起翻角了。

下图中同样给出了棱块 DBL 在这次编码复原中移动的轨迹，黑色字母是角块本身的编码，红色字母是实际复原后的位置编码。



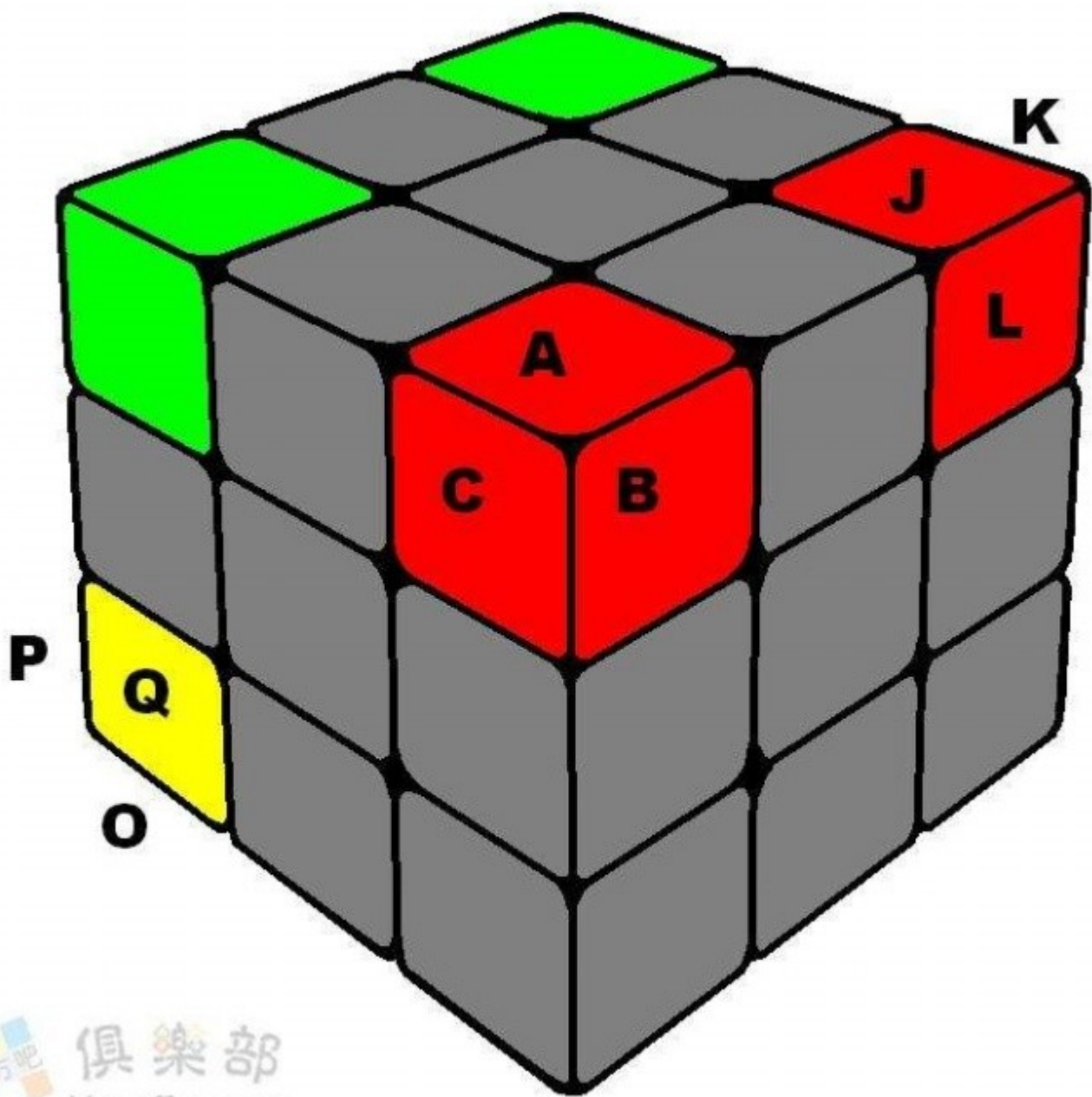
可以看出编码“P”最终回到了 DBL 的位置 2.

好，看了这两个例子你是否已经对位置“0 1 2”的作用有所了解呢？

这两个例子同时也说明在彳亍法当中使用小循环之后就不会出现四个角两两交叉或平行互换的情况，最后只能出现两个角加两个棱互换的情况另外在使用了小循环之后最后肯定有一个角块是处于 DBL 位置的。而彳亍法中的棱块同角块一样，也不会出现四棱两两互换的情况。这种特点也证明了彳亍法的奇偶校验所出现的情况相对四步法来说要少的多，也简单的多，因此学会使用小循环在彳亍法中很重要。另外想说，遇到翻角的情况建议大家还是先翻角

再来讲下关于角块 **setup** 和 **reverse** 的问题

角块的公式只给了 18 个，而这 18 个公式都以 DBL 为缓冲块（图交换场所中的黄色块位置）以 ULF 和 URF 这两个块（图中的红色块位置）的三循环公式，每次都只涉及三个角块的位置及方向变化。



但 18 个公式只涉及了这三个角块，另外的 5 个角块怎么办呢？这就涉及了 **setup** 和 **reverse** 的问题了，学过四步法盲拧的人可能知道这个，但我还是要讲一下，因为有许多人没看四步法直接杀进了盲拧，**setup** 和 **reverse** 是指把不在三循环公式“控制”之内的“块位置”依照规则转到三循环公式能够“控制”的地方，然后做相应的三循环公式，之后再让“块位置”原路返回自己原来的位置。

看到这里有的人已经蒙了（我自己也有点蒙）没关系，举个简单的例子，图中两个红色的“块位置”是交换场所，就是我刚才说的三循环公式可以“控制”的地方，任何块进了两个位置就可以任由 18 个角块的公式

“摆布”了（这句话听起来有点猥琐）

那我们现在有两个角块在绿色的“块位置”区域，想要把它们设定到红色的“块位置”区域，怎么办呢？有没有什么规则呢，规则很简单，就是在不要去动 D, B, L, 这三个层面。那么这两个绿色“块位置”经过一次 U2 就可以转到红色受控制区域了。

然后做相应角块公式，之后再做一次 U2 让两个绿色的“块位置”原路返回，就 OK 了。但是要记住在 setup 的过程中需要 setup 的一个或两个角块的编码会经由 setup 而改变，这一点在白河的教程中已经讲的很清楚了，在这里我就不重复了。

注意！！！在讲解上面这个 setup 和 reverse 的过程中我多次提到“块位置”而不是“块”，是为了说明要 setup 和 reverse 的其实是块所在的位置，而不是块本身，因为块本身在整个三循环过程中是会移动的。

关于彳亍法中的编码

彳亍法中的编码可以任意选择，就是说每个块上的颜色编码可以自己选择，字母也好，数字也好，汉字也好，偏旁笔画也好，可以随便选择，怎么喜欢怎么来，可以选择一种方便自己记忆的编码。这点无需多说，还想说的一点就是编码代表的含义，编码不仅代表了不同棱角块的色面，同时在复原过程中复原编码也代表了色面在复原过程中的移动位置。

还是以上面的例 1 为例，例 1 的编码是 **GA GD JD**

这里很多初学者可能奇怪，这里怎么出现了重复的编码？怎么同时出现了两个 G 和两个 D？

下面我就来仔细解释一下这组编码的具体含义，开篇我们说过，彳亍法是面到面位置的移动，那么这些编码的含义其实也是代表了色面在复原过程中的移动位置，首先看第一组编码的第一编码 G，这个 G 其实是为了把缓冲块换出让缓冲区域的位置腾出来，同时让小循环中的“一个角块”进入缓冲块区域来为下一次的正确归位做好准备同时这个 G 也保证了缓冲块蓝橙白在被换出时位置正确。

我们看 O 跑到了 G 的位置，就是说缓冲块被换出来了，

接下来我们要让原来 G 位置的红蓝黄角块正确归位，红蓝黄的黄色面的正确位置是哪里呢？我们看是 A, 因此第二个编码选 A.

那三循环当中每次涉及到三个块的移动，另一个块哪去了呢？很简单，跑到缓冲区域的“O”上面去了，就是我们刚才说的那个小循环中的“一个角块”，我们看，就是那个绿橙黄角块我们来回顾一下第一组编码的循环 **O-G, G-A, A-O**。

接下来我们要让绿橙黄的黄色面正确归位了，我们看黄色面的正确位置是 G, 因此第二组编码的第一个字母还选 G。

第二组编码的第二个字母 D，和第一组编码的第一个编码 G 的作用相同，让第二组小循环当中的“一个角块”进入缓冲区域为下一步的正确归位做准备。而这个准备正确归位的角块就是那个绿红黄角块。

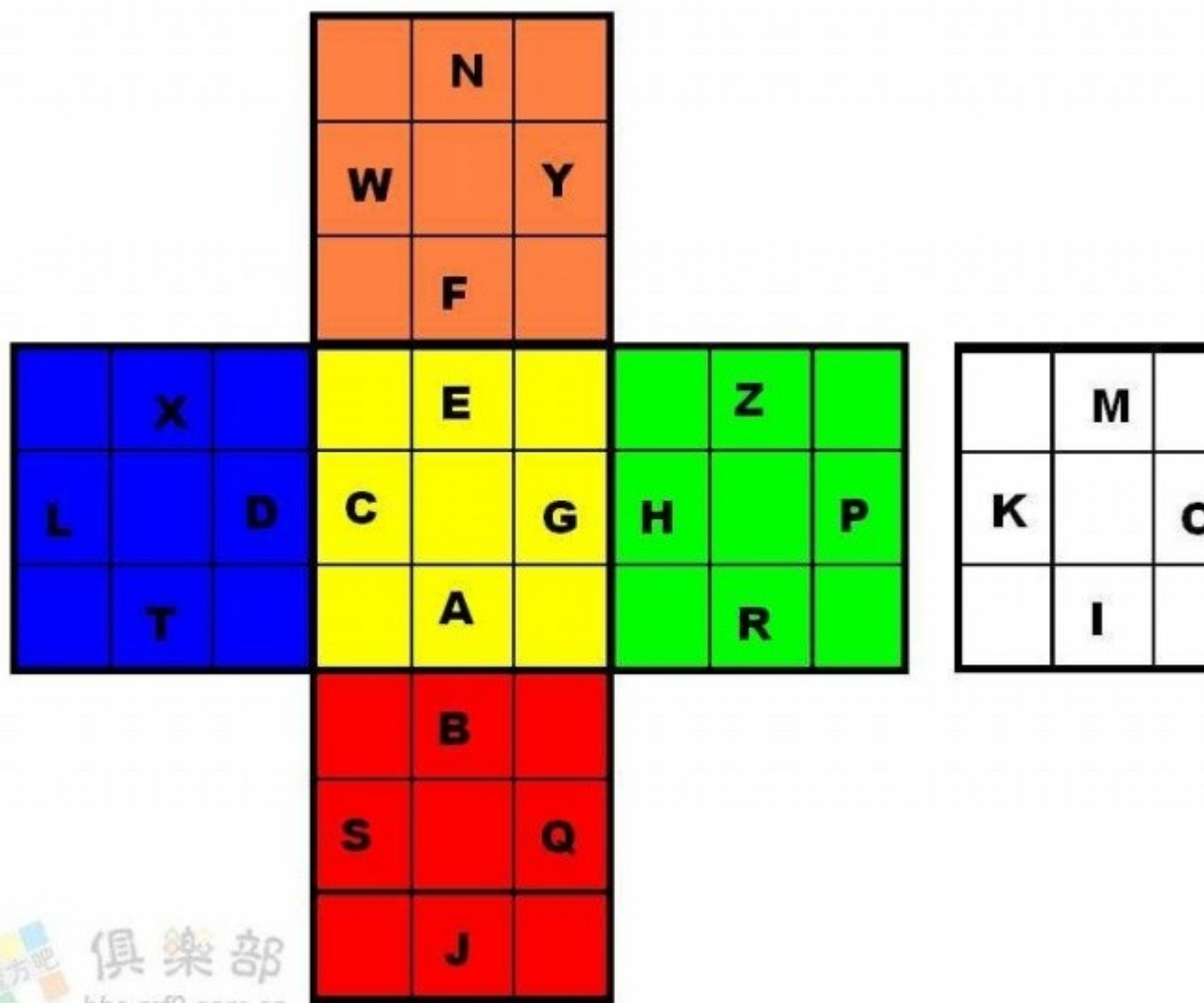
这组编码的作用也清晰了，我们来回顾下第二组编码循环 **O-G, G-D, D-O**。

接下来要让绿红黄角块归位了，我们看正确位置是 J，而 J 位置上的蓝橙黄角块的正确位置是 D，因此在编码过程中又选了一次 D。

第三组编码的循环就是 **O-J, J-D, D-O**。

看了这个例子，你是否对编码所代表的含义及作用有了一个更清晰的思路呢？

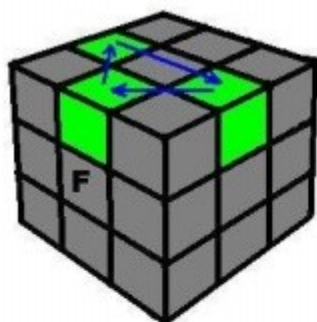
关于彳亍法中的棱块



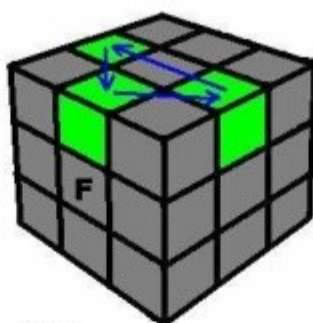
上图为棱块编码展开图，注意，白色面为自上而下的透视图。

于法当中以 UF 块为缓冲块的棱块复原法非常灵活，每个面都可以作为操作面，这一点白河教程里已经讲的很清楚，在这里就不多言了。

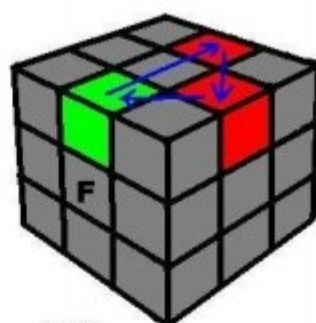
想讲的是如何运用 UF 当中的同层公式，大家可能看了同层公式之后很迷惑，什么 FH，DF 之类的到底怎么用啊？特别是那个 EDH，EGC 什么的，和 UF 缓冲块也联系不上啊？没关系，我来解释下



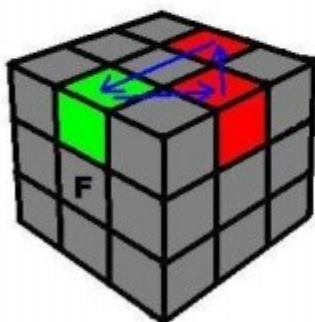
CG



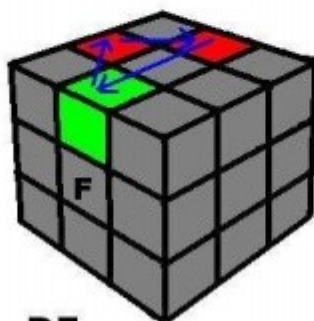
GC



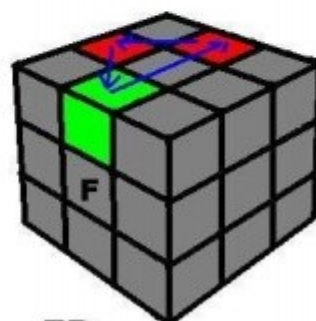
FH



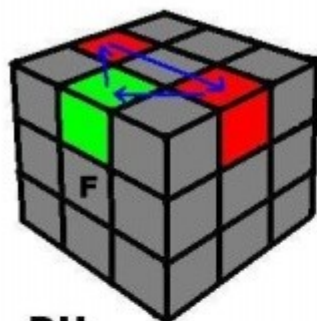
HF



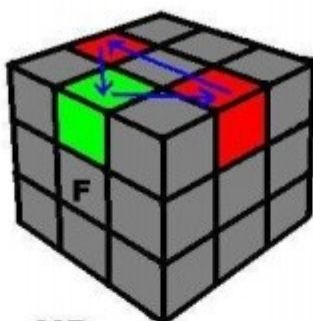
DF



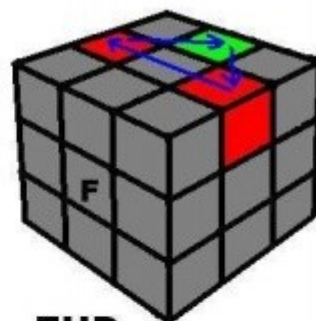
FD



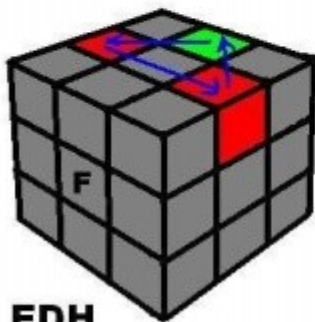
DH



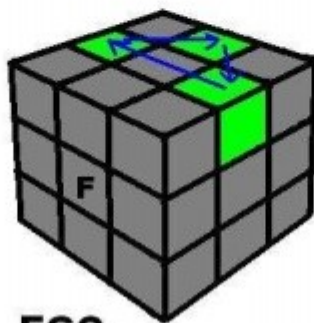
HD



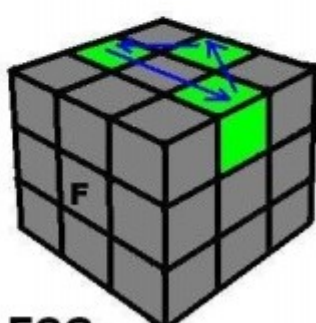
EHD



EDH



EGC



ECG

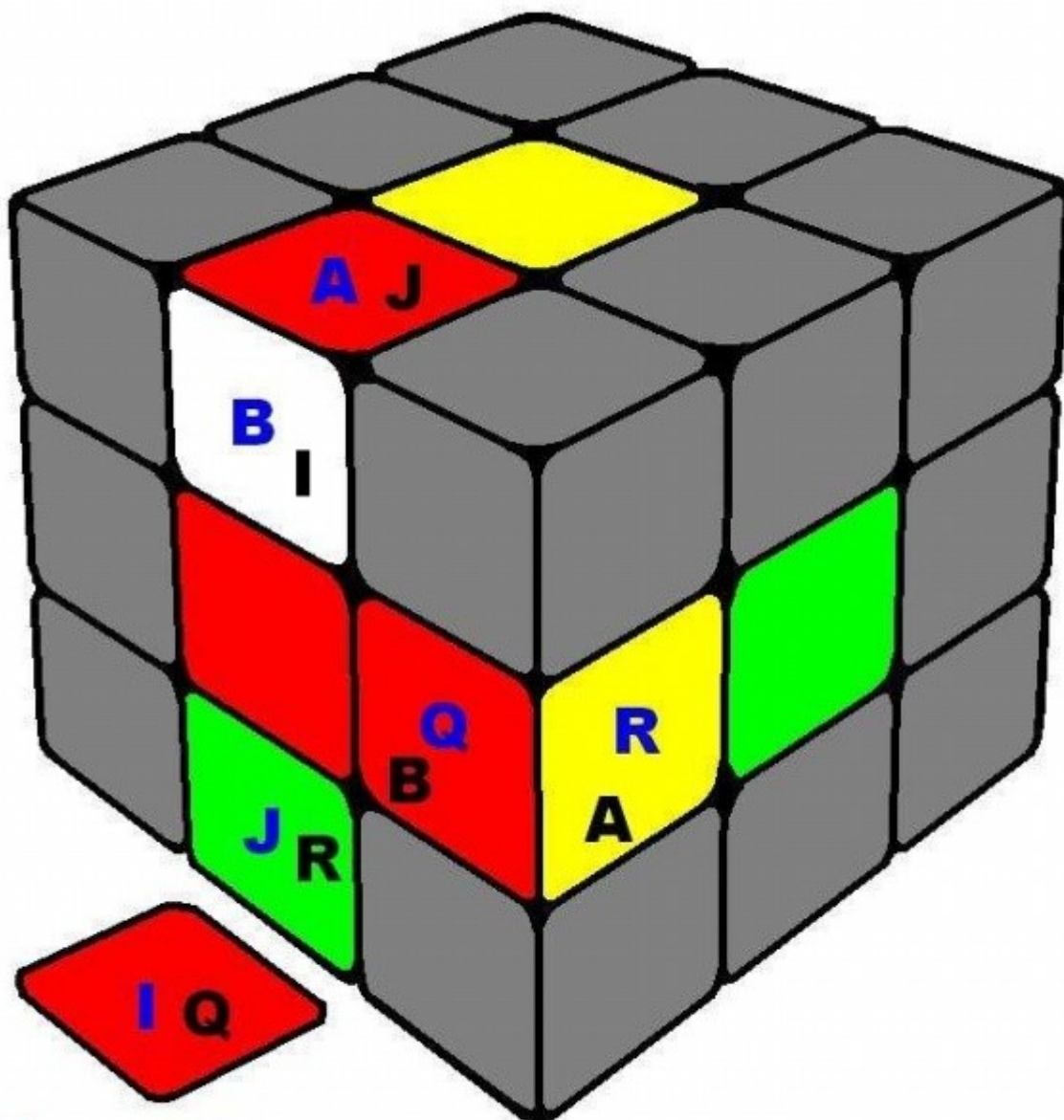
看图 E1,E2, 图中你会看到绿色块和红色块, 以及蓝色箭头表示的方向, 有人会想, “哦, 绿色块是缓冲块, 红色块是受公式“控制”的区域”, 注意!!!! 这样想你就错了, 大大的错了! 给你来个当头棒喝。因为这里不存在置换块和缓冲块一说, 任意块都可以是缓冲块的位置, 任意块也都可以是置换块的位置, 这要靠操作时的具体情况来看, 这里只存在三循环。

绿色块和红色块是为了区分棱块色面所在的不同层面。也就是说, 两个红色块的编码在同一层面, 而绿色块的编码在另一层面。

另外要注意的是, 我在图中没有标示任何面的颜色, 也就是说, 公式可以在任何一个面操作, 但需要注意的是操作方向, 我在每一张图的魔方面里面都标了一个 F, 就是说操作的时候要 F 对着自己。

还是来看一下具体实例, 看图例 3

蓝色字母为复原后的实际位置编码, 黑色编码为棱块本身的色面编码。



这个 **case** 做一次 **JR** 的编码就可以复原了，具体说就是红白棱的红色到绿红棱的绿色面位置，绿色再到红黄棱的黄色位置，这样一个循环，就是 **JRA** 这样一个循环。

但问题是如何 **setup**?

有人说，“哦，先做一次 **R**,把红黄棱块 **setup** 到 **U** 层，然后做 **D'L2**,把绿红 **setup** 到 **U** 层，然后做公式，然后 **reverse**.....等一下，我刚才怎么 **setup** 上来的？”

我想说，汗，你 **setup** 这么麻烦，不忘才怪，**setup** 和 **reverse** 一共用了 6 步，太费时间了。

现在我教你个方法，就是观察三个要交换的色面所在的位置，经过观察发现，如果已红色的中心块做顶面，**J** 面（红白棱的红色面）和 **A** 面（红黄棱得黄色面）是朝向外围的，只有 **R** 面（绿红棱的绿色面）是向上的，我们再来看刚才我给的两张图 **E1,E2**,两个红色块代表 **J** 和 **A**，一个绿色块代表 **R**

这样就可以锁定公式在 **FH** 和 **HF** 这两个范围内，然后再看一下循环方向，最后确定是 **HF**，那我们就先做一次 **x** 的整体翻转，然后做公式 **HF**，之后再做一次 **x'** 的整体翻转，整个 **setup** 和 **reverse** 只用了两步。