# VCB-Studio 教程 11 编码器参数研发方法

编码器的参数，特别是类似 x264/x265 这种，对成品画质的影响是很大的，哪怕同样码率下，不同的设置可能有极大的区别。而且编码器参数的选取，对于不同类型的片源，又有不同的侧重点。一般来说参数的选取有三种选择：信官方，信高人，自己测。本篇教程讲述如何系统性的研发测试编码器的参数。

## 1. 参数分类

一般来说，编码器都有 document 来描述有哪些参数供你设置，这些参数大概是做什么的。测试的第一步就是先阅读 doc，根据你的经验，把参数分为这三类:

编码规范/specification, 这类参数一般是规定编码一些格式规范、编码器工作的。比如 x264/x265 中--profile --level --matrix, --display-window --sar 等等。这些参数一般无需测试，该怎样就怎样。一般这些参数的调整也不会显著影响编码速度和编码画质。

取舍性/trade off, 这类参数一般是时间换画质的，比如 x264/x265 中 --ref --bframes --me --subme --merange --rect --amp 等。这些参数对画质的影响往往是通用性的，不随片源类型、码率高低变化太多。

码率控制/rate control, 这类参数决定码率的分配，分配的多少，怎么个分配法。比如 x264/x265 中 --crf --qcomp --aq --psy 等等。这些参数对画质的影响往往体现在目视效果上，跑分并不能很好的体现，且不同类型片源、码率表现很不一样。

参数组合的效果，随着参数数量的增加，是指数级别的。所以参数测试第一步是选取测试价值最大的。历史经验表明，这类参数一般在 rate control 分类中，rate control 类型的参数可以在不增加编码运算量前提下极大地优化码率分配和目视效果。进一步你需要找出哪些参数是重中之重，一般来说无非这几类：

1. --crf/-q, 最直接影响画质和码率的；
2. --qcomp/mbtree，影响码率的时间分配
3. --aq/psy，影响码率的空间分配

其他一些影响较少的包括 --scenecut/--b-bias/--ipratio/--pbratio 这种 ipb 帧设定的，--sao/--deblock 这种内置 filter 的。甚至在 HEVC 上，过大的 block size 都可能影响画质（x265 中的--ctu --max-tu-size）

其他的参数，我们一般选择默认，或者用官方推荐的。比如说 tradeoff 相关的，我们就一般采用官方给定的--preset。为了保证 tradeoff 相关的那些不会对画质构成瓶颈，我们一般基于接近官方的--preset slower/veryslow 来进行测试。

## 2. 片源选择

下一步骤是选择一个具有代表性的片源。一般来说选源遵循以下几个原则：

(1). 和你需要测试的画风一致。如果你要测试重噪点电影，那就剪一段重噪点电影；如果你是测试一般性新番动漫蓝光，那么就选一段各方面均衡，噪点适中，亮场暗场都有，静态动态均衡的新番蓝光。

(2). 你可以用 avs/vs 对片源做预处理，来使得它更符合要求；只不过如果你的处理比较复杂，测试的规模又大，建议先压一个 YUV 无损视频。

(3). 片源的长度适中。太长则压制时间长，需要的电量增加；太短有两个缺点：第一个场景少而集中，测试的普适性不高；第二个难以估算码率。一般片源我们选择半集到一集的长度。

## 3. 测试所用遗传算法简介

我们假设你选定的片源是环境（草原，湖泊，森林……）

编码器编码出来的视频是一个种群（咸鱼，蛤蟆，猫娘……）

每一个编码器参数是一段基因，编码器参数组合就是基因汇总，决定了种群里每一个单体的性状。

物竞天择，适者生存，这个“天”，就是来评价成品好坏的标准，可以是客观跑分(psnr/ssim)，也可以是你的眼睛。

测试参数的过程，就是让种群在环境中进化的过程，让基因自由的变异，然后挑选出这一轮变异最成功的个体存活，存活的个体进一步进化。

如何理解呢？假如说我们已经选择了片源（环境），准备测试 x265，现在一开始我们有个个体：

--crf 28 --psy-rd 0.3 --aq-strength 0.3 --qcomp 0.4

可以想象，这个参数编码出来的视频体积很小，画质很差，表现为个体很弱小。我们现在让它进化。它有 4 个方向可以进化：

降低 crf，比如--crf 27

提高 psy-rd，比如--psy-rd 0.8

提高 aq，比如--aq-strength 0.6

提高 qcomp，比如--qcomp 0.5

单独设置这四个参数，画质和体积相比进化前都会有提升，那么哪一种进化方向最有效呢？这时候我们就可以先调节参数改变的幅度，使得 4 个成品的体积很接近，然后进行比较。如果比较下来，提高 psy 是最有效的，那么我们就可以保留--crf 28 --psy-rd 0.8 --aq-strength 0.3 --qcomp 0.4，并以此为基础，进行下一轮测试。如果下一轮测试表明提高 aq 最有效，那么我们保留--crf 28 --psy-rd 0.8 --aq-strength 0.6 --qcomp 0.4 再继续……

直到这个种群中存活个体足够强壮，那么参数也就具备实用意义了。

## 4. 测试步骤和各种操作规范

我们以 vcb-s 测试 x265 为例

1. 先选定一个合适的 sample。之前前两轮用的是 LL S2 的半集，可以说亮场暗场、动态静态、锐利度、噪点强度都比较具有代表性，长度也合适；

2. 选定一个适合的 tradoff 设置。我们用的是之前对着 preset 改的，大概相当于--preset slower

3. 选定一系列主要测试的参数，包括--crf --aq-strength --psy --qcomp，以及其他次要参数，比如--sao --cu-tree --rdoq-level --ip-ratio/--pb-ratio --sao 等。

4. 创造一个“原始”的个体，各参数偏向于小体积渣画质的。

5. 让当前个体分别提高主要测试的参数（画质/码率意义上的提高，非数值上，比如 crf 实际就是应该降低），小心的调节参数幅度，使得新成品之间体积差距很小（一般 最大/最小<100%+K，K 取值 3%左右比较合适）。新成品比起上一轮成品的体积提升幅度在 3*K 左右（比如每一轮提升 10%）。K 取值越小，测试的精确度越高，一路提升码率的过程中，覆盖的码率段越精细，但是测试量增加：一方面，达到同样的体积，需要更多轮的测试；另一方面，每一轮因为体积过大过小必须重新微调参数的可能性增加。

6. 有些参数可能需要同时设置才能发挥作用，比如--psy-rd+psy-rdoq。

7. 你可以增加一些对比样本，比如相近码率段的 avc 样本。

8. 比较这一轮成品，判断优劣。一般判断优劣的方法是，选取 20 个左右的随机采样点，然后用 avspmod 反复切换看，每个场景优劣顺序，然后在 excel 表格中记录，并在最后汇总。比如这是某一轮的测试结果:

| | | | | | |
|---|---|---|---|---|---|
| 314 | avc | qcomp | rd | aq | crf |
| 522 | qcomp | aq | avc | crf | rd |
| 1059 | avc | crf | rd | qcomp | aq |
| 3522 | avc | rd | aq | crf | qcomp |
| 5697 | crf | aq | rd | avc | qcomp |
| 7697 | qcomp | crf | aq | avc | rd |
| 8030 | avc | qcomp | crf | rd | aq |
| 8625 | avc | qcomp | crf | rd | aq |
| 10581 | crf | qcomp | rd | aq | avc |
| 11755 | avc | qcomp | rd | aq | crf |
| 7363 | avc | crf | qcomp | aq | rd |
| 7885 | qcomp | crf | rd | aq | avc |
| 8088 | rd | qcomp | avc | crf | aq |
| | | | | | |
| avc | 31 | | | | |
| qcomp | 32 | | | | |
| crf | 38 | | | | |
| rd | 44 | | | | |
| aq | 50 | | | | |

说 314 帧，avc 的对比样本画质最好，提高 qcomp 次之，然后是 psy-rd+rdoq，aq，最后是降低 crf。最后，每一种方案，算一下名次总和。总和最低的可以认为是最好。比如说这次测试，avc 样品微弱优势最高，然后参数改变中，提高 qcomp 的方案总体来说最好，已经几乎可以与 avc 样本不相上下。

9. 每一帧比较中，碰到无法区分优劣的这种简直是家常便饭，这种一般可以删掉这个采样点，选取一个新的随机的。但是更多时候只能是硬着头皮放大图像比，所以每一轮比较其实都对眼睛是一种折磨。每一轮比较可能会出现两个甚至多个方案都好的很接近，甚至得分一样。这时候可以采取的策略有：(1). 随机选取；(2). 保留稀有，选择提高之前几轮不曾提高的参数；(3). 融合保留，同时提高 2 个参数，但是只提高正常幅度的 50%，这样总体积不至于一下暴增；(4). 同时保留两个，下一轮两个样本各自测试，然后汇总比较，在更多的组合中选取优胜者

10. 从初始参数组合开始，如果哪一轮测试之后，所有的主要参数都被提升过至少一次，那么这一轮可以被认为是 critical round。初始参数可能很不理想，有些偏高，有些偏低，但是到这一轮结束，本来偏低的参数理应经过了数次提升，本来偏高的参数则应该提升次数较少，结果是没有任何一个参数很严重的过高/过低。critical round 过后的参数开始具备实用意义，因为其组合已经比较合理，在当前码率段应该具备较好的平衡性。

11. 时不时的，我们会加入一些非重要参数的改变，来看看会有什么效果。比如某一轮测试我们加入--no-cutree 的比较，某一轮测试我们试着调整--rdoq-level。如果调整的参数几乎不改变码率，我们就在某一轮测试之后，用胜出参数做基础；如果调整的参数也会改变码率，我们把它加入当前一轮的码率提升过程。其实就是非重要参数测试频率低，每个参数并非每一轮都测试，而是隔几轮测试一次。这种行为可以视为基因突变。

12. 时不时的，对于重要参数，我们会做出一些随机性的调整，来看看效果。假设我们只测试--crf 和 psy，上一轮是--crf 18 --psy 1.6:8.0，当前是--crf 18 --psy 2.0:10.0，除了测试--crf 17.5 --psy 2.0:10.0 和--crf 18 --psy-rd 2.4:12.0，我们决定看看--crf 17.0 --psy 1.6:8.0 的表现。这增加的一组大幅降低了 crf，削弱了 psy（或者可以看做，从更早的测试结果直接强制大幅变动 crf）。一种可能的情况是：从--crf 18 --psy 1.6:8.0，如果我们要增加 10%的码率，那么提升 psy 最好；但是如果要提升 20%，反而是降低 crf 好。我们一旦先提升了 psy，后续无论怎么做，当码率达到 120%的时候，都不如不提升 psy 直接降低 crf 好。这样，这个调整就可以让我们确认一下这样的可能性。这种行为看做基因重组。

13. 加入参照性的对比。这个对比可以是同样编码器，更早的测试中，测试的参数组合，这样可以视为同类竞争。如果当前测试参数组合，在同码率下不如以前的推荐参数好，那么选取以前推荐的组合，取而代之继续测试；这个对比也可以是另一种编码器的，比如测试 HEVC 的时候用 AVC 某个固定画质定位做对比（比如一套基于 crf=17 的参数），当我们某一轮的优胜者第一次好于 avc 的样本时候，我们可以说，当前测试，得出怎样的参数组合，画质据测试略好于 avc 版本，码率对比多少多少……这种可以看做外来生物入侵。

14. 换不同的源验证，特别是类似“怎样的参数组合比 avc 参照好，码率对比多少”这种结论，在不同的源上试着验证一下结论是否依旧靠谱。这种可以看做环境改变，检查当前基因组合所缔造的个体，在不同环境中的存活能力。

## 5. 测试的系统性问题和解决思路

这个测试机制并非完美，有一些内在问题，主要表现在：

(1). 依赖人眼，判断可能不准确。这对测试者要求很高，而且是技术+状态。光有一双训练有素的眼睛还不够，最好比较前来杯咖啡+闭目养神 10 分钟，比较过程中不要被任何东西打扰分神。如果某一轮判断有误，这个测试机制自带一套( 缓慢的 )纠错机制。比如这一轮应该调 crf 但是调了 qcomp ,那么下一轮很可能你判断不会调 qcomp，因为 qcomp 当前过高，调节它收益极小。后续的测试你会选择去调节 crf 之类的，最终，参数还是会趋向于最优化。

(2). 需要的测试量非常大。总测试量=深度*宽度*广度，其中深度定义为测试的轮数，深度越深，测试轮数越多，测试的结果越精确 ,某一轮决策失误造成的影响越小 ;宽度定义为总调节的参数数量和基因突变、基因重组的频率，宽度越宽，单轮测试压制越多，能达到的效果就越好；广度定义为测试的次数，比如 vcb-s 目前大规模测试一共 3 次，广度越广，每一次新测试可以利用的知识越多，更利于纠正前几次测试的偏差，且验证参数在不同片源类型上的效果。深度、宽度、和广度，这三个是用乘号相连的。如果我们需要保证测试的可靠性，这三个都不能低：

vcb-s 每次测试深度大约是 15 轮；

宽度大约是 4 个主要参数+4~6 个不常用参数，加上基因突变和重组，以及每轮重做体积过大过小的成品，大约宽度算 6，平均每轮 6 次压制；

广度是 3，大规模测试一共 3 次

那么总测试数量=15*6*3=225 次。270 次半集压制，相当于 10 季度新番的压制量，相当恐怖，然而更恐怖的是每个样品观察 20 帧左右的画面，270*20=5400 帧的对比，这更恐怖。

(3). 一条路走到黑：按照设计，我们的主要参数一般都是随着码率提升只加不减，但可能并非如此：比如有人认为高码率下应该限制 psy，因为 psy 过大，造成的失真和时域瑕疵，在高画质下反而得不偿失。换言之，中码率可能鼓励高 psy，但是高码率应该适当降低 psy。问题是中码率下我们可能已经将 psy 拉的很高，高码率怎么降回去？这时候上一章 12 点里面说的基因重组就可以帮助我们。高码率下重新组合给出一个低 psy 的设置，看看效果。

(4). 局部最优陷阱：考虑如下问题，$f(x,y)=x+0.8y^2$, x,y>0. 问 x+y<=10 前提下，f(x,y)最大值？很容易得出，x=0,y=10,f(x,y)=80 是最大值。但是按照我们的思路：

从(0,0)开始，每次 x 增加 1 或者 y 增加 1，看看哪个更好，就选择哪个，直到 x+y=10。如果按照这个算法走，你每一步都会选择增加 x，因为每次 x 增加 1，总和增加 1；y 增加 1，总和增加 0.8。提高深度，每次只增加 0.5，只会让你更鼠目寸光。但是你如果一下增加 2，反而会让你做出正确的选择，这下 x 增加是 2，y 的增加是 3.2。这就是基因重组的另一个作用：在降低一些参数的同时，也会飞跃一些参数，看看是不是之前陷入了局部最优。

(5). 我们假定 tradeoff 类的参数对目视画质没有影响，但是影响是存在的，只不过一般很小。我们可以偶尔对于一套参数，用--preset placebo/medium 跑跑看，看看目视质量是不是有突变，如果有，再找出是哪个参数造成的。

## 6. tradeoff 参数的测试

那些以时间换效率的参数，测试思路类似：

(1). 从一个很快的参数开始

(2). 每轮测试，改变不同参数，使得压缩慢一点，比如可以增加 ref，可以加强 ME，但是所有改变总用时相似。微调 crf 使得所有改版体积相似。（通常不需要超过 0.3 级别地调整，否则说明你改的参数会显著地影响 rate control 策略）

(3). 比较 psnr/ssim 这种客观跑分值。选择优胜者

(4). 引入基因突变，重组，外来生物和环境改变。