

参考手册

DiscretePhoton H.264 encoder

参考手册

总观的DiscretePhoton H.264 encoder

DiscretePhoton H.264 encoder的 Windows版本是以 DirectX Media Object (DMO)的32-bit 和64-bit的形式提供的。因此, 直接或者在Direct-Show环境中可以使用。

DiscretePhoton H.264 encoder能支援最多64个thread, 但实际上编码(encoding)的时候所用的thread数目是根据使用者的电脑 CPU core(或是 hyper-thread)的数目和屏幕的大小来决定。使用者可以减少thread的数目。

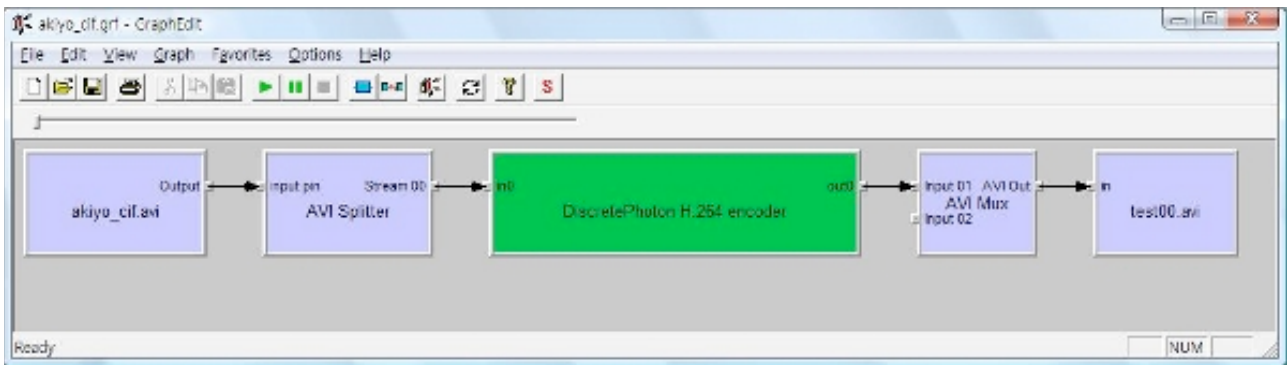
DiscretePhoton H.264 encoder的 multi-threading 是基于wavefront方式。所有的工作thread是集中于最近输入的屏幕上的 encoding工作, 这个方式没有最终品质或压缩大小的低下, 而可以带来非常低的 encoding处理迟延效果。本产品的 CBR 比率控制 (rate-control)方式也适合实时处理方案。所以DiscretePhoton H.264 encoder, 比如视频会议一样的即时性非常重要部分的application特别合适。

较低的 encoding处理迟延还意味着处理迟延变动量的减少。所以这种方式实时 encoding的时候会带来非常低的丢帧率(frame drop-rate)。

DiscretePhoton H.264 encoder的性能比较测试资料可以在下网站上可以找到。
[7-th annual video codec comparison by MSU](#)。

为了更仔细的内容和试用版的下载, 请访问 www.discretephoton.com。

利用 GraphEdt.exe的使用方法



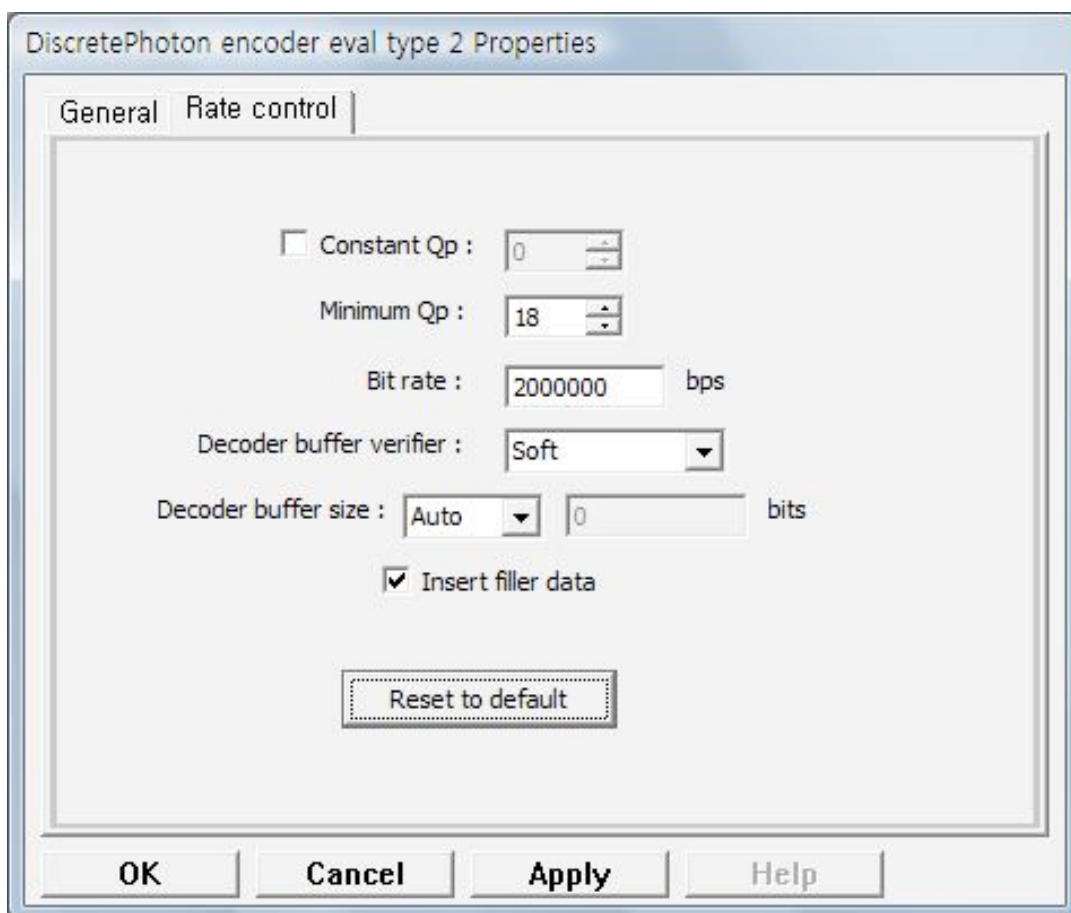
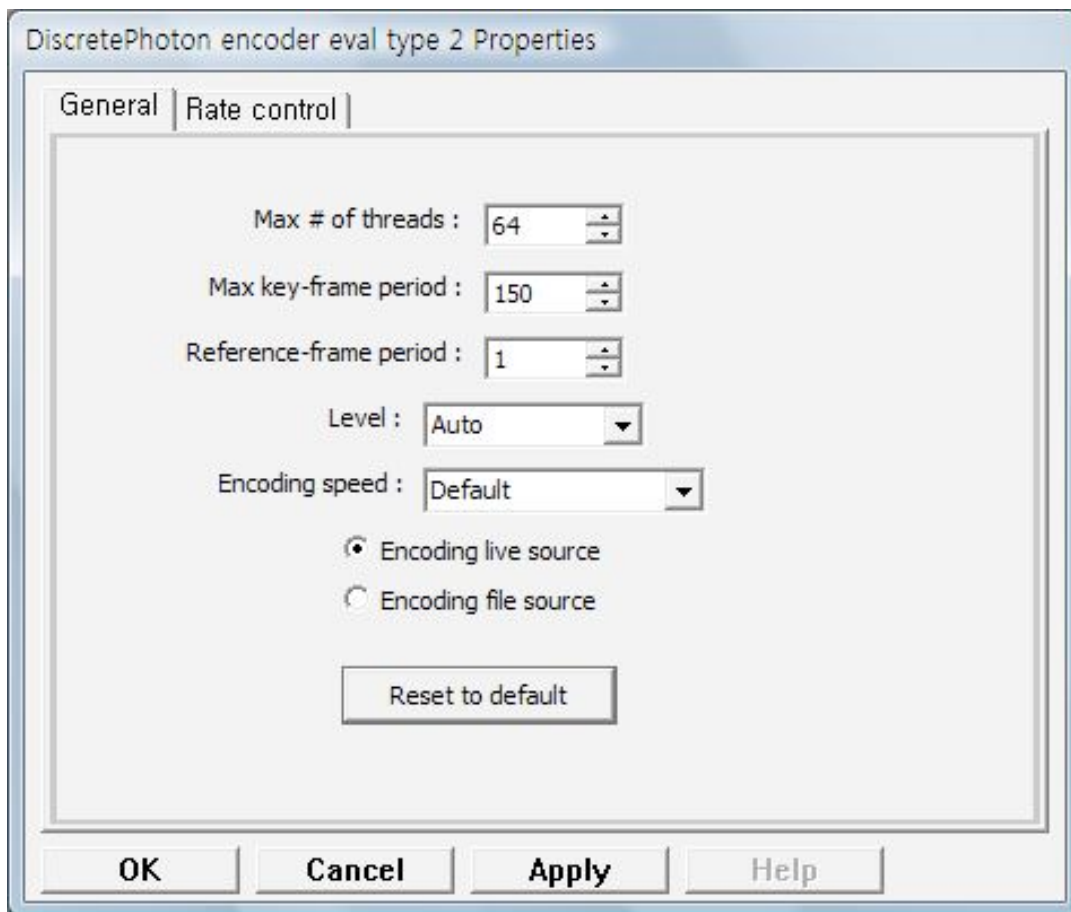
GraphEdt.exe包含在Windows SDK。利用这个软件，可以视觉地构成 Direct-Show过滤器图表，然后进行测试。

安装32-bit版本的 DiscretePhoton H.264 encoder后，从GraphEdt.exe选单上在 Graph -> Insert Filters... -> Video Compressors 可以找到。

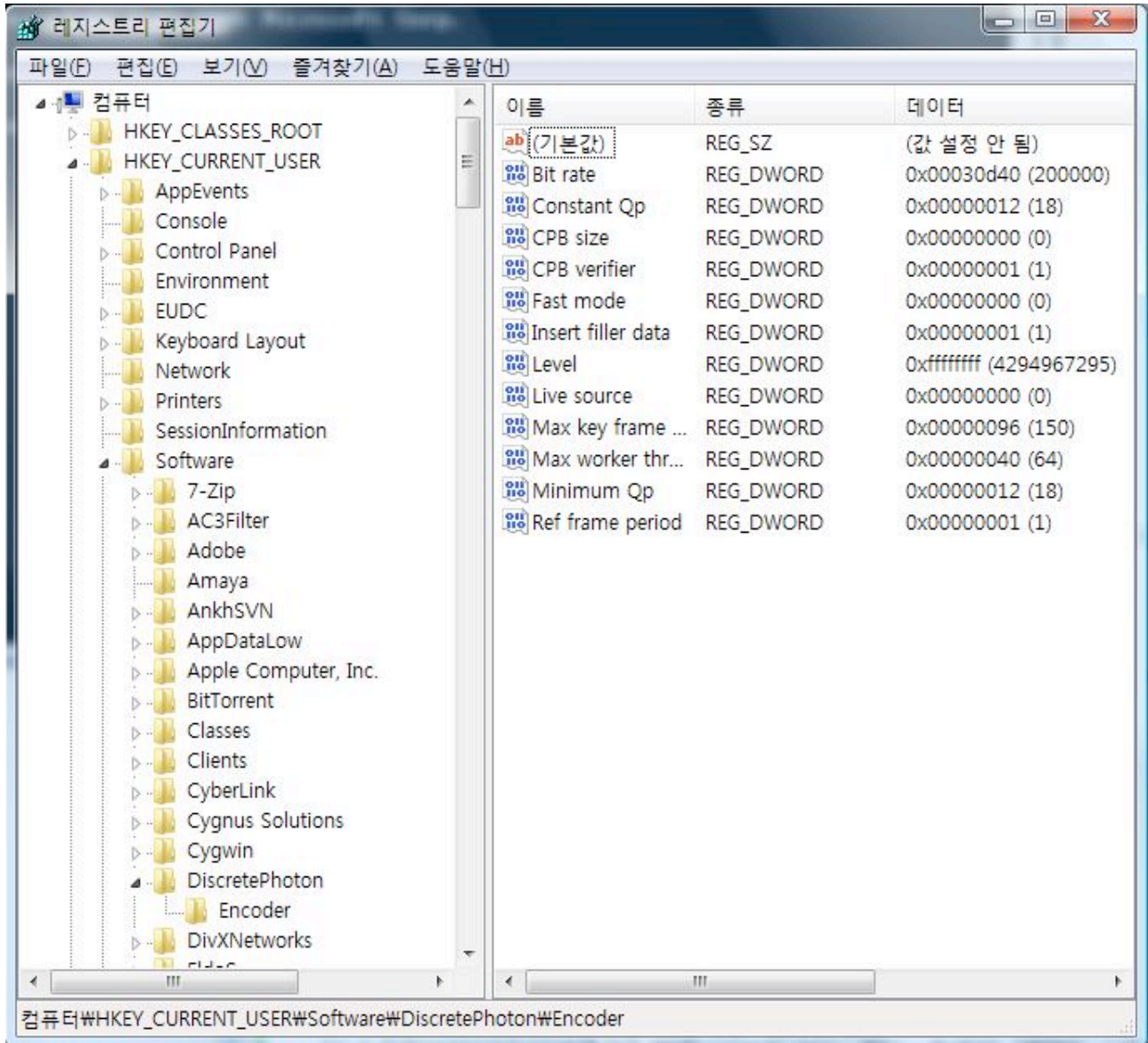
在上图中H.264 encoder 节点显示为绿色，这就表示着事实上这不是Direct-Show过滤器而是 DMO(DirectX Media Object)。

DiscretePhoton H.264 encoder用视频输入格式可以处理 **I420, YV12, UYVY, YUYV, RGB24, RGB32**，而且那处理结果用 H.264 byte stream (FourCC: **H264**)输出。

右击上面的DiscretePhoton H.264 encoder 节点 (node) 会出现以如下属性页面。



属性页面有'General' 和 'Rate control' 两个选项卡。在这里变更数值后使用的时候, 变更后的数值保存在 windows registry (位置: HKEY_CURRENT_USER\Software\DiscretePhoton\Encoder)。使用者可以直接读取或变更 registry上的数值。



各属性的意义如下。

Max # of threads: (registry上的名字 : "Max worker threads") encoding过程中使用的最大 thread数目。这个数字不是实际值而是最大值, 这是因为实际值是跟据使用者电脑的 CPU core(或是 Hyper thread)数目和屏幕的大小来决定的。使用者如果在 encoding过程中不想CPU使用100%的话, 可以把这个数值调成比CPU core的数目少的。

Max key-frame period: (registry上的名字: "Max key frame period") 这里的 关键帧 (key-frame) 意味着 H.264上的IDR帧。这也指定最大值的, 不是实际值, 因为实际的 keyframe间隔可能encoding效率上比指定值更短。 Key-frame间隔的最小值是在这里的指定值的1/2。

Reference-frame period: (registry上的名字: "Ref frame period") 基本上所有的帧都以参考帧(reference frame) 来使用。但是, 增加这个值 (这时只有一部分的帧以reference使用) 和减少 Max key- frame period (像UDP一样的), 可以减少不确定的网络上视频传送时错误。

Level: (registry上的值: -1:Auto, 0:Level_1, 1:Level_1b, ...) 这个是指定H.264 encoding等级(出现在标准文件 annex A.3)的。通常情况下, 使用者可以把这个值指定为'Auto'。这时该等级根据已有的其他参数和输入视频来决定。

Encoding speed: (registry上的名字: "Fast mode". 0:Default, 1:Fast, 2:Even faster) 更快的 encoding速度可以通过 (稍微) 牺牲品质的方式来取得。所以使用者如果使用比较高的输出比特率 (bit-rate) 的话, 此项可以指定为'Fast' 或是 'Even faster'。相反, 如果使用比较低的输出比特率 (bit-rate) 的话, 指定为'Even faster'会带来屏幕上错误的现象。

Encoding live source / Encoding file source: (registry上的名字: "Live source". 0:file source, 1:live source) 如果编码 (encoding) webcam或是 TV tuner等直播视频的话, 请设定为'Encoding live source', 如果编码 (encoding) 文件的话, 请设定为'Encoding file source'。

Constant Qp: (registry上的值: 0是非活性化) 应用这一项目的情况下, 输出视频品质是可以维持几乎相同, 但输出bit-rate会有相当大的变动。在这种情况下, 输出bit-rate的调节过程被省略。因此, 对通过网络传输的视频来说, 最好不使用此项目。这个值的范围是18 - 51。数值越大, 压缩率越高而品质会越下降。

Minimum Qp: 对输出比特率(bit-rate)的比率控制(rate-control)过程中, 比在这里指定的数值更低的 Qp (quantization parameter)值是不会使用的。

Bit rate: 输出比特率(bit-rate)的指定。每秒钟比特(bit)数。

Decoder buffer verifier: (registry上的名字: "CPB verifier". 0:Hard, 1:Soft, 2:None) 设定解码器缓冲器(decoder buffer)验证过程的严密性程度。这里所说的decoder buffer是出现在H.264标准文件annex C的 CPB(coded picture buffer)。(注意:即使把这个值设为'Hard'也不意味着完善的验证。)

Decoder buffer size: (registry上的名字: "CPB size") 指定Bit单位的大小。把这个值指定为'Auto'的话, 会设定为最大容许值(VCL HRD)。

Insert filler data: 如果decoder buffer接近overflow, Qp也接近最小值的话, 通过把 filler data插入在输出比特流(bit stream)来维持输出bit-rate。因此, 通常此项目是必须设定的。

利用DirectShow的编程

关于在 DirectShow环境上的编程的参考资料是应该很容易取得的。

利用 DiscretePhoton H.264 encoder的比较简单例子程序在以下地点中可以下载到。

32-bit: <http://www.discretephoton.com/php/downloader.php?f=Examples.zip>.

64-bit: http://www.discretephoton.com/php/downloader.php?f=Examples_x64.zip.

为了build上面例子程序的 C++版本, 需要包含在Windows SDK的DirectShow base class library。还有 C#版本是利用DirectShow.Net。更仔细的内容请参照在举例程序中添加的 README.htm。