

참조 매뉴얼

DiscretePhoton H.264 encoder

참조 매뉴얼

DiscretePhoton H.264 encoder 개요

DiscretePhoton H.264 encoder는 DiscretePhoton 사의 자체 창작물로서 국산 소프트웨어입니다.

본 H.264 encoder의 Windows version은 32-bit 와 64-bit의 DirectX Media Object (DMO) 형태로 제공됩니다. 따라서 직접 또는 DirectShow 환경에서 사용될 수 있습니다.

본 H.264 encoder는 최대 64개의 thread를 지원하고 있습니다. 하지만 실제로 인코딩 시 활용되는 thread 갯수는 사용자PC의 CPU코어(또는 hyper-thread) 갯수와 화면의 크기에 의해 결정됩니다. 사용자가 thread의 갯수를 더 감소시킬 수는 있습니다.

본 H.264 encoder의 multi-threading 은 wavefront 라는 방식에 기초하고 있습니다. 모든 작업 thread는 가장 최근에 입력을 받은 화면의 인코딩 작업에 집중하게 되며, 이 방식은 최종 품질이나 압축 크기의 저하 등이 없이 아주 낮은 인코딩 처리 지연효과를 가져올 수 있습니다. 본 제품의 CBR rate-control 방식 또한 실시간 처리 시나리오에 맞추어져 있습니다. 따라서 DiscretePhoton H.264 encoder는 비디오 컨퍼런싱과 같은 즉시성이 중요한 분야의 어플리케이션에 특히 더 적합합니다.

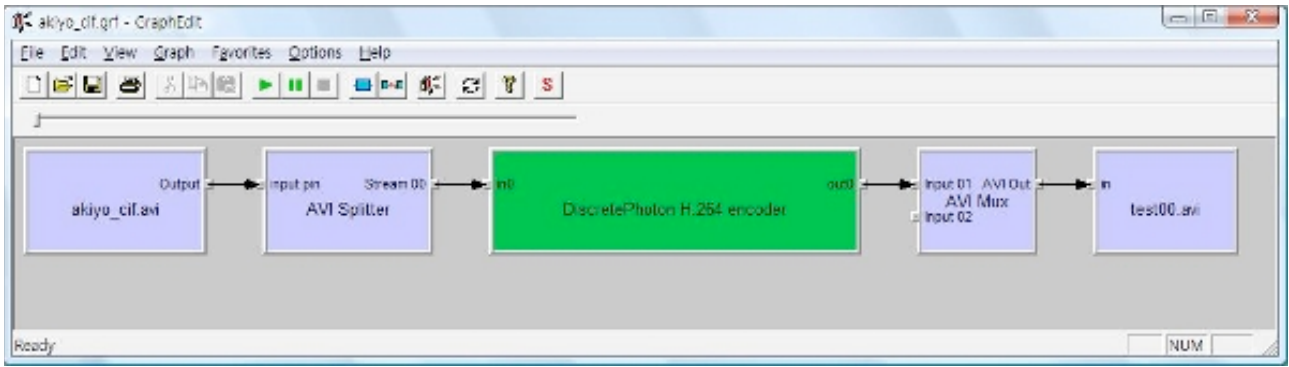
낮은 인코딩 처리 지연은 또한 처리 지연 변동량의 감소를 의미합니다. 따라서 이 방식은 실시간 인코딩 시 아주 낮은 프레임 drop-rate를 가져올 수 있을 것입니다.

본 H.264 encoder의 성능 비교 테스트자료를 다음 위치에서 찾아보실 수 있습니다.

[7-th annual video codec comparison by MSU.](#)

더 자세한 내용과 평가판의 다운로드를 위해서는 www.discretephoton.com을 방문하여 주시기 바랍니다.

GraphEdt.exe 를 이용한 사용법



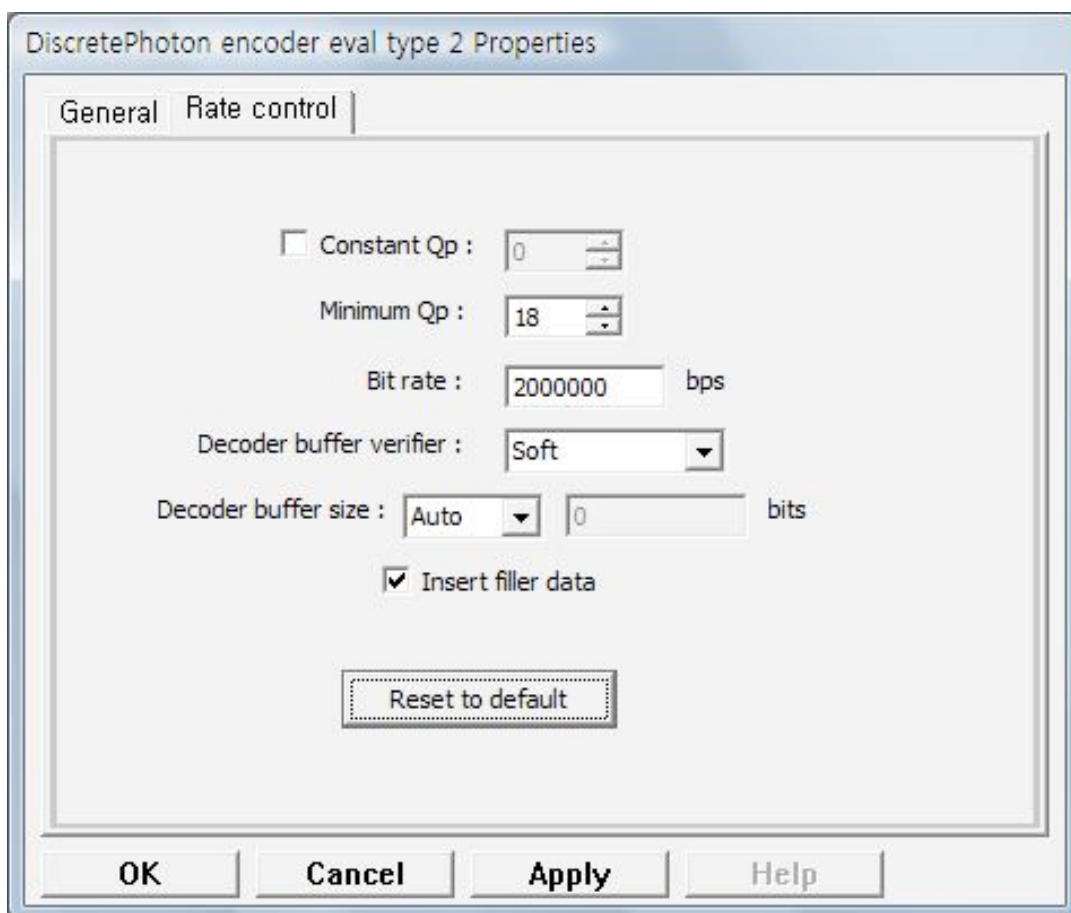
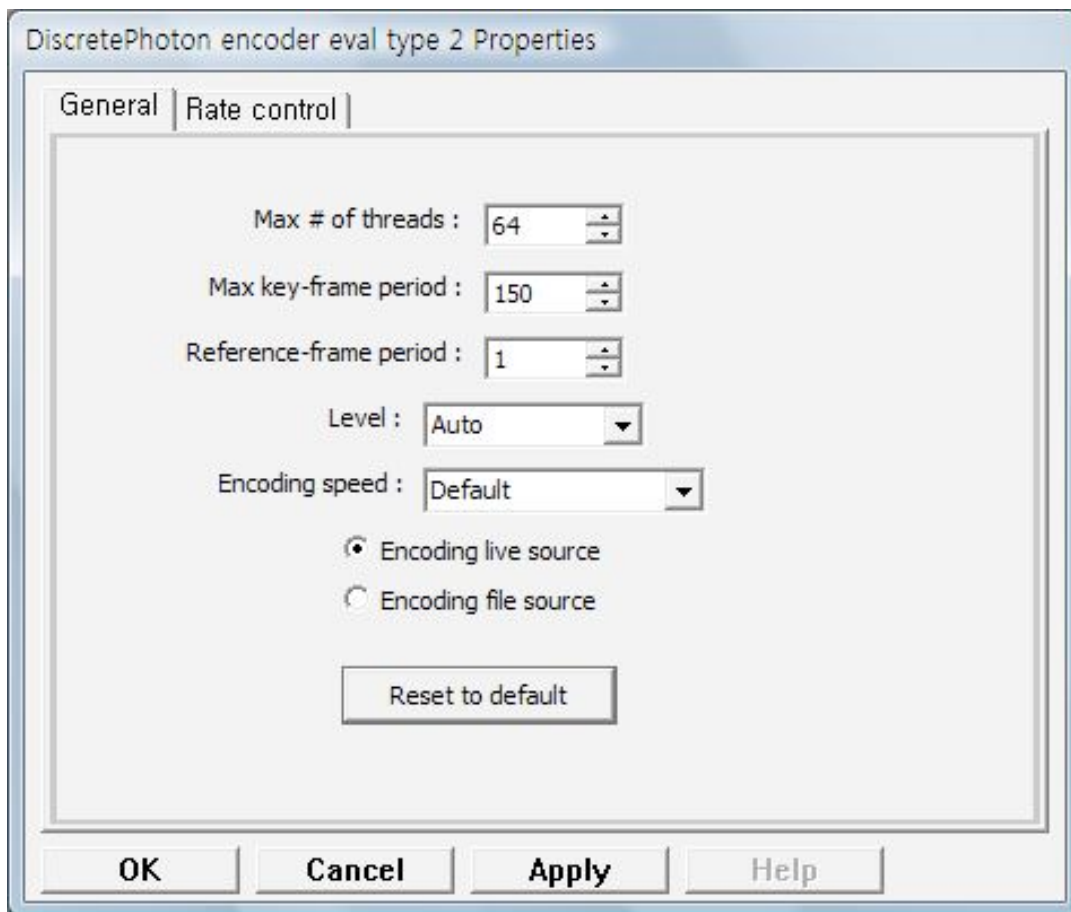
GraphEdt.exe는 Windows SDK에 포함되어 있습니다. 이 프로그램을 이용하면 DirectShow 필터 그래프를 시각적으로 구성하고 테스트 해 볼 수 있습니다.

32-bit 버전의 DiscretePhoton H.264 encoder 설치 후, GraphEdt.exe의 메뉴바로 부터 Graph -> Insert Filters... -> Video Compressors 위치에서 찾을 수 있습니다.

위 그림에서 H.264 encoder 노드가 녹색으로 표시되어 있는 것은 이것이 사실상 DirectShow 필터가 아닌 DMO(DirectX Media Object) 임을 나타냅니다.

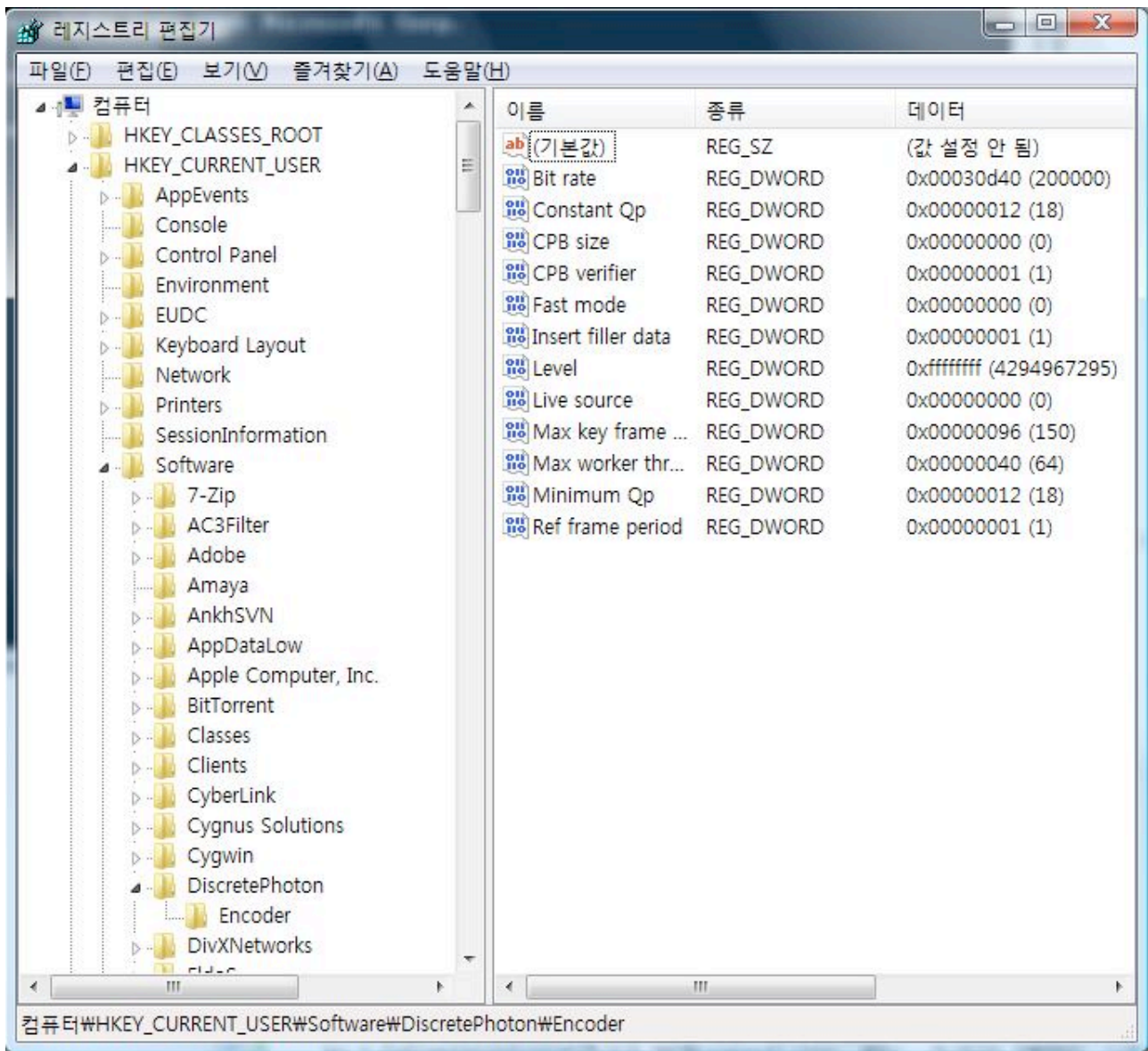
DiscretePhoton H.264 encoder 는 입력 비디오 포맷으로 **I420, YV12, UYVY, YUYV, RGB24, RGB32** 를 처리할 수 있으며 그 처리결과는 H.264 byte stream (FourCC: **H264**)으로 출력 합니다.

위 DiscretePhoton H.264 encoder 노드를 마우스 오른쪽 버튼으로 클릭하면 다음과 같은 속성 페이지를 나타나게 할 수 있습니다.



속성 페이지에는 'General' 과 'Rate control' 이라는 두 개의 탭이 있습니다. 이곳에서 값을 변경 후 적용할 경우 그 변경된 값들은 윈도우즈 레지스트리(위치:

HKEY_CURRENT_USER\Software\DiscretePhoton\Encoder)에 저장 됩니다. 사용자는 이 레지스트리 상의 값들을 직접 읽거나 변경할 수도 있습니다.



각 속성 항목의 의미는 다음과 같습니다.

Max # of threads: (레지스트리상 이름: "Max worker threads") 인코딩 과정에서 활용될 최대 thread 갯수. 이 숫자는 실제값이 아닌 최대값으로 되어 있고, 그 이유는 실제값이 사용자 PC의 CPU 코어(또는 Hyper thread) 갯수와 화면의 크기에 의해 결정되어지기 때문입니다. 사용자는 만약 CPU의 100%를 인코딩 과정에서 사용하고 싶지 않다면 이 값을 CPU 코어의 갯수보다 더 낮게 설정할 수도 있습니다.

Max key-frame period: (레지스트리상 이름: "Max key frame period") 여기서의 키 프레임은 H.264상의 IDR 프레임을 의미합니다. 이것 역시 실제값이 아닌 최대값을 지정하도록 되어 있는데, 실제의 키 프레임 간격은 인코딩 효율상 지정된 값 보다 더 짧게 나타날 수도 있기 때문입니다. 키 프레임 간격의 최소값은 여기서 지정된 값의 1/2로 설정하고 있습니다.

Reference-frame period: (레지스트리상 이름: "Ref frame period") 기본적으로는 모든 프레임이 레퍼런스 프레임으로 사용됩니다. 하지만 이 값을 증가시키고(이 경우 일부의 프레임들만 레퍼런스로 사용됨) Max key- frame period 값을 감소시킴으로써 (UDP와 같은) 불확실한 네트워크를 통한 비디오 전송 시의 오류를 감추는 효과를 얻을 수도 있을 것입니다.

Level: (레지스트리상 값: -1:Auto, 0:Level_1, 1:Level_1b, ...) 이것은 H.264 인코딩 레벨(표준 문서 annex A.3에 나타나는)을 지정합니다. 보통의 경우 사용자는 이 값을 'Auto'로 지정할 수 있습니다. 이 때 레벨은 주어진 다른 파라미터와 입력 비디오에 의해 자동 결정 됩니다.

Encoding speed: (레지스트리상 이름: "Fast mode". 0:Default, 1:Fast, 2:Even faster) 더 빠른 인코딩 속도는 품질을 (조금)희생시키는 방식으로 얻어질 수도 있습니다. 따라서 만약 사용자가 비교적 높은 출력 bit-rate를 사용하고 있다면, 이 항목을 'Fast' 또는 'Even faster'로 지정할 수 있을 것입니다. 반대로 낮은 출력 bit-rate를 사용하고 있다면 'Even faster'로 지정할 경우 결과 화면에서 블러가 보이는 현상이 나타날 수도 있습니다.

Encoding live source / Encoding file source: (레지스트리상 이름: "Live source". 0:file source, 1:live source) 만약 웹캠이나 TV튜너 등과 같은 라이브 비디오를 인코딩하는 경우라면 'Encoding live source'를, 그렇지 않고 파일 등을 인코딩하는 경우라면 'Encoding file source'를 설정하시기 바랍니다.

Constant Qp: (레지스트리상 값: 0은 비활성화) 이 항목이 적용될 경우 출력 비디오 품질은 거의 동일성을 유지하게 될 것이나, 출력 bit-rate은 상당히 큰 변동성을 가질 수 있습니다. 즉 이 경우 출력 bit-rate의 조정 과정이 생략 됩니다. 따라서 네트워크를 통해 전송되어질 비디오에 대해서는 이 항목을 사용하지 않는 것이 좋습니다. 이 값의 범위는 18 - 51 입니다. 값이 클수록 압축률이 높아지고 품질은 낮아지게 됩니다.

Minimum Qp: 출력 bit-rate에 대한 rate-control 과정에서 여기에서 지정된 값 보다 더 낮은 Qp (quantization parameter) 값을 사용하지 않게 됩니다.

Bit rate: 출력 bit-rate의 지정. 초당 bit수.

Decoder buffer verifier: (레지스트리상 이름: "CPB verifier". 0:Hard, 1:Soft, 2:None) 디코더 버퍼 검증 과정의 엄밀성 정도를 설정합니다. 여기서 말하는 디코더 버퍼는 H.264표준 문서 annex C에 나타나는 CPB(coded picture buffer)에 해당합니다. (주의: 이 값을 'Hard' 로 설정하더라도 완벽한 검증을 의미하지는 않습니다.)

Decoder buffer size: (레지스트리상 이름: "CPB size") Bit 단위의 크기를 지정합니다. 이 값을 'Auto'로 지정 하면 허용되는 최대값(VCL HRD)으로 설정되어 집니다.

Insert filler data: 만약 디코더 버퍼가 오버플로우에 근접하고 Qp 또한 최소값에 근접해 있다면 filler data 라는 것을 출력 비트 스트림에 삽입함으로써 출력 bit-rate을 유지하게 됩니다. 따라서 일반적으로 이 항목은 필수적으로 설정되어 있어야 합니다.

DirectShow 를 이용한 프로그래밍

DirectShow 환경에서의 프로그래밍에 관한 참고자료들은 어렵지 않게 접할 수 있으리라 생각합니다.

DiscretePhoton H.264 encoder 를 이용한 비교적 단순한 예제프로그램들을 다음 위치에서 다운로드하실 수 있습니다.

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.

위 예제프로그램의 C++ 버전을 빌드하기 위해서는 Windows SDK에 포함된 DirectShow base class library라는 것이 필요합니다. 그리고 C# 버전의 경우 DirectShow.Net을 이용하고 있습니다. 더 자세한 내용은 예제프로그램에 첨부된 README.htm 을 참조하시기 바랍니다.