Xvid

Junjie Cao

Introduction

- What’s Xvid?
  - An open source implementation of the MPEG-4 standard.
- Why Xvid
  - Open Source DivX 3.11 - > DivX 4
  - Closed source DivX 5
  - Open Source Xvid

Architecture

- The Xvid source code is written in C
- Encoding part
- Decoding part

The Xvid version

```
#define API_VERSION ((2 << 16) |
(1))
```

– 2 stands for the major XviD version

– 1 stands for the minor version

XVID_INIT_PARAM

```
Typedef struct
{
  int cpu_flags       [in/out]
  int api_version [out]
  int core_build  [out]
} XVID_INIT_PARAM
Xvid_init(NULL, 0,&xinit, NULL);
```

Valid flags

- XVID_CPU_MMX
- XVID_CPU_MMXEXT
- XVID_CPU_SSE
- XVID_CPU_SSE2
- XVID_CPU_3DNOW
- XVID_CPU_3DNOWEXT
- XVID_CPU_TSC
- XVID_CPU_IA64
- XVID_CPU_CHKONLY
- XVID_CPU_FORCE

```
xinit.cpu_flags |=
  desired_flag_constant;
```

XVID_ENC_PARAM

```
Typedef struct
{
   int width, height;             [in]
   int fincr, fbase;       [in] eg. [1,25] [1000,
   23996]
   int rc_bitrate;         [in] (default value: 900000)
   int rc_reaction_delay_factor;[in]
   int rc_averaging_period;[in]
   int rc_buffer;          [in]
   int max_quantizer;             [in] (default value:
   31)
   int min_quantizer;             [in] (default value:
   1)
   int max_key_interval;   [in] (default value:
   10*framerate)

   void *handle;           [out]
```

XVID_ENC_FRAME

```
Typedef struct
{
   int general;       [in]
   int motion;        [in]
   void *bitstream;   [in]
   int length;        [out]
   void *image;       [in]
   int colorspace;    [in]
   unsigned char *quant_intra_matrix; [in]
   unsighed char *quant_inter_matrix; [in]
   int quant;                         [in/out]
   int intra;                         [in/out]
   HINTINFO hint;                     [in/out]
}
Xerr = xvid_encore(enchandle, XVID_ENC_ENCODE, &xframe,
   &xstats);
```

Genral flag member

- XVID_CUSTOM_QMATRIX : informs xvid to use the custom user matrices.
- XVID_H263QUANT : informs xvid to use H263 quantization algorithm.
- XVID_MPEGQUANT : informs xvid to use MPEG quantization algorithm.
- XVID_HALFPEL : informs xvid to perform a half pixel motion estimation.
- XVID_ADAPTIVEQUANT : informs xvid to perform an adaptative quantization.
- XVID_LUMIMASKING : infroms xvid to use a lumimasking algorithm.
- XVID_INTERLACING : informs xvid to use the MPEG4 interlaced mode.

General flag member

- XVID_HINTEDME_GET : informs xvid to return Motion Estimation vectors from the ME encoder algorithm. Used during a first pass.
- XVID_HINTEDME_SET : informs xvid to use the user given motion estimation vectors as hints for the encoder ME algorithms. Used during a 2nd pass.
- XVID_INTER4V : forces XviD to search a vector for each 8x8 block within the 16x16 Macro Block. This mode should be used only if the XVID_HALFPEL mode is activated (this could change in the future).
- XVID_ME_ZERO : forces XviD to use the zero ME algorithm.
- XVID_ME_LOGARITHMIC : forces XviD to use the logarithmic ME algorithm.
- XVID_ME_FULLSEARCH : forces XviD to use the full search ME algorithm.
- XVID_ME_PMVFAST : forces XviD to use the PMVFAST ME algorithm.
- XVID_ME_EPZS : forces XviD to use the EPZS ME algorithm.

Motion member

Valid flags for 16x16 motion estimation (no XVID_INTER4V flag in the general flag).

- PMV_ADVANCEDDIAMOND16
- PMV_HALFPELDIAMOND16
- PMV_HALFPELREFINE16
- PMV_EXTSEARCH16
- PMV_EARLYSTOP16
- PMV_QUICKSTOP16
- PMV_UNRESTRICTED16
- PMV_OVERLAPPING16
- PMV_USESQUARES16

Motion member

Valid flags when using 4 vectors mode prediction.

- PMV_ADVANCEDDIAMOND8
- PMV_HALFPELDIAMOND8
- PMV_HALFPELREFINE8
- PMV_EXTSEARCH8
- PMV_EARLYSTOP8
- PMV_QUICKSTOP8
- PMV_UNRESTRICTED8
- PMV_OVERLAPPING8
- PMV_USESQUARES8

Quant

- The quantizer value is used when the DCT coefficients are divided to zero those coefficients not important (according to the target bitrate not the image quality)
- 0 (zero) : Then the rate controler chooses the right quantizer for you. Tipically used in ABR encoding or first pass of a VBR encoding session.
- != 0 : Then you force the encoder to use this specific quantizer value. It is clamped in the interval [1..31]. Tipically used during the 2nd pass of a VBR encoding session.

Intra (in usage)

The intra value decides wether the frame is going to be a keyframe or not.

- 1 : forces the encoder to create a keyframe. Mainly used during a VBR 2nd pass.
- 0 : forces the encoder not to create a keyframe. Minaly used during a VBR second pass
- -1: let the encoder decide (based on contents and max_key_interval). Mainly used in ABR mode and dunring a 1st VBR pass.

Intra (out usage)

When first set to -1, the encoder returns the effective keyframe state of the frame.

- 0 : the resulting frame is not a keyframe
- 1 : the resulting frame is a keyframe (scene change).
- 2 : the resulting frame is a keyframe (max_keyframe interval reached)

XVID_ENC_STATS

```
Typedef struct
{
  int quant;      [out]
  int hlength;        [out]
  int kblks, mblks, ublks;
}
Xerr = xvid_encore(enchandle,
  XVID_ENC_ENCODE, &xframe,
  &xstats);
```

The xvid_encode function

```
int xvid_encore(void * handle, int opt, void *
  param1, void * param2);
```

- XviD uses a single-function API, so everything you want to do is done by this routine. The opt parameter chooses the behaviour of the routine
- XVID_ENC_CREATE: create a new encoder, XVID_ENC_PARAM in param1, a handle to the new encoder is returned in handle.
- XVID_ENC_ENCODE: encode one frame, XVID_ENC_FRAME-structure in param1, XVID_ENC_STATS in param2 (or NULL, if you are not interested in statistical data).
- XVID_DEC_DESTROY: shut down this encoder, do not use handle afterwards.

XVID_DEC_PARAM

```
typedef struct
{
  int width; [in]
  int height;
  void *handle;   [out]
}
xerr = xvid_decore(NULL,
  XVID_DEC_CREATE, &xparam, NULL);
```

XVID_DEC_FRAME

```
typedef struct
{ void * bitstream;   [in]
  int length;         [in]

  void * image;       [in]
  int stride;         [in]
  int colorspace;     [in]
}
xerr = xvid_decore(dechandle,
  XVID_DEC_DECODE, &xframe, NULL);
```

XVID_DEC_FRAME

- To provide the MPEG4-bitstream and it's length,
- image is the position where the decoded picture should be stored.
- stride is the difference between the memory address of the first pixel of a row in the image and the first pixel of the next row. If the image is going to be one big block, then stride=width, but by making it larger you can create an "edged" picture.
- By colorspace the output format for the image is given, XVID_CSP_RGB24 or XVID_CSP_YV12 might be might common.

The xvid_decore function

```
int xvid_decore(void * handle, int
opt,  void * param1, void *
param2);
```

- XviD uses a single-function API, so everything you want to do is done by this routine. The opt parameter chooses the behaviour of the routine:
- XVID_DEC_CREATE: create a new decoder, XVID_DEC_PARAM in param1, a handle to the new decoder is returned in handle
- XVID_DEC_DECODE: decode one frame, XVID_DEC_FRAME-structure in param1
- XVID_DEC_DESTROY: shut down this decoder, do not use handle afterwards

Referance

- http://www.xvid.org
- http://people.ee.ethz.ch/~fhoesli/video/test/