

# Scalable Video Technology for AV1 Encoder (SVT-AV1 Encoder) User Guide

# Table of Contents

1. Introduction.....	3
2. System Requirements .....	3
2.1 CPU requirements.....	3
2.2 RAM requirements .....	3
2.3 Operating systems .....	3
Windows* Operating Systems (64-bit).....	3
Linux* Operating Systems (64-bit).....	3
Unix* Operating Systems (64-bit) .....	3
2.4 Build the code .....	3
Windows* Operating Systems (64-bit).....	3
Linux* Operating Systems (64-bit).....	4
2.5 Installation .....	4
3 Sample Application Guide.....	5
3.1 Input Video Format.....	5
3.2 Compressed 10-bit format .....	5
3.2.1 Unpack the 10 bit picture .....	5
3.2.2 Compress the 2 bit Plane.....	6
3.2.3 Unroll the 64x64 .....	6
3.3 Running the encoder .....	6
3.3.1 List of all configuration parameters.....	7
Legal Disclaimer.....	11

## 1. Introduction

This document describes the system requirements and how to use the Scalable Video Technology for AV1 Encoder (SVT-AV1). In particular, this user guide section describes how to run the sample application with the respective dynamically linked library.

## 2. System Requirements

The SVT-AV1 Encoder library was developed to be supported on x86 for Windows\* and Linux\* operating systems.

### 2.1 CPU requirements

In order to achieve the performance targeted by the SVT-AV1 Encoder, the specific CPU model listed above would need to be used when running the encoder. Otherwise, the encoder runs on any 5th Generation Intel® Core™ processor, (Intel® Xeon® CPUs, E5-v4 or newer).

### 2.2 RAM requirements

In order to run the highest resolution supported by the encoder, at least 48GB of RAM is required to run a single 4k 10-bit encode. The encoder application will display an error if the system does not have enough RAM to support this resolution. The table below lists the minimum amount of RAM required for some standard resolutions of 10bit video per channel:

Resolution	Minimum Footprint in GB
4k	48
1080p	16
720p	8
480p	4

### 2.3 Operating systems

The list below includes the operating systems that the encoder application and library could run on, assuming the above pre-requisites are met.

#### Windows\* Operating Systems (64-bit)

- Windows\* 10
- Windows\* Server 2016 Standard

#### Linux\* Operating Systems (64-bit)

- Ubuntu\* 16.04 Desktop LTS
- Ubuntu\* 16.04 Server LTS
- Ubuntu\* 18.04 Desktop LTS
- Ubuntu\* 18.04 Server LTS

#### Unix\* Operating Systems (64-bit)

- MAC OS\*X

### 2.4 Build the code

The list below includes the build tools necessary for the encoder application and library to build properly.

#### Windows\* Operating Systems (64-bit)

- Build requirements
  - o Visual Studio\* 2017

\*Other names and brands may be claimed as the property of others.

- YASM Assembler version 1.2.0 or later
  - Download the yasm exe from the following [link](#)
  - Rename yasm-1.3.0-win64.exe to yasm.exe
  - Copy yasm.exe into a location that is in your system PATH environment variable
- CMake 3.5 or later [link](#)
- Build instructions
  - Generate the Visual Studio\* 2017 project files by following the steps below in a windows command line prompt:
    - In the repository main directory go under the <repo dir>\Build\windows location
    - run generate\_vs17.bat [such would generate the visual studio project files]
  - Open the "svt-av1.sln" using Visual Studio\* 2017 and click on Build --> Build Solution
- Binaries Location
  - Binaries can be found under <repo dir>\Bin/Release or <repo dir>\Bin/Debug, depending on whether Debug or Release were selected in the build mode

## Linux\* Operating Systems (64-bit)

- Build requirements
  - GCC 5.4.0
  - CMake 3.5.1
  - YASM Assembler version 1.2.0 or later
- Build instructions
  - cd Build/linux
  - chmod +x build.sh
  - ./build.sh <release | debug> (if none specified, both release and debug will be built)
- Binaries Location:
  - Binaries can be found under Bin/Release or Bin/Debug

## 2.5 Installation

For the binaries to operate properly on your system, the following conditions have to be met:

- Windows\*:
  - On any of the Windows\* operating systems listed in section 2.3, Install Visual Studio 2017
  - Once the installation is complete, copy the binaries to a location making sure that both the sample application "SvtAv1EncApp.exe" and library "SvtAv1Enc.dll" are in the same folder.
  - Open the command line at the chosen location and run the sample application to encode.
- Linux\*:
  - On any of the Linux\* operating systems listed in section 2.3, copy the binaries under a location of your choice.
  - Change the permissions on the sample application "SvtAv1EncApp" executable by running the command:
    - chmod +x SvtAv1EncApp
  - Open the terminal and cd into your chosen location
  - Run the sample application to encode.

\*Other names and brands may be claimed as the property of others.

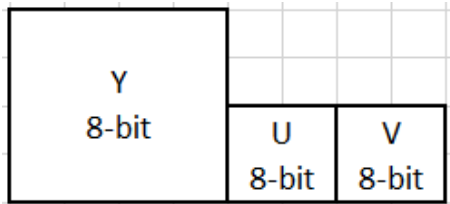
## 3 Sample Application Guide

This section describes how to run the sample encoder application that uses the SVT-AV1 Encoder library. It describes the input video format, the command line input parameters and the resulting outputs.

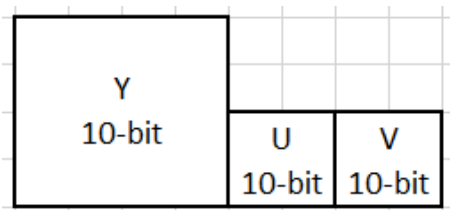
### 3.1 Input Video Format

The SVT-AV1 Encoder supports the following input formats:

- 8-bit yuv420p



- 10-bit yuv420p10le

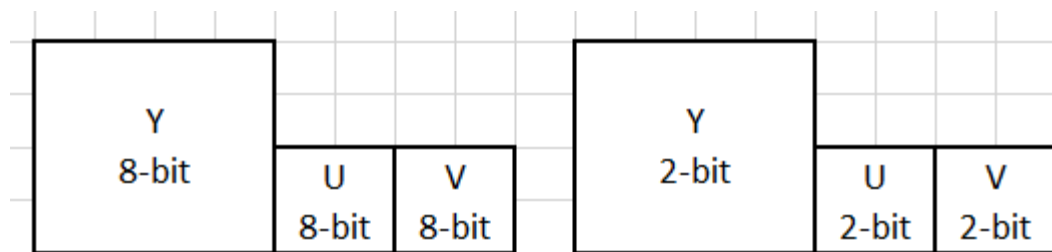


### 3.2 Compressed 10-bit format

In order to reduce the size of the input original YUV file, the SVT-AV1 Encoder uses a compressed 10-bit format allowing the software to achieve a higher speed and channel density levels. The conversion between the 10-bit yuv420p10le and the compressed 10-bit format is a lossless operation and is performed using the following steps.

#### 3.2.1 Unpack the 10 bit picture

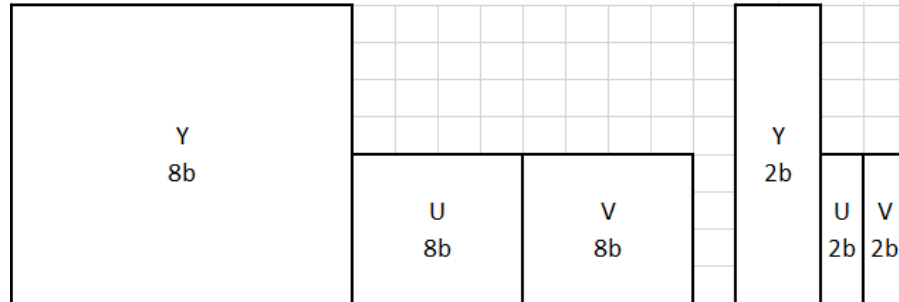
This step consists of separating the 10 bit video samples into 8 bit and 2 bit planes so that each 10-bit picture will be represented as two separate pictures as shown in the figure below. As a result of the operation, the 2 least significant bits of the 10 bits will be written into a full byte.



10-bit yuv420p10le unpacked

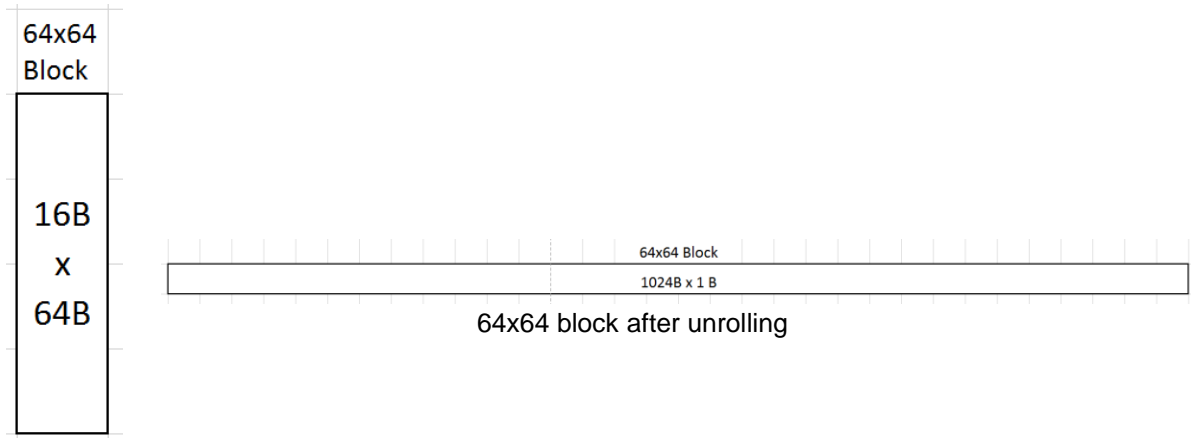
### 3.2.2 Compress the 2 bit Plane

The unpacking steps separates the 10bits into a group of 8 bits and a group of 2 bits, where the 2 bits are stored in a byte. In this step, every group of consecutive 4 bytes, each containing 2bits from the unpacking step, are compressed into one byte. As a result, each 10bit picture will be represented as two separate pictures as shown in the figure below.



### 3.2.3 Unroll the 64x64

Now for a faster read of the samples, every 64x64 block of the 2 bit picture should be written into a one dimensional array. Therefore, the top left 64x64 sample block which is now written into a 16 bytes x 64 bytes after the compression of the 2bit samples, will be written into a 1024 bytes x 1 byte array as shown in the picture below.



64x64 block after 2 bit compression.

## 3.3 Running the encoder

This section describes how to run the sample encoder application SvtAv1EncApp.exe (on Windows\*) or SvtAv1EncApp (on Linux\*) from the command line, including descriptions of the most commonly used input parameters and outputs.

The sample application typically takes the following command line parameters:

**-c filename [Optional]**

A text file that contains encoder parameters such as input file name, quantization parameter etc. Refer to the comments in the Config/Sample.cfg for specific details. The list of encoder parameters are also listed below. Note that command line parameters take precedence over the parameters included in the configuration file when there is a conflict.

\*Other names and brands may be claimed as the property of others.

**-i filename [Required]**

A YUV file (e.g. 8 bit 4:2:0 planar) containing the video sequence that will be encoded. The dimensions of each image are specified by `-w` and `-h` as indicated below.

**-b filename [Optional]**

The resulting encoded bit stream file in binary format. If none specified, no output bit stream will be produced by the encoder.

**-w integer [Required]**

The width of each input image in units of picture luma pixels, e.g. 1920

**-h integer [Required]**

The height of each input image in units of picture luma pixels, e.g. 1080

**-n integer [Optional]**

The number of frames of the sequence to encode. e.g. 100. If the input frame count is larger than the number of frames in the input video, the encoder will loopback to the first frame when it's done.

**-intra-period integer [Optional]**

The intra period defines the interval of frames after which you insert an Intra refresh. It is strongly recommended to use (multiple of 8) -1 the closest to 1 second (e.g. 55, 47, 31, 23 should be used for 60, 50, 30, (24 or 25) respectively). When using closed gop (`-irefresh-type 2`) add 1 to the value above (e.g. 56 instead of 55).

**-rc integer [Optional]**

This token sets the bitrate control encoding mode [1: Variable Bitrate, 0: Constant QP]. When rc is set to 1, it's best to match the `-lad` (lookahead distance described in the next section) parameter to the `-intra-period`. When `-rc` is set to 0, a qp value is expected with the use of the `-q` command line option otherwise a default value is assigned (25).

For example, the following command encodes 100 frames of the YUV video sequence into the bin bit stream file. The picture is 1920 luma pixels wide and 1080 pixels high using the Sample.cfg configuration. The QP equals 30 and the md5 checksum is not included in the bit stream.

```
> SvtAv1EncApp.exe -c Sample.cfg -i CrowdRun_1920x1080.yuv -w 1920 -h 1080 -n 100 -q 30 -intra-period 31 -b CrowdRun_1920x1080_qp30.bin
```

It should be noted that not all the encoder parameters present in the Sample.cfg can be changed using the command line.

### 3.3.1 List of all configuration parameters

The encoder parameters present in the Sample.cfg file are listed in this table below along with their status of support, command line parameter and the range of values that the parameters can take.

Encoder Parameter as shown in the configuration file	Command Line parameter	Range	Default	Description
ChannelNumber	-nch	[1 - 6]	1	Number of encode instances
ConfigFile	-c	any string	null	Configuration file path
InputFile	-i	any string	None	Input file path
StreamFile	-b	any string	null	output bitstream file path
ErrorFile	-errlog	any string	stderr	error log displaying configuration or encode errors
UseQpFile	-use-q-file	[0 - 1]	0	When set to 1, overwrite the picture qp assignment using qp values in QpFile
QpFile	-qp-file	any string	Null	Path to qp file
EncoderMode	-enc-mode	[0 - 3]	3	Encoder Preset [0,1,3] 0 = highest quality, 3 = highest speed
EncoderBitDepth	-bit-depth	[8 , 10]	8	specifies the bit depth of the input video
CompressedTenBitFormat	-compressed-ten-bit-format	[0 - 1]	0	Offline packing of the 2bits: requires two bits packed input (0: OFF, 1: ON)
SourceWidth	-w	[64 - 4096]	None	Input source width
SourceHeight	-h	[0 - 2304]	None	Input source height
FrameToBeEncoded	-n	[0 - 2^64 -1]	0	Number of frames to be encoded, if number of frames is > number of frames in file, the encoder will loop to the beginning and continue the encode. Use -1 to not buffer.
BufferedInput	-nb	[-1, 1 to 2^31 -1]	-1	number of frames to preload to the RAM before the start of the encode If -nb = 100 and -n 1000 -- > the encoder will encode the first 100 frames of the video 10 times
FrameRate	-fps	[0 - 2^64 -1]	25	If the number is less than 1000, the input frame rate is an integer number between 1 and 60, else the input number is in Q16 format (shifted by 16 bits) [Max allowed is 240 fps]
FrameRateNumerator	-fps-num	[0 - 2^64 -1]	0	Frame rate numerator e.g. 6000
FrameRateDenominator	-fps-denom	[0 - 2^64 -1]	0	Frame rate denominator e.g. 100
HierarchicalLevels	-hierarchical-levels	[0 - 5]	3	0 : Flat 3: 4-Level Hierarchy Minigop Size = (2^HierarchicalLevels) (e.g. 3 == > 7B pyramid, 2 == > 3B Pyramid)
IntraPeriod	-intra-period	[-2 - 255]	-2	Distance Between Intra Frame inserted. -1 denotes no intra update. -2 denotes auto.
IntraRefreshType	-irefresh-type	[1 - 2]	1	1: CRA (Open GOP) 2: IDR (Closed GOP)
QP	-q	[0 - 63]	50	Quantization parameter used when RateControl is set to 0



<b>UseDefaultMeHme</b>	<b>-use-default-me-hme</b>	[0 - 1]	1	0 : Overwrite Default ME HME parameters 1 : Use default ME HME parameters, dependent on width and height
<b>HME</b>	<b>-hme</b>	[0 - 1]	1	Enable HME, 0 = OFF, 1 = ON
<b>HMELevel0</b>	<b>-hme-l0</b>	[0 - 1]	1	Enable HME Level 0 , 0 = OFF, 1 = ON
<b>HMELevel1</b>	<b>-hme-l1</b>	[0 - 1]	Depends on input resolution	Enable HME Level 1 , 0 = OFF, 1 = ON
<b>HMELevel2</b>	<b>-hme-l2</b>	[0 - 1]	Depends on input resolution	Enable HME Level 2 , 0 = OFF, 1 = ON
<b>InLoopMeFlag</b>	<b>-in-loop-me</b>	[0 - 1]	Depends on –enc-mode	0=ME on source samples, 1= ME on recon samples
<b>LocalWarpedMotion</b>	<b>-local-warp</b>	[0 - 1]	0	Enable warped motion use , 0 = OFF, 1 = ON
<b>ExtBlockFlag</b>	<b>-ext-block</b>	[0 - 1]	Depends on –enc-mode	Enable the non-square block 0=OFF, 1= ON
<b>SearchAreaWidth</b>	<b>-search-w</b>	[1 - 256]	Depends on input resolution	Search Area in Width
<b>SearchAreaHeight</b>	<b>-search-h</b>	[1 - 256]	Depends on input resolution	Search Area in Height
<b>NumberHmeSearchRegionInWidth</b>	<b>-num-hme-w</b>	[1 - 2]	Depends on input resolution	Search Regions in Width
<b>NumberHmeSearchRegionInHeight</b>	<b>-num-hme-h</b>	[1 - 2]	Depends on input resolution	Search Regions in Height
<b>HmeLevel0TotalSearchAreaWidth</b>	<b>-hme-tot-l0-w</b>	[1 - 256]	Depends on input resolution	Total HME Level 0 Search Area in Width
<b>HmeLevel0TotalSearchAreaHeight</b>	<b>-hme-tot-l0-h</b>	[1 - 256]	Depends on input resolution	Total HME Level 1 Search Area in Width
<b>HmeLevel0SearchAreaInWidth</b>	<b>-hme-l0-w</b>	[1 - 256]	Depends on input resolution	HME Level 0 Search Area in Width for each region, separated in spaces, the number of input search areas must equal to NumberHmeSearchRegionInWidth, and the sum must equal to HmeLevel0TotalSearchAreaWidth
<b>HmeLevel0SearchAreaInHeight</b>	<b>-hme-l0-h</b>	[1 - 256]	Depends on input resolution	HME Level 0 Search Area in Height for each region, separated in spaces, the number of input search areas must equal to NumberHmeSearchRegionInHeight , and the sum must equal to HmeLevel0TotalSearchAreaHeight

<b>HmeLevel1SearchAreaInWidth</b>	<b>-hme-l1-w</b>	[1 - 256]	Depends on input resolution	HME Level 1 Search Area in Width for each region, separated in spaces, the number of input search areas must equal to NumberHmeSearchRegionInWidth
<b>HmeLevel1SearchAreaInHeight</b>	<b>-hme-l1-h</b>	[1 - 256]	Depends on input resolution	HME Level 1 Search Area in Height for each region, separated in spaces, the number of input search areas must equal to NumberHmeSearchRegionInHeight
<b>HmeLevel2SearchAreaInWidth</b>	<b>-hme-l2-w</b>	[1 - 256]	Depends on input resolution	HME Level 2 Search Area in Width for each region, separated in spaces, the number of input search areas must equal to NumberHmeSearchRegionInWidth
<b>HmeLevel2SearchAreaInHeight</b>	<b>-hme-l2-h</b>	[1 - 256]	Depends on input resolution	HME Level 2 Search Area in Height for each region, separated in spaces, the number of input search areas must equal to NumberHmeSearchRegionInHeight
<b>LookAheadDistance</b>	<b>-lad</b>	[0 - 120]	17	When Rate Control is set to 1 it's best to set this parameter to be equal to the Intra period value (such is the default set by the encoder)
<b>SceneChangeDetection</b>	<b>-scd</b>	[0 - 1]	1	Enables or disables the scene change detection algorithm
<b>AsmType</b>	<b>-asm</b>	[0 - 1]	1	Assembly instruction set (0: Automatically select lowest assembly instruction set supported, 1: Automatically select highest assembly instruction set supported,)
<b>UseRoundRobinThreadAssignment</b>	<b>-rr</b>	[0 - 1]	0	For Dual socket systems running a Windows* OS on systems with > 32 physical processors. When enabled, allows the encoder to run on both sockets
<b>ReconFile</b>	<b>-o</b>	any string	null	Recon file path. Optional output of recon.
<b>TargetSocket</b>	<b>-ss</b>	[0-1]	1	For Windows based dual socket systems, this can specify which socket the encoder should start / run on (depending on whether UseRoundRobinThreadAssignment is set to 1 or 0) 0= Socket 0, 1=Socket 1 )
<b>ImproveSharpness</b>	<b>-sharp</b>	[0-1]	0	Improve sharpness (0= OFF, 1=ON )

## Legal Disclaimer

Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice Revision #20110804

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request. No product or component can be absolutely secure.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

Intel, Intel Xeon, Intel Core, the Intel logo and others are trademarks of Intel Corporation and its subsidiaries in the U.S. and/or other countries.

\*Other names and brands may be claimed as the property of others.

Copyright 2019 Intel Corporation.