

フレーム欠落について

本書について

SilkyEvCam BothView には、Pythonベースのサンプルコードが付属しています。その中の「2 動画録画」機能において、フレームカメラ側 MP4ファイルのフレーム欠落が起きることがあります。その原因と対策について説明します。

本書の構成

本書は、次の内容で構成されています。

1. 発生原因
2. フレームカメラからの出力時に発生するフレーム欠落
3. MP4ファイル生成時に発生するフレーム欠落
4. MP4ファイル生成時の補足説明
5. フレーム欠落時の対応
6. フレームレートの設定方法



フレーム欠落について

1. 発生原因

MP4ファイルのフレーム欠落は、サンプルコード上、大きく2か所で発生する可能性があります。

- フレームカメラからの出力時
- MP4ファイル生成時

いずれの場合も、フレームカメラに設定したフレームレートを、実行PCが処理しきれず遅延が発生した際に起こります。次項より、それぞれの発生時の確認方法を記述します。（しかしながら、MP4ファイル生成時にフレーム欠落が発生することは、ほとんどないと考えられます（理由は後述）。）

2. フレームカメラからの出力時に発生するフレーム欠落

サンプルコードでは、「フレームカメラ出力時における、フレーム欠落のログテキストファイル出力」を行う事で、どのフレームが欠落したかを後から検知できるようにしています。

ログファイル名： `recording_[YYYYMMDD_hhmmss_sss]_lost-cam-frame.txt`

※ ハイライト部は、他録画ファイル名と同値です。

※ フレーム欠落が起きなかった場合、当ファイルは作成されません。

以下のキャプチャのように、欠落したフレーム情報を1行ごとに出力します。

※ (Frameの)id は 0 からの開始です。

※ status はほとんどの場合「Incomplete」となります。その他の欠落に関する status は2種類ありますが、発生条件が限られているため、使用されることはまれです。

参考URL：

https://docs.alliedvision.com/Vimba_X/VmbPy_Function_Reference/Frame.html?utm_source=chatgpt.com#vmbpy.FrameStatus

```
Missing frame... frame info: Frame(id=422, status=FrameStatus.Incomplete, buffer=0x7f1001f6e910)
Missing frame... frame info: Frame(id=423, status=FrameStatus.Incomplete, buffer=0x7f10021ad520)
Missing frame... frame info: Frame(id=424, status=FrameStatus.Incomplete, buffer=0x7f10023ec130)
Missing frame... frame info: Frame(id=425, status=FrameStatus.Incomplete, buffer=0x7f100262ad40)
Missing frame... frame info: Frame(id=426, status=FrameStatus.Incomplete, buffer=0x7f1002869950)
Missing frame... frame info: Frame(id=431, status=FrameStatus.Incomplete, buffer=0x7f1003164990)
Missing frame... frame info: Frame(id=432, status=FrameStatus.Incomplete, buffer=0x7f10033a35a0)
Missing frame... frame info: Frame(id=433, status=FrameStatus.Incomplete, buffer=0x7f10035e21b0)
Missing frame... frame info: Frame(id=437, status=FrameStatus.Incomplete, buffer=0x7f1003c9e5e0)
Missing frame... frame info: Frame(id=438, status=FrameStatus.Incomplete, buffer=0x7f1004000030)
Missing frame... frame info: Frame(id=439, status=FrameStatus.Incomplete, buffer=0x7f100423ec40)
```



フレーム欠落について

3. MP4ファイル生成時に発生するフレーム欠落

MP4 作成時の ffmpeg エンコードログでは、最終（または途中経過）の統計情報として、欠落数は出力されますが、具体的な欠落フレームの ID は出力されません。そのためサンプルコードでは、最終の統計情報ログのみを出力しています。

ログファイル名： `recording_[YYYYMMDD_hhmmss_sss]_ffmpeg-final-stats.txt`

※ ハイライト部は、他録画ファイル名と同値です。

※ フレーム欠落が起きなかった場合でも、当ファイルは作成されます。

以下のキャプチャのように、エンコード処理の最終統計情報を 1 行で表示しています

```
[ffmpeg log] frame= 1220 fps= 60 q=25.0 Lsize= 46826kB time=00:00:20.32 bitrate=18877.8kbits/s speed= 1x
```

【各項目について】

- frame=n 書き出したフレーム総数
- fps=n 平均エンコード速度
- q=n 量子化パラメータ（QP、画質指標、低いほど高画質）
- Lsize=nKiB 出力ファイルサイズ
- time=HH:MM:SS.ff 出力動画の長さ
- bitrate=n.kbits/s 平均ビットレート（1秒あたりのデータ量）
- speed=n.nnnx 処理速度（リアルタイム比）

フレーム欠落が起きていた場合、以下のキャプチャのように「drop=フレーム欠落数」が表示されます。

```
[ffmpeg log] frame= 431 fps= 25 q=41.0 Lsize= 3028kB time=00:00:17.36 bitrate=1428.8kbits/s dup=0 drop=569 speed= 1x
```



フレーム欠落について

4. MP4ファイル生成時の補足説明

ここでは、「MP4ファイル生成時」にフレーム欠落が起きにくい理由について説明します。

サンプルコードでは、ffmpeg エンコード処理を行う際のパラメータ指定の1つである「vsync」において、「passthrough」を指定しています。

```
ffmpeg_process = (  
    ffmpeg  
    .input('pipe:', format='rawvideo', pix_fmt='bgr24',  
           s='{}x{}'.format(img_w, img_h), use_wallclock_as_timestamps=1, fflags='+genpts')  
    .crop(g_value.img_trim_offset_x, g_value.img_trim_offset_y, g_value.img_trim_width, g_value.img_trim_height)  
    .output(file_path, pix_fmt='yuv420p', vcodec=v_codec, vsync='passthrough',  
            preset='ultrafast', bf='0', tune='zerolatency', loglevel="info", stats=None) # loglevel-default:info  
    .run_async(pipe_stdin=True, pipe_stderr=True, overwrite_output=True) # Display log on Text  
)
```

この指定は、入力フレームをすべてそのまま受け付ける設定です。そのため、他の指定「cfr（固定フレームレート）」や「vfr（可変フレームレート）」のように、エンコード処理能力や同期の都合でフレームを削除することはありません。その分エンコード処理の遅延が生じやすくなりますが、それ以前に「フレームカメラからの出力時点」で、先にフレーム欠落が起きることが多いと考えられます。



フレーム欠落について

5. フレーム欠落時の対応

「フレームカメラ出力時」や「MP4ファイル生成時」にフレーム欠落が発生した場合は、実行 PC の処理リソース不足が原因と考えられるため、指定フレームレートの見直しが必要です。

「フレームカメラ出力時」のみでフレーム欠落が発生した場合は、残存フレームと対応する外部イベントトリガーを照合することで、残存フレームに対するイベントデータとの同期が可能です。

※ フレーム欠落が起きた場合でも、フレームカメラのトリガー信号は出力されます。

「MP4ファイル生成時」にフレーム欠落が発生した場合は、ログファイルの最終統計情報「speed」の値を確認してください。0.99x 以上の場合は、リアルタイム（1.0x）にほぼ一致しており、大きな問題はありません。

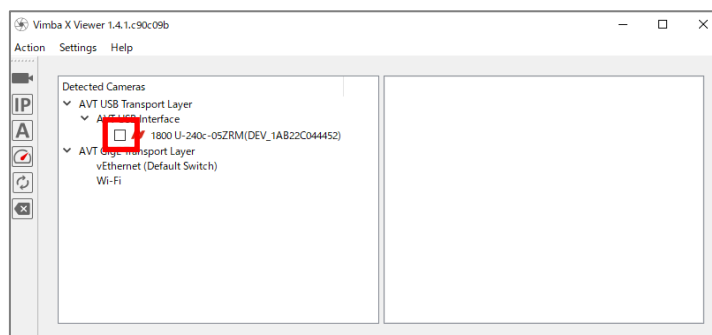
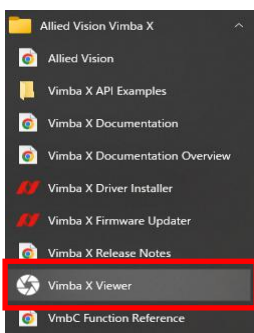
一方、0.90x 以下など明らかに処理速度が入力バッファに追いついていない場合は、実行 PC の処理リソース不足と考えられるため、指定フレームレートの見直しが必要です。（録画が短時間であればフレーム欠落が発生しない場合もありますが、長時間の録画では、いずれバッファオーバーフローとなりフレーム欠落が発生します。）

6. フレームレートの設定方法

サンプルコード一式には、フレームカメラの設定を記述しているファイル「settings.xml」が含まれています。必要に応じて設定内容を変更してください。以下に変更方法をご説明します。

1. Vimba X Viewerの起動

BothView を PC に接続し、「Vimba X Viewer」を起動します。

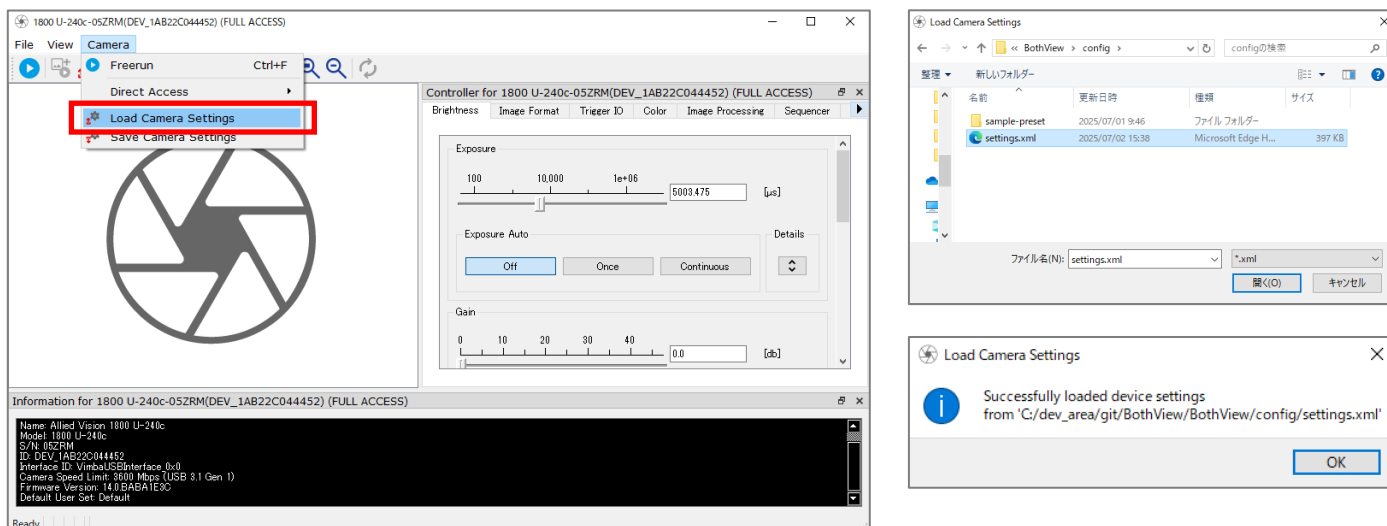


フレーム欠落について

6. フレームレートの設定方法（つづき）

2. 設定ファイルの読み込み

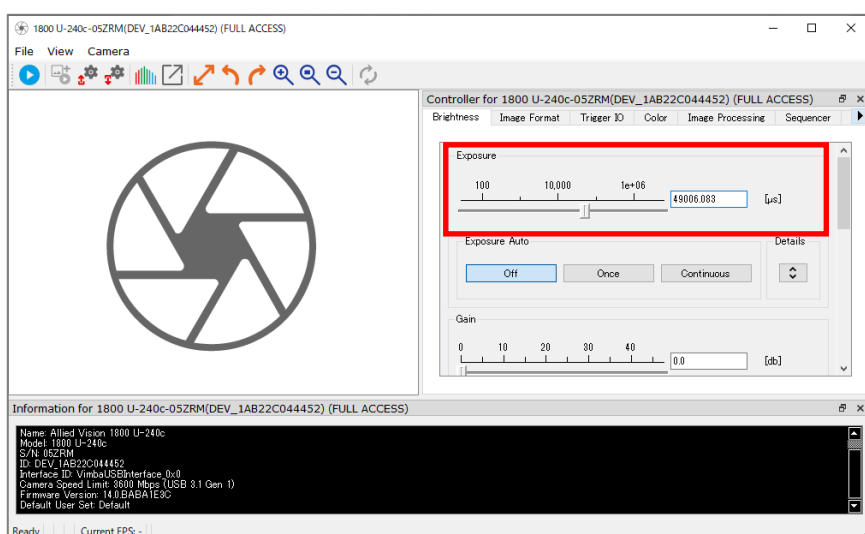
Vimba X Viewer が起動したら、メニューバーの「Camera」 - 「Loading Camera Settings」を押下し、サンプルコードの config フォルダ内にある既存の設定ファイル (settings.xml) を選択し読み込みます。



3. 露光時間 (Exposure Time) の変更

ビューアの右ペイン、「Brightness」タブ内の「Exposure」の値を大きくします。

配布時の初期設定では、60fpsで動作するように、読み出し (readout) およびオーバーヘッド (idle) を除いた露光時間として 15,006.32 μ s が設定されています。



参考URL(PDF)：

https://cdn.alliedvision.com/fileadmin/content/documents/products/cameras/Alvium_USB/techman/Alvium-USB-Cameras_User-Guide.pdf

Rev. 1.0



フレーム欠落について

6. フレームレートの設定方法（つづき）

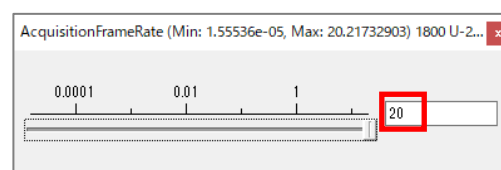
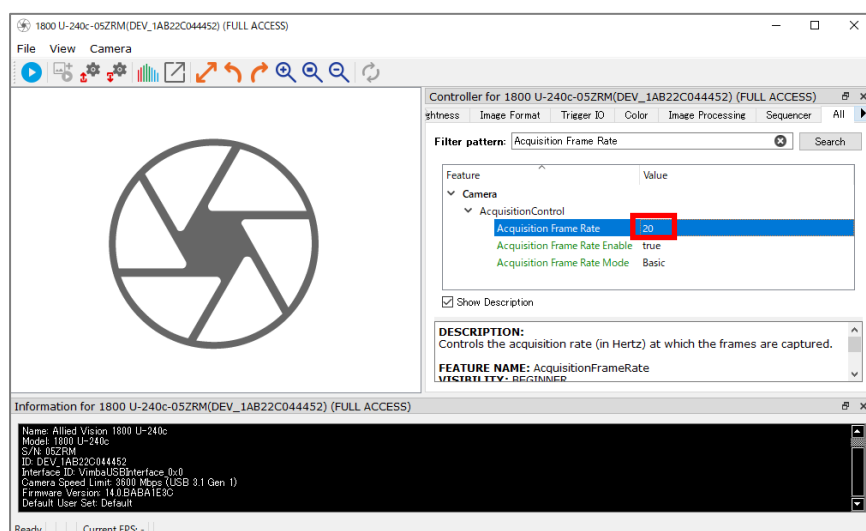
4. フレームレート (Acquisition Frame Rate) の変更

ビューアの右ペイン、「All」タブ内の「Filter pattern」に「Acquisition Frame Rate」を入力し、Search ボタンを押下します。リストに表示された「Acquisition Frame Rate」の Value をクリックすると、設定ウィンドウが表示されます。その値を少なくしてください。

※ サンプルコード配布時の初期設定は 60fps です。

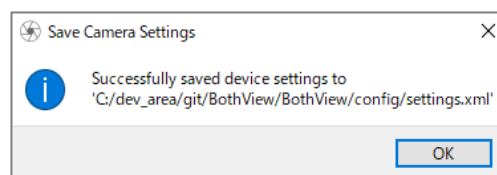
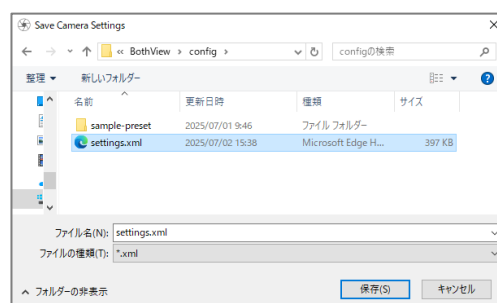
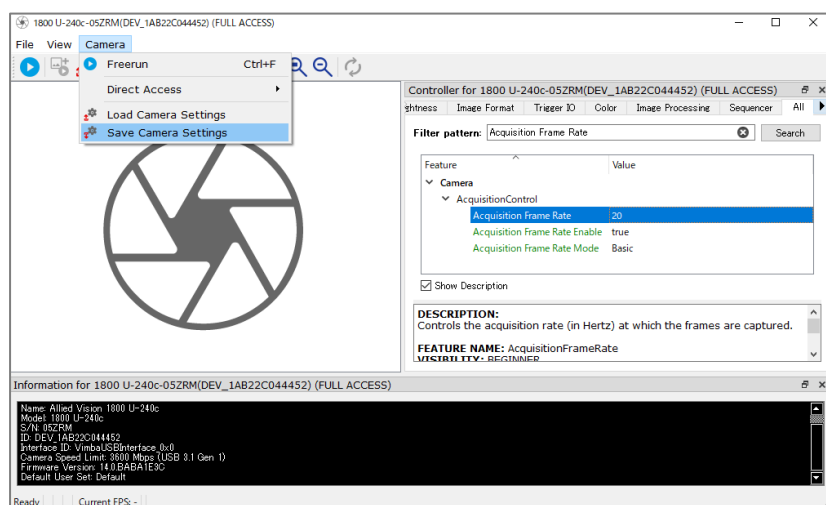
※ 露光時間 (Exposure Time) はフレームレート (Acquisition Frame Rate) より優先されます。

露光時間が、設定されたフレームレートの逆数 (1 フレームに使える時間) より長くなると、実際のフレームレートはそれに応じて自動的に下がります。



5. 設定ファイルの保存

メニューバーの「Camera」 - 「Saving Settings」を押下し、設定ファイルを保存します。その際、カメラのストリーミングを必ず停止してから保存してください。



Rev. 1.0



フレーム欠落について

6. フレームレートの設定方法（つづき）

6. サンプルコードでの動作確認

サンプルコードを実行し、被写体を2動画録画します。作成されたログファイルの内容から、フレーム欠落が発生していないことを確認してください。

