

## 1 目的

これまでのビューワは、実験環境において約 12.5FPS のフレームレートでしかコンテンツを表示することができなかった。そこで、このビューワのフレームレートの向上を試みた。

## 2 プログラム修正点

フレーム画像の切り出しと色空間の変換を既に済ませたファイルを入力とすることで、ビューワ側からこれらの処理を取り除いた。変更前と変更後の入力ファイルの違いを表 1 に示す。

表 1: 入力ファイル仕様

	動画形式	色空間	フレーム画像
変更前	AVI ファイル (MotionJPEG)	RGB	5120 × 480
変更後	独自形式 (mvv)	YUY2	640 × 480 が 8 枚

変更前後のビューワの 1 ループの処理の流れを図 1 及び図 2 に示す。

1. AVI ファイル読み込み
2. スレッドを 8 つ生成
  - (a) 5120 × 480 のフレーム画像から、各スレッドの担当する 640 × 480 の画像を切り出す。
  - (b) 切り出し画像の色空間を RGB から YUY2 に変換
3. スレッドをジョイン
4. 8 つのフレーム画像を XvExt の関数でデバイスに転送

図 1: 変更前の処理

1. mvv ファイル読み込み
2. 8 つのフレーム画像を XvExt の関数でデバイスに転送

図 2: 変更後の処理

## 3 実験

実験に用いたマシン (sidamo) の環境を表 2 に示す。また、プロジェクタには、LED プロジェクタ (AD-MP15A) を 8 台用いた。

FPS の測定結果を表 3 に示す。FPS は 150 フレーム分の平均値として算出した。このように、解像度 800 × 600 であっても、30FPS で動作させることに成功しており、パフォーマンスの改善が行えたといえる。

表 2: 実験環境

OS	Debian/GNU Linux 5.0.3(lenny)
CPU	Intel Xeon E5520 2.27GHz × 2
ハイパースレッディング設定	有効
メモリ	16GB
グラフィックボード	nVidia Quadro NVS 440 × 2 + nVidia Quadro NVS 295

表 3: 実験結果

フレーム画像解像度	プロジェクタ解像度	改良前	改良後
640 × 480	640 × 480	12.5	51 ~ 59
640 × 480	800 × 600	13.1	43.0
800 × 600	640 × 480	8.43	29.8
800 × 600	800 × 600	8.72	30.0

## 4 問題

今回の修正によって、処理速度の向上には成功したものの、コンテンツファイルのサイズが肥大化してしまっている。そのため、長い動画を扱う場合は、運用上の問題が発生する可能性がある。

表 4 は、プロジェクタ数 8、フレームレート 30FPS で 2 分間の動画の場合、それぞれの解像度でのファイルサイズがどうなるかを示したものである。

表 4: コンテンツファイルのサイズ

フレーム画像解像度	ファイルサイズ (GB)
640 × 480	17
800 × 600	27
1024 × 768	45