

画像を表現する二種類の方法 — ラスタ画像とベクトル画像 と ffmpeg を使った動画の作成と編集

高橋芳幸

神戸大学大学院理学研究科惑星学専攻

2021 年 9 月 3 日

前置き

- 本日の話は二本立てです.
 - 画像を表現する二種類の方法
 - ラスタ画像とベクトル画像
 - ffmpeg を使った動画の作成
- ふたつに繋がりは(ほぼ)ありません.

画像を表現する二種類の方法

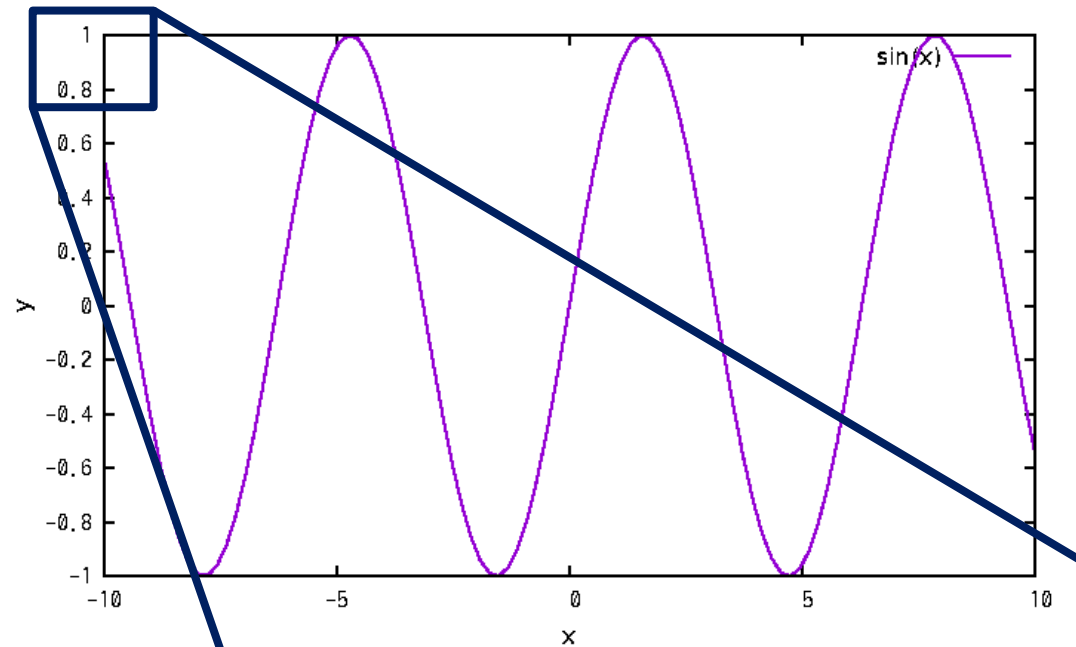
— ラスタ画像とベクトル画像

はじめに

- 計算機で画像を表現する方法は二種類
 - ラスタ画像
 - ベクトル画像
- 今後、学位論文（博士論文，修士論文，学士論文（卒業論文））を執筆するにあたって，このあたりを知らないようでは何なので(?)，簡単に解説する。

ラスタ画像

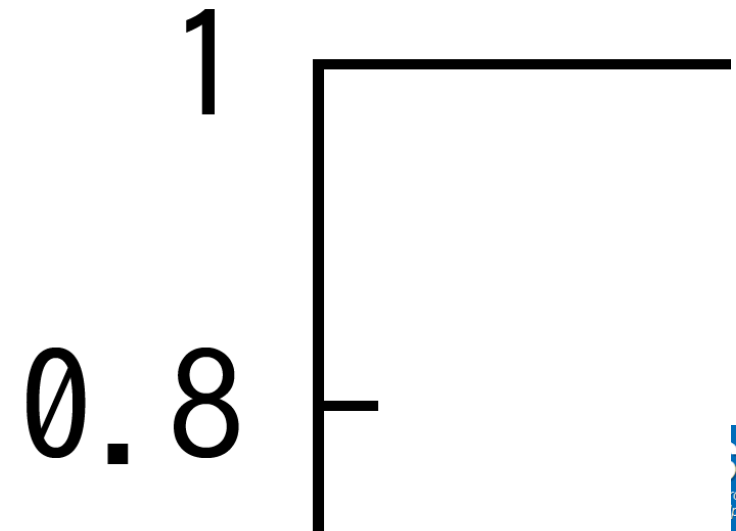
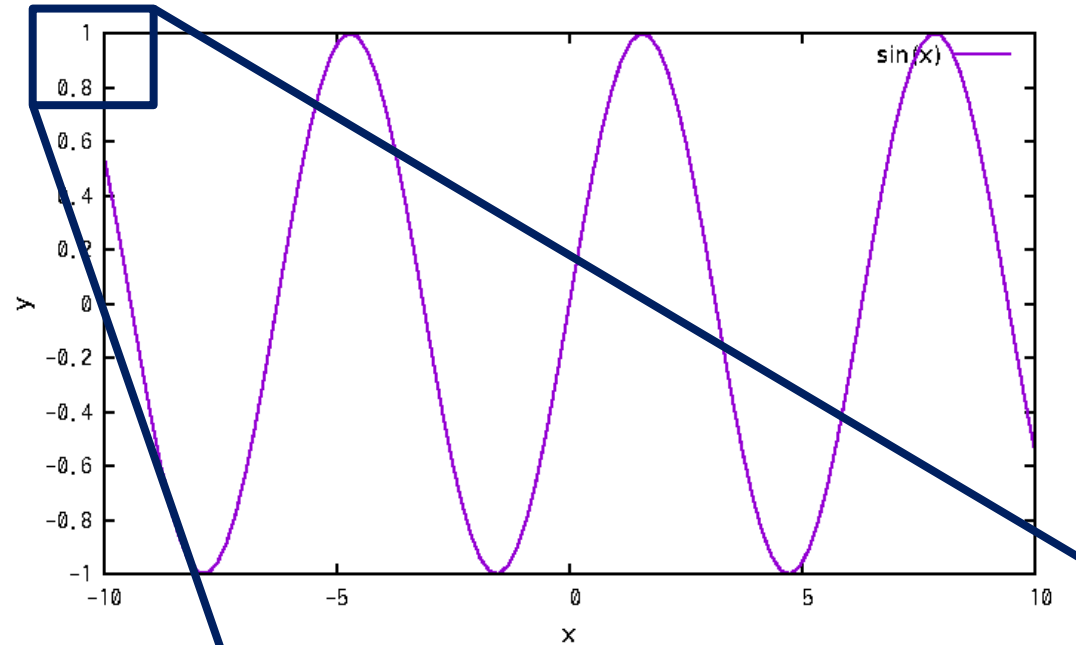
- ラスタ : raster
- 「ビットマップ画像」ともいう
 - ビットマップ : bitmap
- 点の色情報を用いて, 点の集まりとして画像を表現
- 利点
 - 複雑なデータ(写真など)を表示可能
- 欠点
 - 拡大縮小に適さない
- 画像形式
 - bmp, gif, jpg, png, tiff



1
0.8

ベクトル画像

- ベクトル : vector
- 点や線の位置, 文字, それらの色などの情報を用いて画像を表現
- 利点
 - サイズ変更(拡大縮小)に適している
- 欠点
 - 複雑なデータ(写真など)を表示するのは難しい
- 画像形式
 - ps, eps, pdf, svg
 - pdf はラスタ画像データを含むこともできる



二つの画像表示方式

- ラスタ画像とベクトル画像は使い分けるのが賢い.
 - 写真にはラスタ画像を用いるのが自然
 - 過度な拡大には耐えられない.
 - しかし, 元々それだけの情報しか持っていない.
 - 幾何学図形はベクトル画像で表現する方が良い(と思う)

高橋のやり方

- 科学的な図(グラフ)は基本的にひとまずベクトル画像として保存
 - もちろん写真をベクトル画像にしたりしない
 - 超高解像度の画像をベクトル画像にするかは場合によるかも
- ラスタ画像が必要な時には変換
 - PowerPoint, Word のファイルに貼るとき
 - 提出規定として求められるとき
 - 動画を作る時

高橋のやり方 — ベクトル画像

- 画像フォーマットは(基本) pdf
 - gnuplot も DCL (Denkou Club Library) も pdf で出力可能
 - TeX も pdf を取り込める
- TeX で取り込むときなどに余白が邪魔ならばトリミング
 - pdfcrop
 - pdf ファイルをトリミングしてくれるプログラム
 - 自動的に余白を判定してトリミングしてくれるので便利
 - debian ならば texlive-extra-utils パッケージに含まれている

高橋のやり方 — ラスタ画像

- (Linux 上で) pdf ファイルをラスタ画像へ変換
 - convert (mogrify), pstoimg
 - 変換時の解像度(画像サイズ)に注意
- あるいは, Windows 上のソフトウェアにペーストするならば, adobe acrobat で開いて「スナップショット」でコピーしてペースト

高橋のやり方 — ラスタ画像

- convert (複数ファイルならば mogrify)
 - pdf から png, gif などへ変換
 - debian なら imagemagick パッケージに含まれている
- pstoimg
 - pdf から png, gif へ変換
 - debian なら latex2html パッケージに含まれている
 - convert の方が有名だが, 経験的には pstoimg が生成するファイルの方がサイズが小さい

ここまでのまとめ

- 画像の表示形式には二種類ある: ラスタ画像, ベクトル画像
 - ラスタ画像は点の集合を用いて画像を表現
 - ベクトル画像は点と線の位置と色を用いて画像を表現
- 用途に合わせて使い分けましょう
- ちなみに高橋は,
 - 科学的な図(グラフ)は(基本)ベクトル画像(pdf)
 - 必要ならばラスタ画像に変換

ffmpeg を使った動画の作成

はじめに

- 数値計算していると、計算結果の動画を作りたくなることがある。
- 手軽な動画形式は gif アニメ形式と思うが、他の形式の方が便利かもしれない。
 - gif アニメ形式では途中停止/再生速度調節できない。
 - xanim では途中停止できたのだが...
- さらに、動画を作ると、動画を編集したくなることもあるだろう。
- 今回は、ffmpeg を使った動画の作成について紹介する。
- さらに、ffmpeg を使った動画の編集方法についても紹介する。

ffmpeg でできることの例 (高橋が経験したこと)

- 静止画から動画の作成
- 動画の情報確認
- 動画のトリミング
 - 時間方向
 - 空間 (x-y) 方向
- 動画のサイズ変更
- 動画の解像度変更
- 動画を並べる
 - 左右
 - 上下
 - タイル状
- 動画の中の小窓に動画を表示

ffmpeg : 動画の情報確認

- input.avi の情報を確認する

```
$ ffmpeg -i input.avi
```

```
ffmpeg version 4.1.6-1~deb10u1 Copyright (c) 2000-2020 the FFmpeg developers
```

```
(中略)
```

```
Input #0, avi, from 'anim.avi':
```

```
Metadata:
```

```
encoder      : Lavf58.20.100
```

```
Duration: 00:00:04.00, start: 0.000000, bitrate: 164265 kb/s
```

```
Stream #0:0: Video: mpeg4 (Simple Profile) (FMP4 / 0x34504D46), yuv4
```

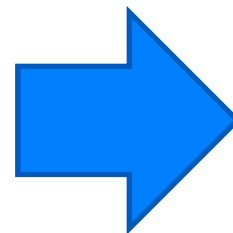
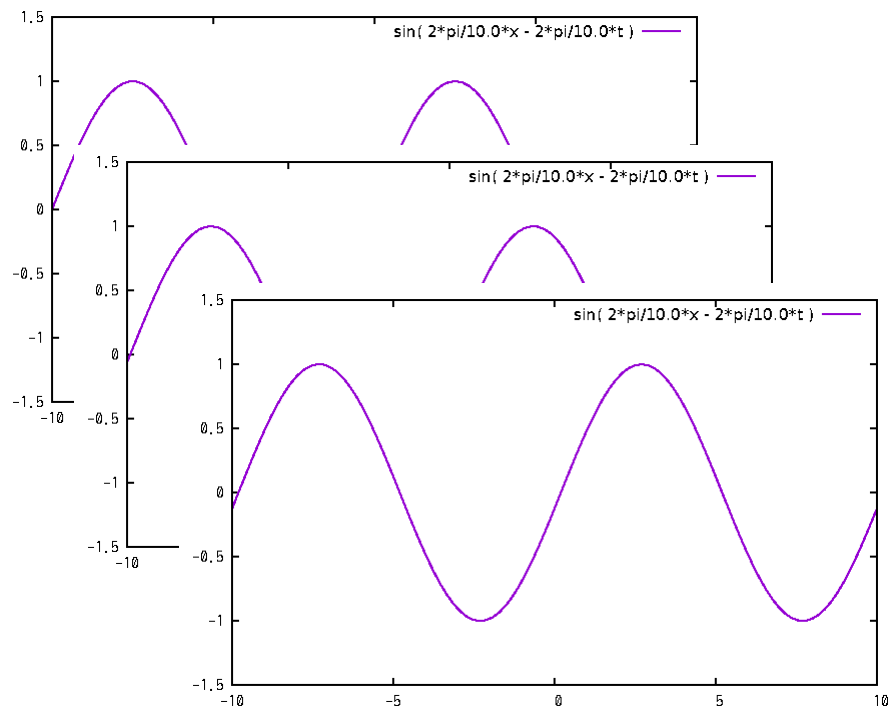

ffmpeg : 静止画から動画の作成

- 静止画ファイルがあるとする
 - fig_000001.png, fig_000002.png, fig_000003.png, ...
- これらをまとめて動画ファイル out.avi を作成する

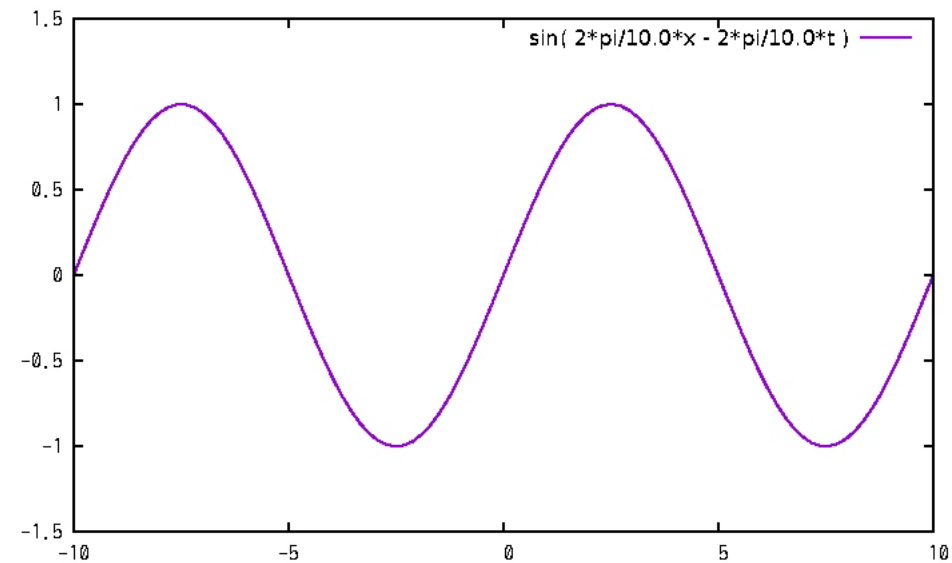
```
$ ffmpeg -i fig_%06d.png out.avi
```

ffmpeg : 静止画から動画の作成

静止画 (png)



動画 (avi)



ffmpeg : 動画を左右/上下に並べる

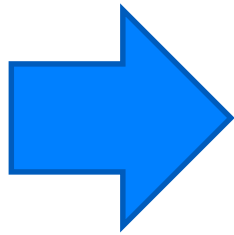
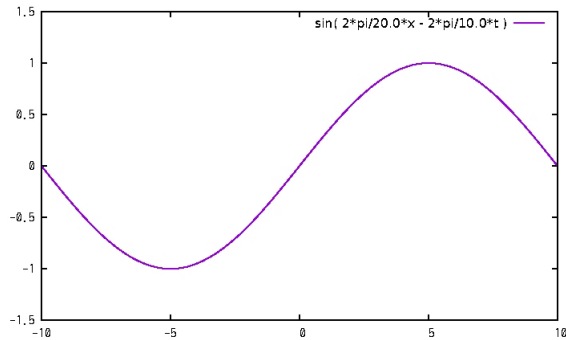
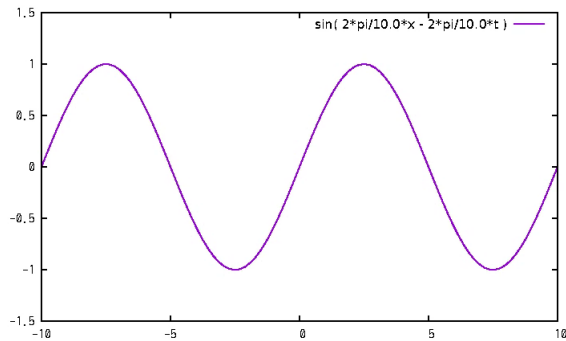
- left.avi と right.avi を左右に並べて out.avi を作成する

```
$ ffmpeg -i left.avi -i right.avi -filter_complex hstack out.avi
```

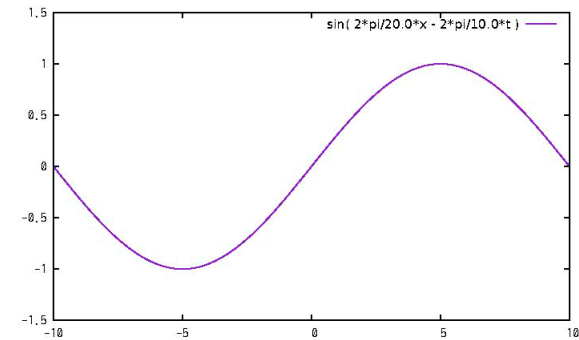
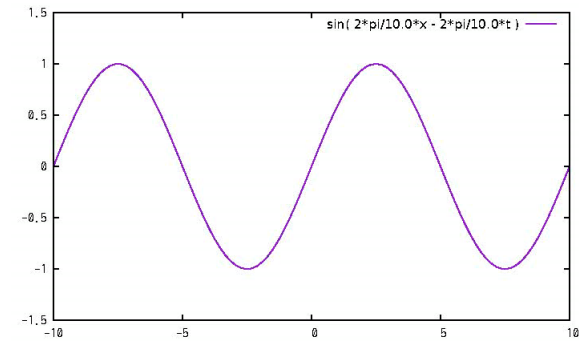
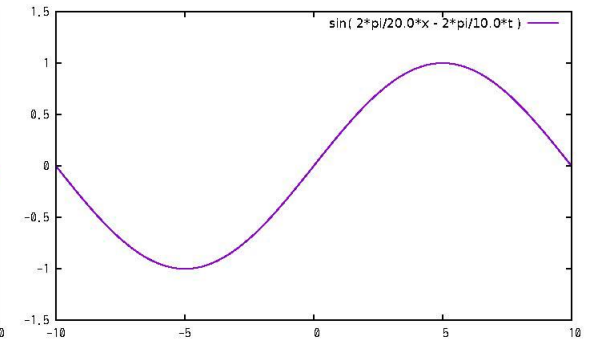
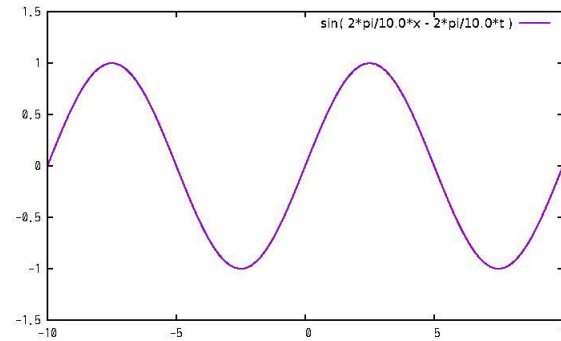
– 縦に並べるには hstack を vstack にする(たぶん)

ffmpeg : 動画を左右/上下に並べる

動画 2 つ (avi)



動画 1 つ (avi)



ffmpeg : 動画を 2 × 2 に並べる

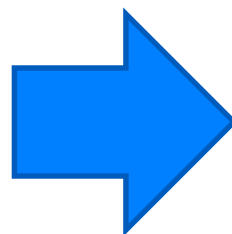
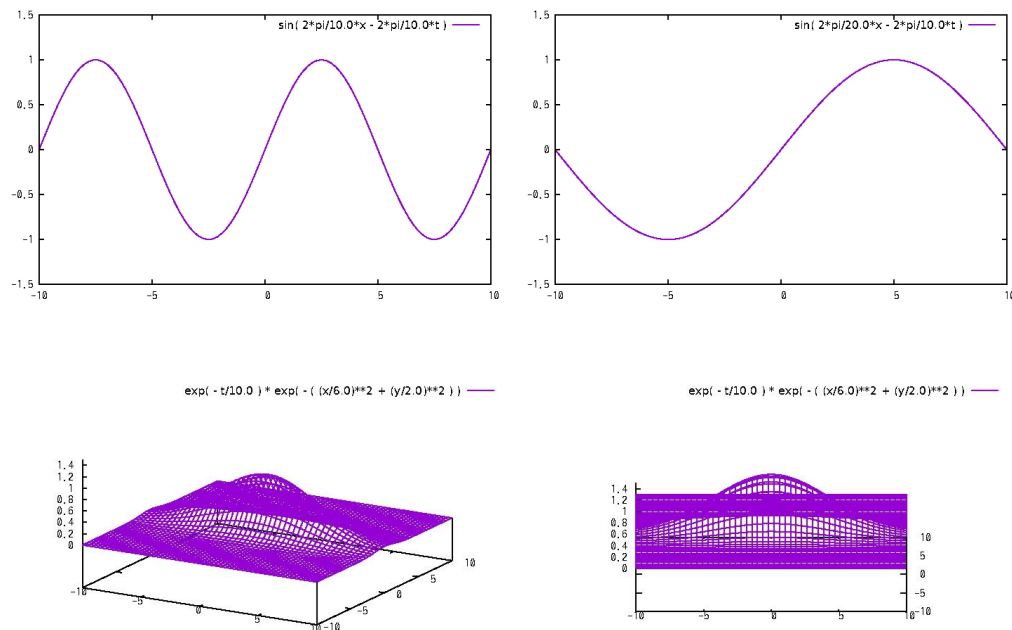
- movieXX.avi を 2 行 2 列に並べて out.avi を作成する

```
$ ffmpeg ¥  
-i movie11.avi ¥  
-i movie12.avi ¥  
-i movie21.avi ¥  
-i movie22.avi ¥  
-filter_complex ¥  
"[0:v][1:v]hstack=inputs=2[top];[2:v][3:v]hstack=inputs=2[bottom];[top][bottom]vstack=inputs=2[v]" ¥  
-map "[v]" ¥  
movie_all.avi
```

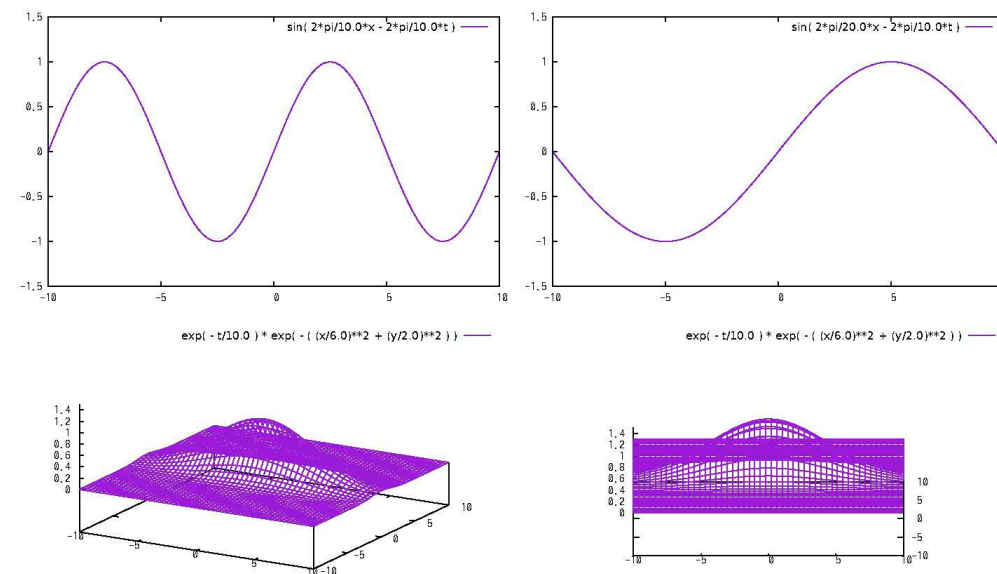
– 規則の詳細をわかっていないが気分はわからんこともない

ffmpeg : 動画を 2 × 2 に並べる

動画 4 つ (avi)



動画 1 つ (avi)



ffmpeg : 動画の小窓を埋め込む

- parent.avi に小さな child.avi を重ねて out.avi を作成する

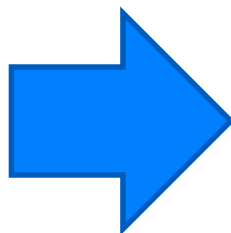
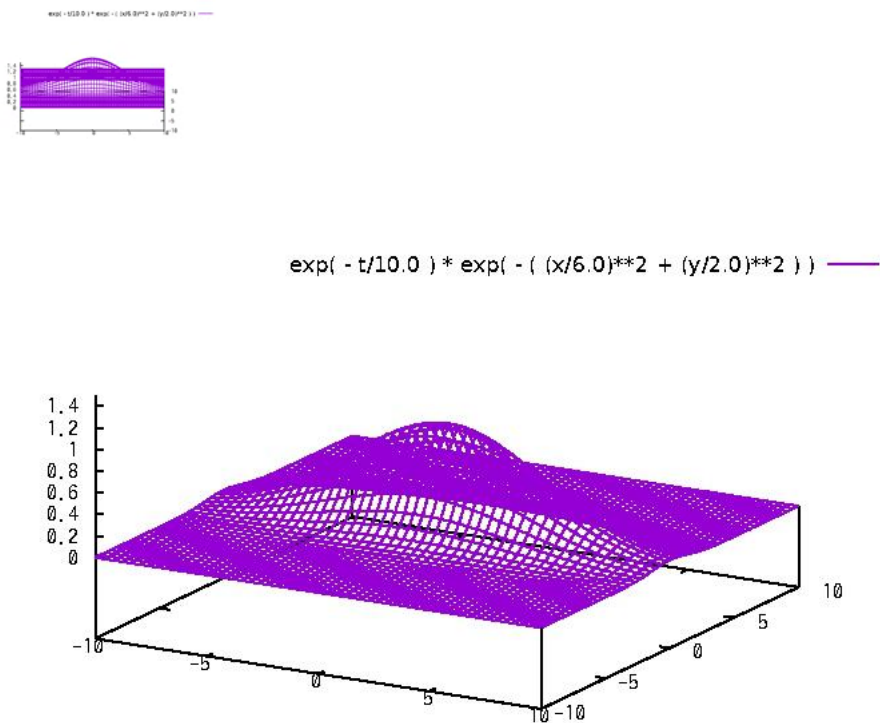
```
$ ffmpeg -i parent.avi ¥
```

```
-vf "movie=child.avi[inner]; [in][inner] overlay=910:440 [out]" ¥  
out.avi
```

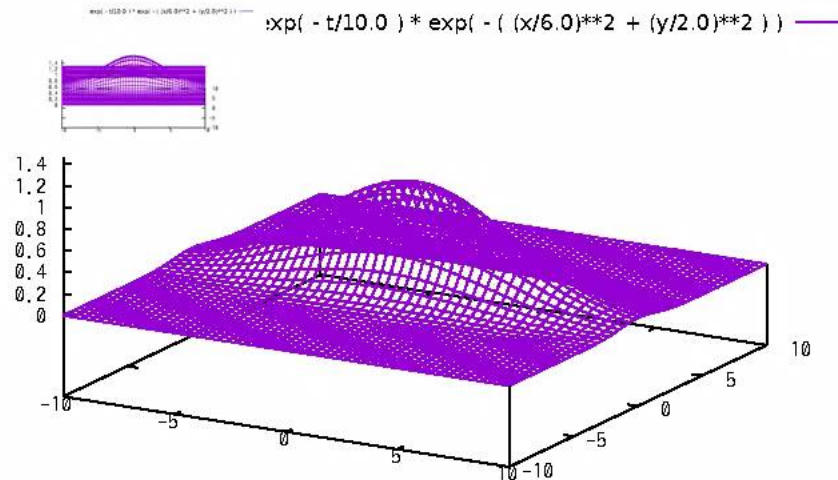
- overlay は小窓を入れる位置.
- (おそらく) 910 が左からの距離. 440 が上からの距離.
- 規則の詳細をわかっていないが気分はわからんこともない

ffmpeg : 動画の小窓を埋め込む

動画 2 つ (avi; 親 + 小)



動画 1 つ (avi)

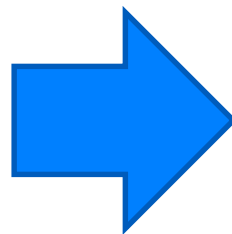
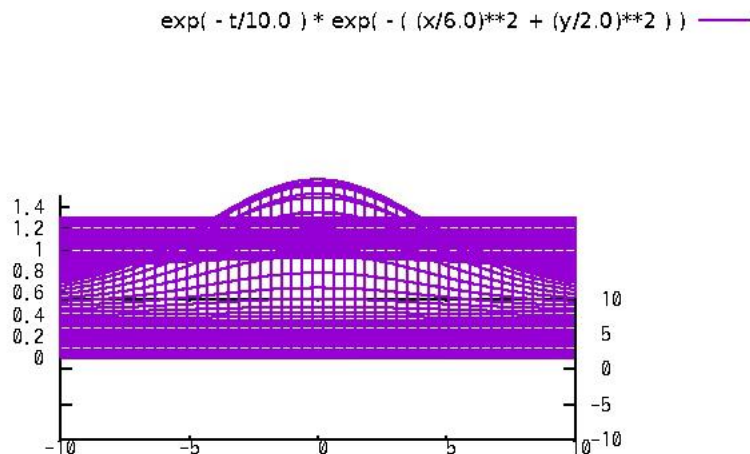


ffmpeg : 動画のサイズ変更

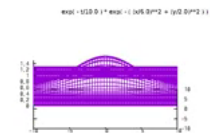
- org.avi を小さくして small.avi (横幅 200) を作成する

```
$ ffmpeg -i org.avi -vf scale=200:-1 small.avi
```

元動画



小さくした動画



後半のまとめ

- ffmpeg は動画の作成と編集に大変有用.
 - 動画の作成
 - 動画の情報確認
 - 動画のトリミング, サイズ変更, 解像度変更
 - 動画を並べる
 - 動画の中の小窓に動画を表示
- 別々の動画を並べて一つの動画にするだけでも比較が容易になる.
- 使い方は, 検索すると色々見つけられます.

議論

- 皆さんは何を使って動画を作っていますか？
 - convert, gifmerge, gifsicle (gif アニメ?)
- 他には？

まとめ

- 画像の形式について紹介した.
 - ラスタ画像
 - ベクトル画像
- ffmpeg を用いた動画の作成方法, 編集方法について紹介した.
- 今後, 適した方法を用いて見やすい図を作ってください.