

# 画像の diff を Neovim で表示する

---

2024/10/30 ゴリラ.vim #33

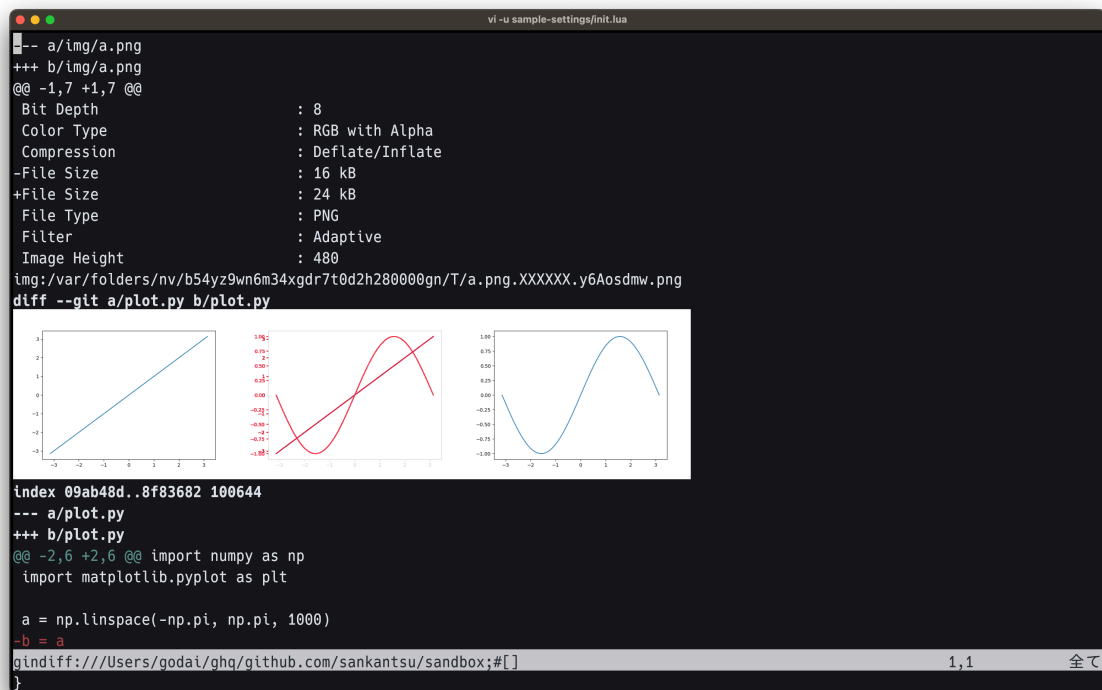
# 自己紹介

- 名前: sankantsu
- エディタ歴: Vim 5 年 → Neovim 1.5 年
- 趣味: 麻雀, Speedcubing, 競プロ…



# 何の発表？

- Neovim のバッファ内で画像の diff を表示できる機能をつかったよ！
- つくったもの
  - gin-diff-image.nvim: <https://github.com/sankantsu/gin-diff-image.nvim>

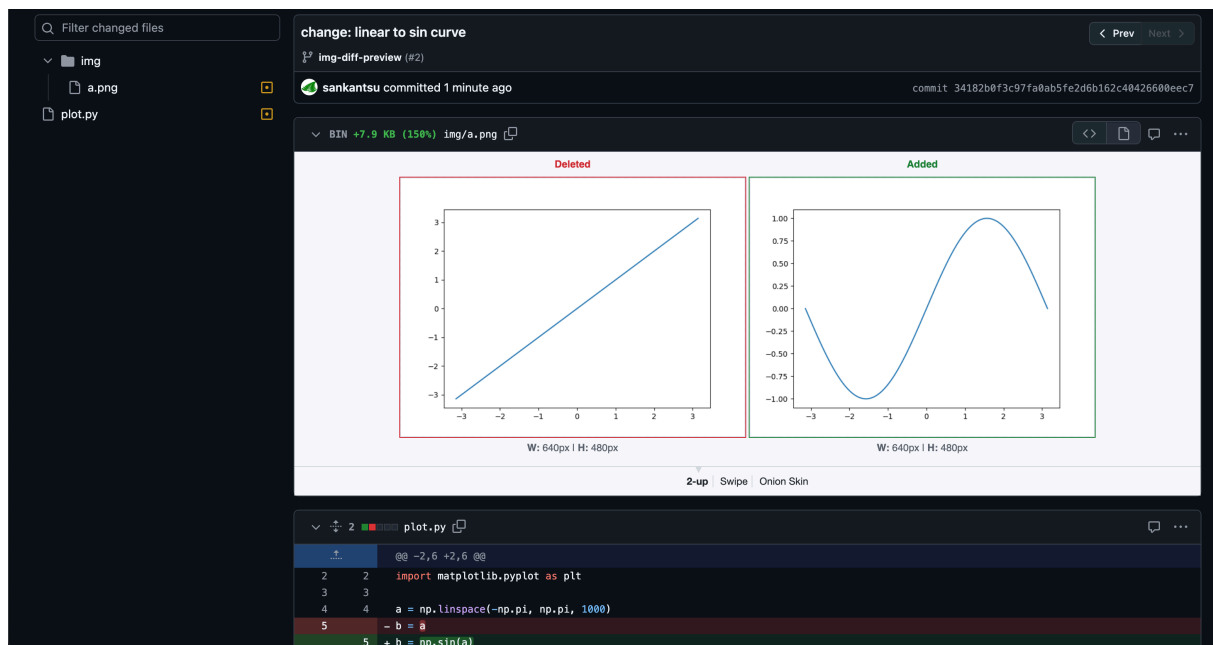


```
vi -u sample-settings/init.lua
-- a/img/a.png
+++ b/img/a.png
@@ -1,7 +1,7 @@
  Bit Depth           : 8
  Color Type          : RGB with Alpha
  Compression         : Deflate/Inflate
  -File Size          : 16 kB
  +File Size          : 24 kB
  File Type           : PNG
  Filter              : Adaptive
  Image Height        : 480
img:/var/folders/nv/b54yz9wn6m34xgdr7t0d2h280000gn/T/a.png.XXXXXX.y6Aosdmw.png
diff --git a/plot.py b/plot.py
index 09ab48d..8f83682 100644
--- a/plot.py
+++ b/plot.py
@@ -2,6 +2,6 @@ import numpy as np
 import matplotlib.pyplot as plt

 a = np.linspace(-np.pi, np.pi, 1000)
-b = a
+
gindiff:///Users/godai/ghq/github.com/sankantsu/sandbox;#[]
}
```

# きっかけ

- Github の画像 diff 便利じゃない?
  - ▶ 特にドキュメント類を git 管理する際重宝する

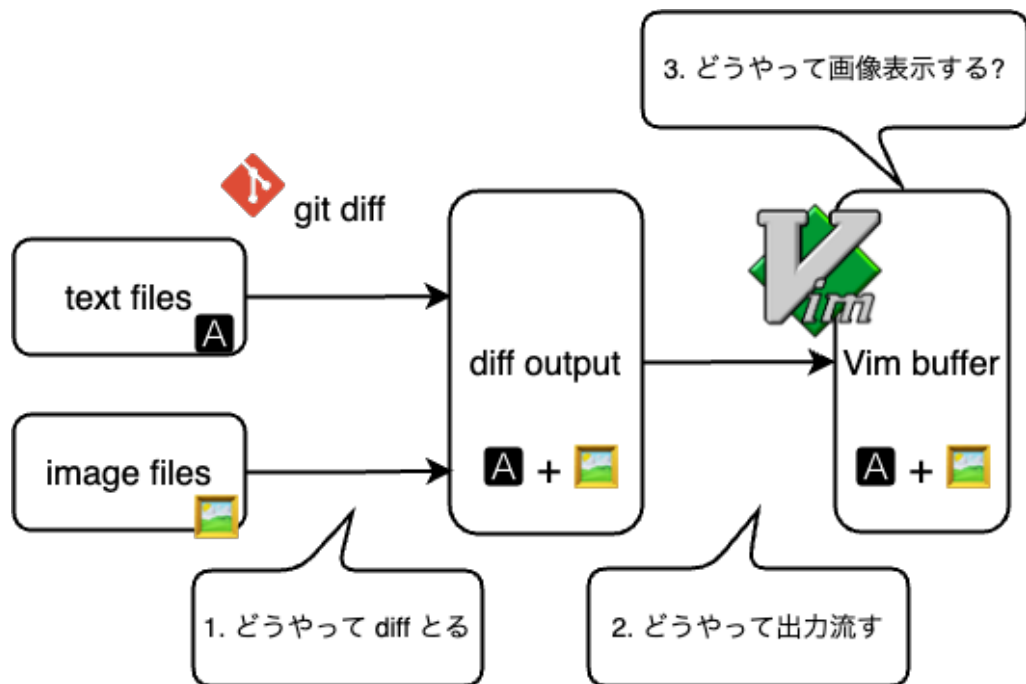


- Github に上げなくても見れたら便利そう

- ターミナル上で `git diff` で画像の diff 見れるようにする
- どうせなら :Vim: の中から見たいよね?
  - Vim は高機能な previewer
  - Vim から出ずに作業を完結できる
  - Git 連携用の強力なプラグイン機能を使える
  - (LT ネタが生える)

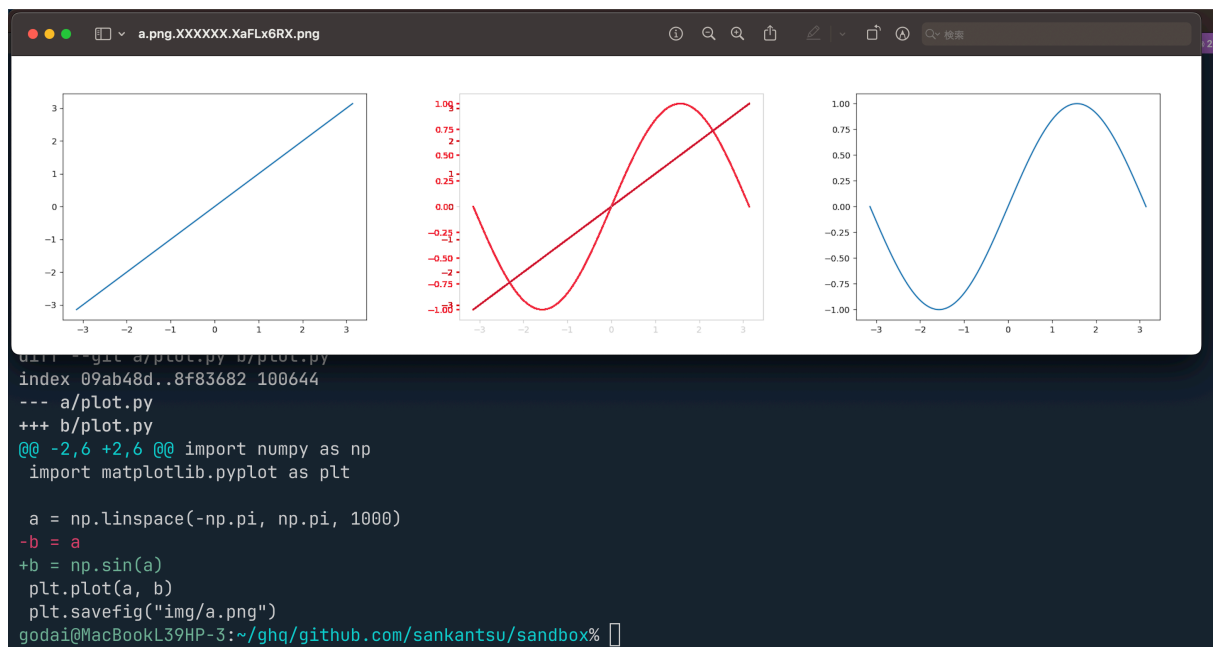
# 技術的課題

- git diff の出力に画像の差分情報を出す
- git diff の出力を Vim に流す
- Terminal/Vim 内部での画像表示



# 1. git diff で画像の差分情報を出す

- まずは git diff で画像の差分についての情報を出せるようにしたい
- 調べたらいい感じのやつあった
  - ▶ git diff-image: <https://github.com/ewanmellor/git-diff-image>
  - ▶ 画像の変更前, 差分, 変更後のプレビューをウィンドウで開いてくれる



# git-diff-image の技術的ポイント

- gitattribute で diff 属性を設定
  - gitattribute は git 管理下のファイルに任意の属性を付加するしくみ
- 設定例

```
# .gitattributes
*.png diff=image # png ファイルの diff 属性をカスタム値に設定

# .gitconfig
[diff "image"]
command = "/path/to/git_diff_image" # カスタムの diff コマンド
```

- 画像ファイルに対してカスタムの diff コマンドを用意
  - imagemagick を使って diff 画像 (一時ファイル) を生成
  - xdg-open などを使って preview window を開く



## git-diff-image だと満たせない部分

- 画像の diff がターミナルとは別のウィンドウで出てくる。
  - ▶ ターミナルからウィンドウのフォーカスが移る。
  - ▶ 変更した画像がたくさんあると、ウィンドウがたくさん出てくる。
  - ▶ 画像とテキストファイルの差分を別々に確認することになる。

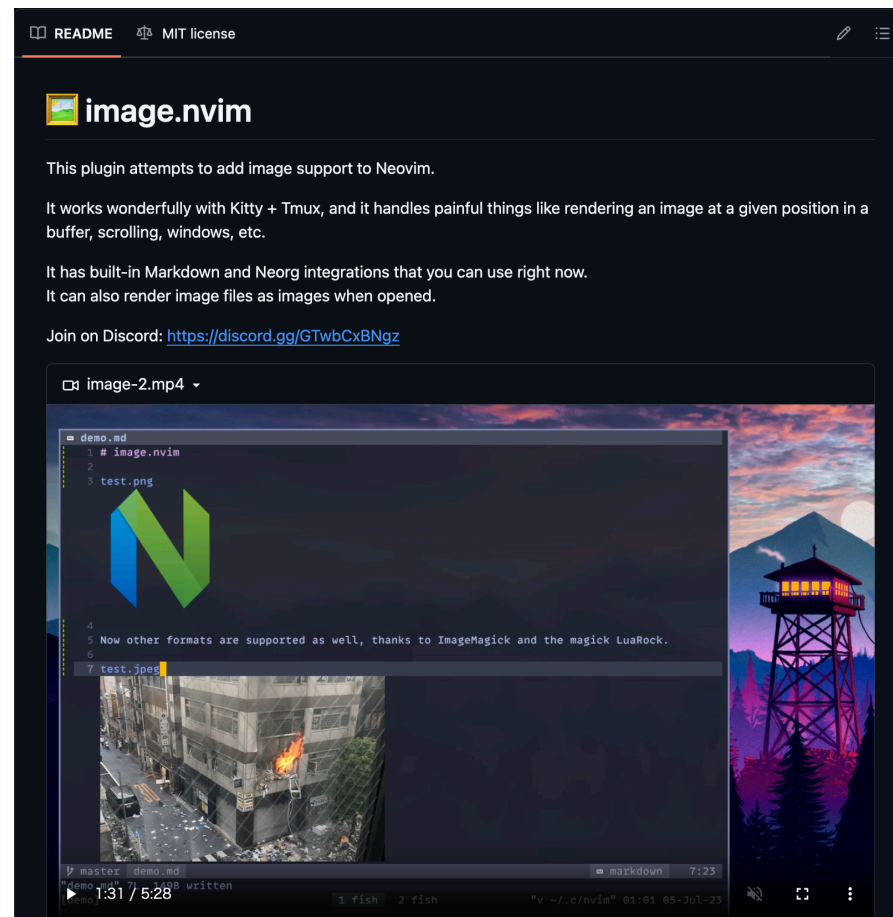
別ウィンドウ表示はターミナル上での作業の中断につながりやすい

## 2. Vim に diff 出力を流す

- GUI ウィンドウを開く代わりに出力を Vim に流す
  - つまり、Vim を pager 代わりに使う
- すでに良いプラグインがある
  - `lambdalisue/vim-gin`: <https://github.com/lambdalisue/vim-gin>
  - `:GinDiff` で Vim の中から diff が見れる
  - diff の行で `<CR>` するとファイル内の該当行に飛べるとかも便利
- あとは `GinDiff` に画像ビューを統合できたら良さそう!
  - テキストの diff の流れの中に画像の diff も出せるとうれしい

### 3. Neovim の中に画像表示

- 端末に生のエスケープシーケンス (e.g. Sixel) 吐けば画像は出る。
  - ▶ Vim: `echoraw()` / Neovim: `chansend()`
  - ▶ しかし、自分で再描画やスクロールを面倒見てやるのはかなり大変...
- 3rd/image.nvim
  - ▶ <https://github.com/3rd/image.nvim>
  - ▶ 画像表示の基本機能を提供
  - ▶ Kitty Graphics Protocol ベース



# vim-gin と image.nvim を連携

- :GinDiff でつくられるバッファの中に image.nvim の機能で画像を表示したい
- image.nvim を新しいファイルタイプに対応させるには...
  - ▶ バッファの中から画像を表示したい場所を探す
  - ▶ 表示したい画像の URL / path を見つける
- 方針: diff の出力に画像 path を埋めこんでいて image.nvim から見つけられるようにする

```
# git diff の出力例
# "gin-diff-image:" prefix で画像 path を埋めこむ
--- a/a.png
+++ b/a.png
gin-diff-image:/var/folders/nv/b54yz9wn6m34xgdr7t0d2h280000gn/T/
a.png.XXXXXX.8RSRBbbn.png
```

## image.nvim 拡張の実装

```
return require("image/utils/document").create_document_integration({
  name = "gindiff",
  default_options = { filetypes = { "gin-diff" }, /* ... */ },
  query_buffer_images = function(buf)
    local images = {}
    -- iterate over all lines in buffer
    local lines = vim.api.nvim_buf_get_lines(0, 0, -1, true)
    for i, line in ipairs(lines) do
      -- find a image tag
      local path = string.match(line, "^gin-diff-image:(.*)")
      if path ~= nil then
        -- create a image to display
        local image = { url = path, range = /* ... */ }
        table.insert(images, image)
      end
    end
    return images
  end,
})
```

# まとめ

- GinDiff のバッファに画像を表示するプラグインができた
  - つくったもの: <https://github.com/sankantsu/gin-diff-image.nvim>
- ポイント
  - カスタム diff driver で diff の出力をいじる
  - image.nvim の integration を書くことで手軽に画像表示対応できる
- 今後の課題
  - diff 画像生成機能を Neovim プラグイン側に移せれば diff driver 不要にできる？
  - image.nvim の Sixel 対応
  - Wezterm の Kitty Graphics Protocol サポート強化

- ターミナル画像表示の良さげな応用が見つかったかも
- 実はそんなにコードは書いてない
  - 既存プラグインの機能に頼る
  - プロトタイプ段階ではほぼ以下だけ
    - `git-diff-image`: 1 行
    - `image.nvim`: 39 行
- 検討したほかの選択肢 (表示部分)
  - `sixel` で直に端末に吐く
  - `less` をいじって `sixel` 対応
  - 出力を HTML 化して `w3m` で見る