

有向グラフを用いたVCSのバージョン間における レイヤファイルの差分の可視化方式

山田 泰宏¹ 竹渕 瑛一¹ 鈴木 浩¹ 服部 哲¹ 速水 治夫¹

概要：画像情報の制作に用いられるレイヤファイルのバージョン管理を行うことにより，制作の効率の向上を図ることができる．レイヤファイルのバージョン管理をするための試みは近年いくつか見られるが，過去のバージョンのリストアが煩雑である．従来のレイヤファイル間の差分表示方式では，特定のレイヤのみの変遷を閲覧することが難しい．加えて，特定のレイヤのみを支援する技術は存在しない．そこで本研究では，複数のバージョンのレイヤファイル間の差分の可視化方式および，その可視化方式をUIとして活用する，レイヤのリストア支援システムを提案する．

Method for Visualizing Layer Files on Version Control System

YASUHIRO YAMADA¹ EIICHI TAKEBUCHI¹ HIROSHI SUZUKI¹ AKIRA HATTORI¹ HARUO HAYAMI¹

1. はじめに

画像情報を扱うソフトウェアの発展により，コンピュータを用いて画像情報の作成を行う機会は一般的なものになっている．例えば，ソフトウェアの開発では，視覚的に認識できるアイコンやボタンなどの画像情報がユーザーインターフェースの構築に用いられる．他方で，近年では画像情報を共有するPinterest[1]やPixiv[2]などのSNSが注目を集めており，専門的な知識を持たない人にも，フォトタッチソフトを用いた写真の加工やドローイングソフトを用いた絵画の作成など，画像情報の作成の機会が広がりつつある．

昨今では，ソフトウェアの開発に用いられるバージョン管理システム(VCS)によるバージョン管理の概念を，画像情報の作成においても活用する動きが出てきている．Github[3]やPixelapse[4]では，複数のバージョン間の画像情報の差分を可視化する機能を実装している．また，Adobe PhotoshopにはSubversionのクライアントとして動作するプラグインが存在し[5]画像情報のバージョン管理が簡単に行える．このように，昨今の画像作成では，画像情報自体のバージョン管理を行う手法が広がりつつある．

ソースコードをバージョン管理する目的として，過去のバージョンにおけるテキスト(文章やソースコード)の一部分(パラグラフ)を，現在のバージョンにおけるファイルに取り込むことで編集し，それを新たなバージョンとする場合が挙げられる．画像情報におけるバージョン管理においても，そのような目的がある．過去のバージョンにおけるレイヤファイルが持つ特定のレイヤの一部分(リージョン)を，現在のバージョンの編集で用いることができることに，画像情報のバージョン管理の必要性がある．

しかしながら，画像情報のバージョン管理を可能にする既存のソフトウェアでは，特定のレイヤの変遷を追跡するのに手間がかかる．既存の技術では，バージョン間のレイヤファイルの差分の表示方法が，単純な2次元の画像の比較になっている．すなわち，レイヤ統合後の比較になっていたりと，レイヤの順番を考慮していなかったりしている．また，過去のバージョンにおける特定のレイヤのみをリストアには煩雑な作業が伴う．よって，既存の技術ではリストアすべきレイヤの選定と，リストアの作業に手間がかかる．

そこで本研究では，複数のバージョン間におけるレイヤファイルの可視化方式およびその可視化方式をユーザーインターフェースとして活用した独自のVCSを提案する．本研究の目的は，提案システムの機能により，画像情報の

¹ 神奈川工科大学大学院
Graduate School of Kanagawa Institute of Technology

作成作業を円滑化することである。

提案する可視化方式では、有向グラフを用いることによって特定のレイヤの変化を視覚的に追跡しやすくする。また、ユーザの直感的な操作で、特定のレイヤを現在のバージョンにリストアする機能を備える。

2. 研究対象の現状

2.1 画像レイヤ情報

画像情報には大別して2種類が存在する。1つはラスタ画像と呼ばれる、画像がドットの集まりとして表現される情報である。もう1つはベクタ画像と呼ばれる、画像の構成要素（直線や曲線、幾何学的な図形など）と、その始点や終点、節点の座標や直径などの数値情報で構成される情報である。本研究ではJPEG, PNG, GIFなどのフォーマットに代表されるラスタ画像を「画像情報」とする。

画像情報を作成するプロセスでは、複数のラスタ画像を階層構造として含んだ情報(画像レイヤ情報)を用いることが一般的である。画像レイヤ情報に階層的に含まれる画像情報を統合し、最終的に1つのラスタ画像としてアウトプットをした方が制作における効率が高い。すなわち、画像作成のプロセスに限っては、画像作成者の用いるコンピュータのストレージ上には、画像情報ではなく、画像レイヤ情報が存在する傾向が高いと考えられる。

画像レイヤ情報を表すファイルフォーマットにはアドビシステムズ社のPhotoshopで扱われるPSDファイルやフリーソフトウェアのフォトレッタッチソフトであるGIMPで扱われるXCFファイルなどが挙げられ、アプリケーションごとに独自のフォーマットが用いられることが多い。一方で、各々のアプリケーションには、他のアプリケーションの画像レイヤ情報のファイルをインポートする機能を備えているものも見られる。

本研究では、PSDファイルを画像レイヤ情報として定義し、研究対象とする。理由として、Photoshopは多くの競合ソフトにおいても広く用いられており[6]、多くの画像作成用アプリケーションはPSDファイルのインポート・エクスポートに対応しているためである。

2.2 バージョン管理システム

VCSの役割は、特定のファイル群の変更履歴を管理することである。ソフトウェアの開発にはバージョン管理システム(VCS)が広く用いられる。ファイル群の変更点の保存(コミット)が可能であるため、ファイル群を特定のコミットの時点の状態(バージョン)に戻すことができる。VCS管理下のディレクトリの変化を検出する機能を持ったものや、コミットをしたユーザの情報を保持する機能があり、ファイルの変更の所在が明確になる。

またGitやSubversionのようなVCSのGUIクライアントには、2つのバージョンにおける特定のプレーンテキス

トのファイルの差分を表示し、変更点を可視化する機能がある(図1)。

```
87 88 //Get Solar Current
88 {
89 - //float offset = -7.33; //with buttryery power supply
89 - float offset = 44.52; //only usb bus power
90 + float offset = 2.48; //with buttryery power supply
91 + //float offset = 44.52; //only usb bus power
92 //float offset = 0;
93 //Get current of solar panel
solar_cur = analogRead(SOLAR_PORT);
```

図1 バージョン間のテキストファイルの差分表示例

VCSの変更履歴の保存機能は、単純にファイルのバックアップの役割も果たしている。すなわち、過去のバージョンにおけるテキスト(文章やソースコード)の一部分(パラグラフ)を、現在のバージョンにおけるファイルに取り込むことで編集し、それを新たなバージョンとする場合に活用されることも多い。変更点の可視化によって、過去のバージョンと現在のバージョンのファイルの、パラグラフ単位での折衷が容易になる。

本研究ではファイルの変更の履歴と、過去のバージョンにおけるファイルを何らかの形で保存することができるシステムを、VCSとして定義する。

2.3 過去のバージョンのレイヤの必要性

画像作成のプロセスにおいて、VCSでレイヤファイルを管理する場面を想定する。この場面において、画像の作成者は、過去のバージョンにおけるレイヤファイルが持つ特定のレイヤの一部(リージョン)を、現在のバージョンの編集で用いることができる。また、このような編集をすることにより、画像作成者の、画像情報の修正作業が効率的になる。

この編集の考え方は、2.2節で述べたプレーンテキストにおけるバージョン管理と同様である。

3. 関連システム

3.1 Github

Githubとは、ソフトウェア開発のプロジェクトを共有するためのWebサービスである。Githubは、VCSとしてGitを利用している。ブラウザ上で、GUIクライアントを用いることが可能であり、ファイルを開覧したり、バージョン間のファイルの差分を開覧することができる。更にGithubでは、バージョン間のラスタ画像の差分を4種類の方法で可視化して見ることができる(図2)。

3.2 Pixelapse

Pixelapseは、画像情報および画像レイヤ情報のファイルをバージョン管理することができるWebサービスである。サービスのユーザは、専用のアプリケーションを端末にインストールすることで、指定したローカルにあるディレク

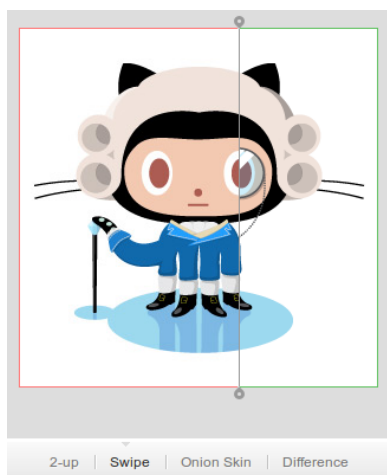


図 2 Github におけるラスタ画像の差分表示機能

トリ以下のファイル群と、サーバのディレクトリを同期することができる。ユーザが画像情報あるいは画像レイヤ情報を更新するたびに、サーバ側に過去のバージョンとしてファイルが自動的に保存される。すなわち、Dropbox[7]のようなオンラインストレージの一種と捉えることもできる。

専用のソフトウェアを用いずとも、ブラウザ上で PSD ファイルのような画像レイヤ情報の過去のバージョンがブラウザ上で確認できる。複数のバージョンの比較方法として、レイヤファイル内のレイヤを全て統合した時に見える画像を、隣接させて提示する機能がある。

4. 問題点

4.1 取り込むレイヤの選定に手間がかかる

過去のバージョンにおける特定のレイヤを用いる際、取り込むべきレイヤを含んだバージョンの選定に時間がかかるという問題が起こる。

なぜなら画像作成者は、コミットメッセージやコミットの更新日時等の情報からしかレイヤファイルの内容を類推することができないためである。また、3.2 節で述べた Pixelapse のようなサービスを利用したとしても、閲覧できる情報は、画像レイヤ情報に含まれる全てのレイヤを統合した後の単一の画像である。ゆえに、下位のレイヤが上位レイヤに隠れる形で、閲覧することができない可能性がある。そのため、過去のバージョンのレイヤファイルの内容を確認するためには、各バージョンにおけるレイヤファイルを各々リストアし、それぞれ専用のソフトウェアでレイヤの内容を確認しなければならない。

また VCS には、レイヤファイルのようなバイナリファイルの変更点を、バージョン間で可視化する機能を持ったものが存在しない。すなわち、この手法による管理では、過去から続く、特定のレイヤの変更の変遷のみを確認することが難しい。

4.2 レイヤの取り込み作業に手間がかかる

レイヤファイルの管理を VCS で行うことで全ての過去のバージョンにおけるレイヤファイルの、ファイル単位でのリストアが可能となる。

しかしながら、VCS にはレイヤ単位でのリストアに対応したものは存在しない。そのため、特定のバージョンにおけるレイヤを、現在のバージョンにおけるレイヤファイルに取り込む際には、現在と過去のバージョンのレイヤファイル両者を専用のソフトで開き、レイヤ単位で取り込む作業が伴う。

5. 提案システム

本研究では、VCS を独自に実装する。レイヤファイルの管理をその VCS で行うと同時に、独自のユーザーインターフェースを備えたクライアントを用いることにより、先述した問題を解決できると考えた。

5.1 レイヤファイル間の差分可視化機能

複数のバージョン間における、レイヤファイルの差分を有向 2 部グラフを用いて可視化する。レイヤファイルに対する操作をレイヤの追加、削除、順位の変更、更新の 4 つに分類し、それぞれを可視化する。提案方式では、レイヤファイルに含まれるレイヤ各々をノードとして表す。その際、2 バージョン間におけるレイヤファイルを、2 つの頂点集合に分割する。追加があったレイヤと、削除されたレイヤは、それぞれの変更に対応した色のノードとして表示する。片方の頂点集合の各ノードから、もう片方の頂点集合の各ノードへフローのあるエッジを示し、レイヤの順位の変更を可視化する。また、レイヤの更新が検出された場合、変更前のレイヤのノードと、変更後のレイヤのノード間のエッジを、専用の色で強調し、レイヤ画像間の可視化をする。可視化方式の概要図を図 3 に示す。

ユーザはマウス操作等により、ノードを指定すると、対応したレイヤの画像を見ることができる。また同様に、レイヤの更新を検出するエッジを指定すると、変更前と変更後のレイヤの画像の差分画像を見ることができる。

次に、この有向 2 部グラフを数珠つなぎにすることで、有向 N 部グラフ (N はコミットされたバージョンの数を表す) を作成する。これにより過去からの、特定のレイヤのみの変更点を追跡することが可能となり、取り込むレイヤの選定が容易になる。具体例を図 4 に示す。

図 4 におけるグラフは有向 3 部グラフになっており、3 つのバージョンが存在する。ここでは説明のため、各ノードにアルファベットを付与し、説明する。例えば画像作成者が図 4 の最新バージョンのレイヤファイルにおける B のレイヤの変遷を確認したい場合、ノード B の矢印を視覚的に追跡することで、直感的に確認することができる。また、 D のレイヤの変遷を追いたい場合、作成者はバージョン

2 までの変遷を確認し、バージョン 2 において新規作成されたレイヤであることを確認できる。言い換えれば、バージョン 1 以前におけるバージョンで D の存在を確認する必要がないことを認識できる。

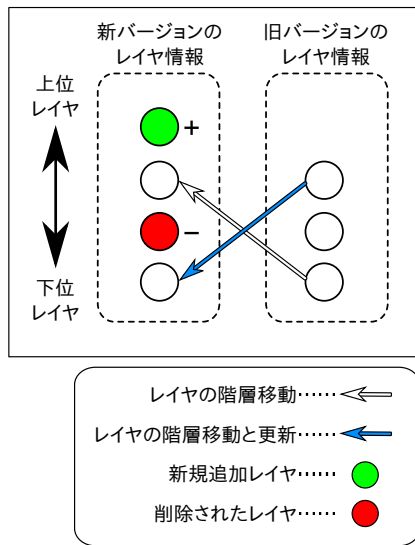


図 3 有向 2 部グラフを用いたレイヤファイルの差分の可視化

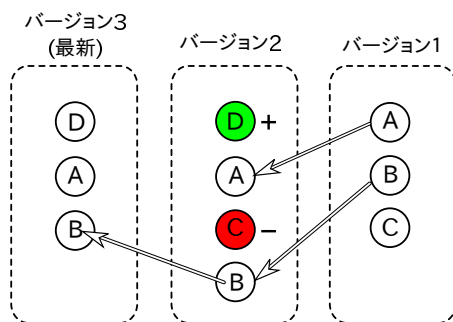


図 4 3 つのバージョンにおけるレイヤファイルの可視化

5.2 レイヤ単位でのリストアおよび取り込み機能

最新版のレイヤファイルと、過去のバージョンにおけるレイヤファイルを、先述の方法で可視化する。その際、ユーザによるドラッグアンドドロップの操作で、可視化された有向グラフのノードを操作することができる。この操作により、レイヤファイルに、過去のバージョンにおける特定のレイヤを取り込み、保存することができる。図 4 にあるレイヤファイルにおいて、画像作成者が、最新バージョンには存在しない C のレイヤをリストアする必要性に迫られた場合のリストア操作の概要を図 5 に示す。

5.3 差分圧縮による容量削減

提案システムでは、ユーザのコミットが発生した場合に JojoDiff[8] を用いて、バイナリファイルの差分情報のみを保存する。これにより、少ない情報量で、レイヤファイルの変更履歴を保存することができる。

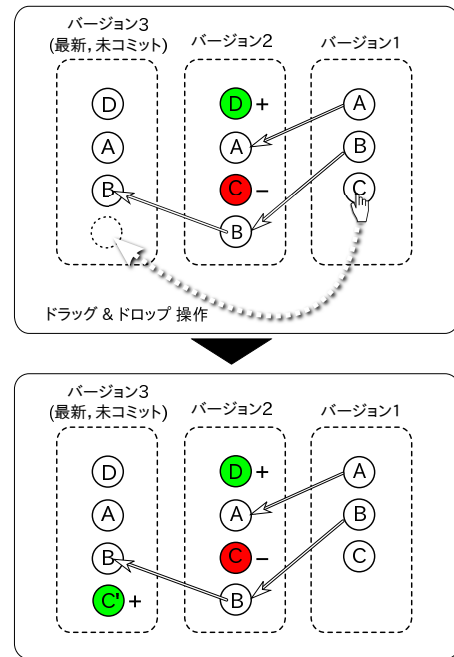


図 5 過去のバージョンのレイヤ単位でのリストア操作

6. おわりに

本研究では、バージョン間のレイヤファイルの可視化方式を提案し、その可視化方式を活用したリストア支援機能を持ったシステムを提案した。提案システムは、Git のようなローカルで変更履歴の保存ができるソフトウェアとして実装を行うことを想定しているが、複数人での画像作成の場合も想定し、バージョンの統合時のシステムの挙動に関しても研究することは、さらなる画像作成の効率化に向けて、有意義なことであると考えられる。

参考文献

- [1] Pinterest, 入手先 (<http://pinterest.com/>) (Accessed 2013.05.13).
- [2] Pixiv, 入手先 (<http://www.pixiv.net/>) (Accessed 2013.05.13).
- [3] Github, 入手先 (<https://github.com/>) (Accessed 2013.05.16).
- [4] Pixelapse - Visual version control for designers, 入手先 (<https://www.pixelapse.com/>) (Accessed 2013.05.16).
- [5] Subversion for designers, 入手先 (<http://www.pixelnovel.com/>) (Accessed 2013.05.15).
- [6] Adobe Photoshop - Wikipedia, 入手先 (http://en.wikipedia.org/wiki/Adobe_Photoshop#File_format) (Accessed 2013.05.14).
- [7] Dropbox, 入手先 (<https://www.dropbox.com/>) (Accessed 2013.05.16).
- [8] JojoDiff - diff utility for binary files, 入手先 (<http://jojodiff.sourceforge.net/>) (Accessed 2013.05.16).