7K-02

編集履歴にもとづいたイラストデータの透過的圧縮機構

吉川 綾乃 †　　松原 克弥 ‡
公立はこだて未来大学 †　公立はこだて未来大学 ‡

1　背景と課題

コンピュータの高性能化やペンタブレットなどの描画向け入力デバイスの普及にともなって，ペイントソフトウェアを使って描くデジタルイラスト（以下，イラスト）が普及した．イラストの投稿や流通，イラストに関する交流を目的としたイラスト特化型 SNS も商用化されており，その会員数も増大している [1]．

イラスト制作において，デジタルデータである特徴を活かして，編集過程における途中経過のイラストデータをストレージへファイル保存することがよく行われている．途中経過のイラストを保存しておくことで，その後の編集を取り消したり，イラストの一部を再利用することが容易になる．また，共同でイラスト制作を行う場合に，リモートストレージを介して編集過程のイラストデータを含めて共有することで，その作業を円滑に進めることができる．

一方，ストレージ保存の観点において，イラストデータは，テキストデータに比べてファイルサイズが大きくなりがちであるという特徴がある．そのため，製作期間が長引くと，前述した編集過程における途中経過のイラスト保存回数が増えて，ストレージ容量を圧迫する．また，リモートストレージにイラストを保存する場合には，ネットワーク転送のオーバヘッドが大きくなるという課題もある．

ストレージ圧迫への対処法として，データの圧縮処理が一般的に用いられている．しかし，多くのペイントソフトウェアがイラストデータをバイナリ形式で保存することや，保存されるイラストデータの内部構造も，ファイル形式やペイントソフトウェアに依存することから，様々なイラストデータに対して，汎用的な圧縮処理のひとつを一様に適用するだけでは，十分な効果が期待できない．

2　関連研究

Chen ら [2] は，ストレージに保存された複数の写真画像データに対する重複除去技術として，視覚的に同じだがデータが異なる画像を検出する方法を提案し，実際の写真データを用いた評価実験でその有効性を示した．本研究が対象とするイラストデータは，写真画像では想定されていない編集やレイヤー情報などのデータが含まれることがある．著者らは，本検出技術をイラストデータに対して適用した場合に検出正答率が低下すると推測する．

画像編集やスケッチのワークフローに着目した関連研究として，Chen ら [3] の画像向けバージョン管理システムがある．既存バージョン管理システムがバージョン毎に画像データを保存するのに対して，本システムは，画像に対して行ったアクションを保存することで，ストレージ消費量を削減できる．しかし，本システムは，特定のペイントソフトウェアに依存するかたちで実現されているため，多くのイラストレータが利用する他のペイントソフトウェアに適用できないという課題がある．

三戸らの研究 [4] は，同一のイラストから派生する複数のバージョンの作成や管理を支援するためのシステムを提案し，ユーザスタディによって有用性を評価している．一方，イラスト制作におけるストレージ容量の課題は解決していない．

3　提案

本研究ではイラスト制作の各編集・加工によるイラストデータの変化パターンに着目して，適用された編集・加工ごとに適切な圧縮方式を選択し，既存のペイントソフトウェアに対して透過的に圧縮処理を適用する手法を提案する．イラストデータを効率的に圧縮することによって，ユーザはリモートストレージにイラストデータをアップロードするのにかかる時間を減らすことができる．また，リモートストレージの提供者もより多くのデータを保存することが可能になる．適用された編集・加工に応じた適切な圧縮方式の選択では，例としてイラストの一部分のみを編集した場合と，イラスト全体に対して規則的に色加工処理を施した場合を考える．イラストの一部分のみを編集した場合には，その編集前後の保存データの差分をとる圧縮方法を採用する．イラスト全体に対して規則的に適用する色加工処理に対しては，加工前後でほとんどすべてのデータが変化してしまうため，加工後の隣接データの並びに着目して圧縮する方式を採用する．圧縮処理の適用はペイントソフトウェアに対

A Mechanism of Transparent Compression for Illustration Data Utilizing Editing History
† Ayano Yoshikawa, Future University Hakodate
‡ Katsuya Matsubara, Future University Hakodate

して透過的に行うことで，既存のイラスト制作環境への影響を抑えて本圧縮機構を導入可能にする．

3.1 適切な圧縮方式の選択

編集・加工ごとに適切な圧縮方式を選択するためには，イラストデータに施された編集がどのようなものであるかを特定する必要がある．ペイントソフトウェアに対して透過的に編集内容を推測するため，本提案手法では Git を使用し，ユーザが入力した Git のコミットメッセージからキーワードを取得する．コミットメッセージとは，Git でファイルをコミットする際に、どのような編集をしたのかなどをユーザが自由に記述できるものである．このコミットメッセージに，編集内容を表すような [○○] や○○:などのプレフィックスをユーザに付けてもらうことで，編集内容の特定を可能にする．そこで特定した編集内容を元に，イラストデータを圧縮する．

プレフィクスとして付ける編集履歴として，加筆，修正，加工，統合，複製，新規の 6 つを定めた．加筆や修正は，あるイラストに対して付け加えたり，削除するような編集が該当する．加筆はイラストに対して広範囲に編集を行うもので，修正はイラストの一部に対して編集を行うものとする．加工は，イラスト全体を調整するような編集が該当する．統合は，複数のイラストやレイヤーをひとつにまとめるような編集が該当する．複製は，あるイラストやレイヤーをコピーして，貼り付けるような編集が該当する．新規は，イラストファイルを新たに作成するような編集が該当する．修正や統合，複製の編集に関しては，編集前後で大きくイラストデータは変化しないと考えるため，編集前後の差分を差分を取る方式で圧縮する．加筆や加工，新規の編集に関しては，編集前後で大幅にイラストデータが変更されると考えるため，隣接データの並びに着目した圧縮方式を用いる．

3.2 編集履歴の取得と圧縮の適応

コミットメッセージの取得と圧縮の適用には，透過的に圧縮処理を適用する機構を実現するためにペイントソフトウェアに対して透過的である Git フック [5] を用いる．Git フックの中でも，コミットやマージといった操作に対してフックを行うので，クライアントサイドフックを利用する．更に，クライアントサイドフックの中でもコミットに着目するため，コミットワークフローフックを利用する．

圧縮処理を実現する流れを示すと図 1 となり，コミットメッセージが入力する前に行われる pre-commit フックで，コミットされたファイルがイラストデータであるかどうかを確認し，コミットされたファイルの情報を一時ファイルに保存する．コミットメッセージが入力された後，commit-msg フックでコミットメッセージにどのプレフィックスが付いているかを確認する．その後，プレフィックスから判断した編集内容を元に圧縮方法を決定し，一時ファイルに保存した情報を元にファイルを圧縮する．

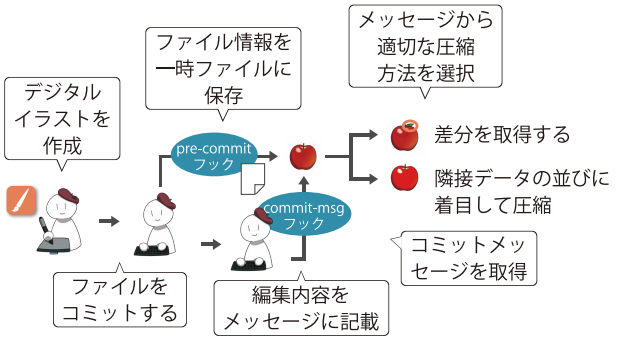

図 1　提案のフロー

4 まとめと今後の課題

本研究では，イラスト制作における各編集・加工情報からイラストデータの変化特性を分類し，分類された特性ごとに最適な圧縮方式を選択することで，イラストに特化した圧縮機構の実現を提案した．本機構の実現では，既存のペイントソフトウェアから独立に圧縮機構を適用できるようにするため，編集内容の入力とシステムへの取得の実現にバージョン管理ツールである Git の仕組みを用いた．

今後の課題として，ユーザ負担をともなわない編集内容の取得がある．ファイルシステムのレイヤで取得できる情報からイラストへの編集内容を推測する方法を検討する．また，その実装手法として，FUSE(Filesystem in Userspace) 機構を活用することで，OS とペイントソフトウェアの両方に依存しない透過的な圧縮機構を実現したい．

参考文献

[1] pixiv 株式会社. *pixiv* **のユーザー登録数が** *5000* **万人を突破**. `https://www.pixiv.co.jp/news/press-release/article/9962/`. 2020(参照 2020-08.02).

[2] Ming Chen, Shupeng Wang, and Liang Tian. “A High-precision Duplicate Image Deduplication Approach”. In: *JCP* 8 (2013), pp. 2768–2775.

[3] Hsiang-Ting Chen, Li-Yi Wei, and Chun-Fa Chang. “Nonlinear revision control for images”. In: *ACM Transaction on Graphics* 30.4 (2011), 105:1–105:10.

[4] 三戸夏美，坂本大介，小野哲雄. “イラストレーションの編集履歴を保持したバージョン管理システムの提案”. In: **インタラクション** *2019* **論文集**. 2019, pp. 512–516.

[5] git-scm. *githooks*. `https://git-scm.com/book/en/v2/Customizing-Git-Git-Hooks`. (参照 2020-11-02).