

グリッドレイアウトに基づく組版作業を支援する 手書きストロークを入力としたインターフェース

和田 優斗^{1,a)} 志築 文太郎^{1,b)}

概要: グリッドレイアウトは、紙面を構成する要素をグリッドに基づいて配置するレイアウト手法である。グリッドレイアウトはエディトリアルデザイン等で頻繁に活用される一方で、その実現には複雑な操作が要求されてきた。本研究では、手書きストロークを入力としてインタラクティブにグリッドレイアウトを生成するインターフェースを提案する。また、提案インターフェースを搭載した組版システムを Web アプリケーションとして実装した。

1. はじめに

書籍、雑誌、ポスター等の印刷物を対象としたグラフィックデザインやエディトリアルデザイン（以降、デザイン）では、グリッドレイアウトと呼ばれるレイアウト手法が広く採用されている [1]。グリッドレイアウトは、水平方向と垂直方向に仮想的な直線を引くことにより紙面を分割した上で、その直線に沿ってテキストや図版等の要素を配置するレイアウト手法である。

また、効果的な情報伝達を可能とするデザインは、近接 (Proximity)、整列 (Alignment)、反復 (Repetition)、コントラストというデザインの 4 原則 [2] を満たすことが重要である。デザイナーや編集者は、デザインを行う際にグリッドレイアウトを活用することにより、4 原則のうちの整列および反復を満たしたレイアウトを構築できる。

一方で、従来の DTP (DeskTop Publishing) ソフトウェアを用いてグリッドレイアウトを行うには、複雑な操作が要求される。すなわち、今日の DTP ソフトウェアを使う場合、複数のオブジェクトの整列と反復を精密に行うには、バウンディングボックスを用いてオブジェクトの編集を行う。オブジェクトの位置や寸法は絶対座標によって個々に指定されるため、ユーザがグリッドを構築するには整列と分布等の処理を逐一行う必要がある。

本研究は、グリッドレイアウトの生成および編集を支援するインターフェースの開発を目的とする。この目的のために我々は次に挙げる特徴を持った、手書きストロークを入力に用いて、グリッドレイアウトの生成をインタラクティブ

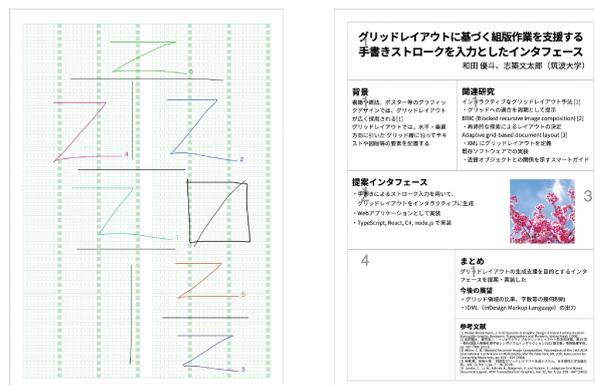


図 1 ストロークの入力 (左) と生成されたグリッドレイアウト (右)

ブに行うインターフェースを提案する。

- ラフスケッチからグリッドレイアウトを生成できる。
- ラフスケッチはストロークを用いて描かれ、タッチインターフェースおよびペンインターフェースとの相性が良い。
- ドラッグ操作を通じてスタイルを WYSIWYG に調節可能なテキスト編集機能を備える。

印刷物のデザインでは、まず大まかなラフスケッチを起こした上で、詳細なレイアウト等を指示する指定紙を作成し、DTP ソフトウェアを用いて最終的な入稿データを作成する [3]。本インターフェースを用いれば、ユーザに複雑な操作と座標計算を求めることなく、ラフスケッチを基に整列されたレイアウトを直接生成することが可能である。ユーザが行う入力とレイアウトの生成例を 図 1 に示す。ひとたび整形されたグリッドレイアウトからは、その定義に用いたラフを再度表示し、編集することが常に可能であるため、デザイン作業の効率化が期待される。

本稿では、開発中の提案インターフェースを述べるととも

¹ 筑波大学

^{a)} s2111710@s.tsukuba.ac.jp

^{b)} shizuki@cs.tsukuba.ac.jp

に、本インタフェースを搭載した組版システムを Web アプリケーションとして実装したのでこれを報告する。

2. 関連研究

グリッドレイアウトに関する研究は、ユーザによるデザインの作成を支援するものから自動生成を目的とするものまで、多面的なアプローチが提案されてきた。

2.1 インタラクティブなグリッドレイアウト手法

松田ら [4] は、規則的なグリッドレイアウトを支援するソフトウェアを開発した。ソフトウェアは、本文書体のスタイルから算出されるグリッドに要素のサイズが適合するかを、正弦曲線による周期として表示する。加えて、ユーザによる要素の配置によって決定されたレイアウトがどのような規則性を持ちうるかを検証し、推定されるレイアウトグリッドを提示した。

2.2 BRIC

Atkins ら [5] は、フォトコラージュの生成にグリッドレイアウトを適用し、BRIC (Blocked recursive image composition) と呼ばれる手法を提案した。BRIC は、キャンパスのサイズと写真の比率、枠線、パディングを入力として、再帰的な探索を用いてレイアウトを決定するアルゴリズムである。その結果、提案手法によって生成された画像はユーザから高い評価を獲得した。石橋ら [6] は、BRIC によって生成したレイアウト候補を提示し、ユーザからの評価に基づいて所望のレイアウトへと近づける対話型進化計算 (Interactive Evolutionary Computation) を導入したグリッドレイアウト生成システムを開発した。

2.3 Adaptive grid-based document layout

Jacobs ら [7] は、XML を利用したテンプレートとしてグリッドレイアウトを定義することで、文書の内容とレイアウトを明確に分離しつつ、画面幅に応じて変動する柔軟なレイアウトの作成を実現した。

2.4 既存ソフトウェアでの実装

DTP に広く活用される Adobe Photoshop, Adobe Illustrator では、スマートガイドと称した機能が搭載されている。スマートガイドは、近接するオブジェクトとの相対関係を補助線で表示し、スナップにより配置を支援する。Microsoft PowerPoint においても同様のインタフェースが採用されている。

2.5 本研究の位置づけ

これらの先行研究はグリッドレイアウトの実現を目的としながらも、以下に挙げる課題を持つことから、グリッドレイアウトの生成および編集に使用するインタフェース自

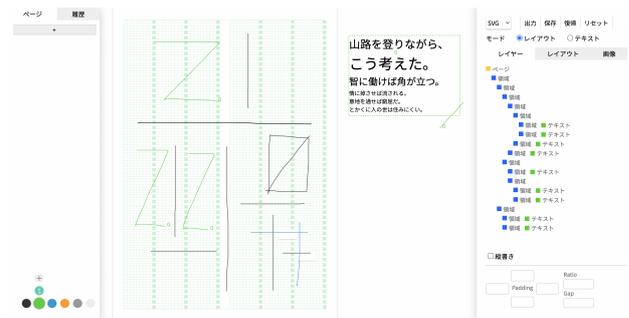


図 2 提案インタフェース

表 1 入力として与えられるストロークの種類と機能

入力	機能
├──	グリッドを分割する
◻	浮動要素を生成する
◻	領域内のパディングを指定する
ㄣ	テキストを関連付けて流し込む
+	幾何制約を付与する
メモ	メモ書きに使用する

体を根本的に改善するには至っていない。

- 最終的にはバウンディングボックスを用いたレイアウトの編集を前提とする (2.1 節, 2.3 節, 2.4 節)
- DTP としての用途を想定しない (2.2 節)

これに対し本研究では、ラフスケッチを描くようにして、グリッドレイアウトに基づいた組版作業を簡便に実現するインタフェースの開発を目的とする。

3. 提案インタフェース

本稿では、手書きによるストローク入力を用いて、グリッドレイアウトをインタラクティブに生成するインタフェースを提案する。なお、本システムは PC のほかに、タッチインタフェースまたはペンインタフェースを備えたタブレットでの作動を想定している。コマンド入力はキーボードショートカットかパイメニュー [8] を介して行われる。

3.1 レイアウトの定義

本インタフェースでは、レイアウトの位置とサイズはすべて描画されたストロークを基に決定される。本インタフェースを 図 2 に、ペンと形状の組み合わせと対応する機能を 表 1 に示す。

ユーザは、左下に表示されるツールパレットからペンを選択し、中央のページ領域にストロークを書き入れてレイアウト、テキストの流し込み、浮動要素 (写真等) の挿入、幾何制約の付与等を指定する。ペンには複数の種類が存在しており、ストロークの機能を定義するものと、後述するようにテキストボックスとグリッド領域を結び付けるもの

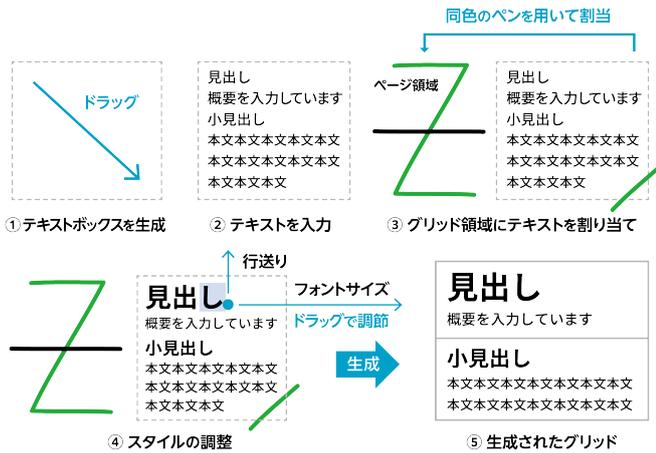


図 3 テキストの編集と割り当てを行うインタフェース

の 2 つに分かれる。入力として与えたストロークは、使用したペンの種類とストロークの形状によって区別される。なお、幾何制約としては、グリッド領域を等幅に設定する制約が導入されている。

3.2 テキストの入力

生成されたグリッド領域には、テキストを流し込んで表示することが可能である。

テキストの編集とグリッド領域への割り当てを行うインタフェースを 図 3 に示す。テキストを編集するには、画面上の余白でドラッグをしてテキストボックスを生成し、文章を入力する。入力したテキストは、行毎にフォントサイズ、行送り、字送り、文字揃え、ウェイト（太さ）、カーニング等のスタイルが設定される。スタイルは、数値等による指定のほかに、当該行にカーソルを置いた状態でマウスをドラッグすることでも調節できる。

対象のテキストボックスと領域それぞれに対して同色のペンを用いてストロークを引くと、テキストボックス内のテキストがグリッド領域に流し込まれる。

4. 組版システムの実装

本インタフェースを搭載した組版システムを実装した。本システムは、ユーザの入力を基に、整列処理と幾何制約の付与等を行った上で、グリッドレイアウトをリアルタイムで生成、提示する。右側のサイドバーには、生成されたレイアウトの構造が表示されるほか、グリッド領域のパディング、テキストのフォント、挿入画像等に関する設定項目が並ぶ。作成したレイアウトは JSON (JavaScript Object Notation) としてシステムに保持され、また、ユーザは作成したレイアウトを JSON, PDF, SVG (Scalable Vector Graphics) 形式のファイルとしてローカルに出力できる。

4.1 プラットフォーム

本システムの構成を 図 4 に示す。我々は本システムをク

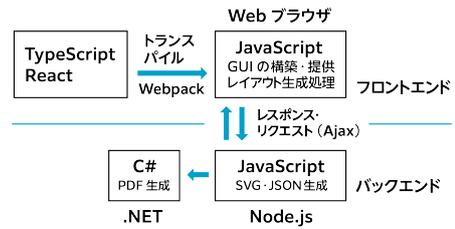


図 4 システムの構成

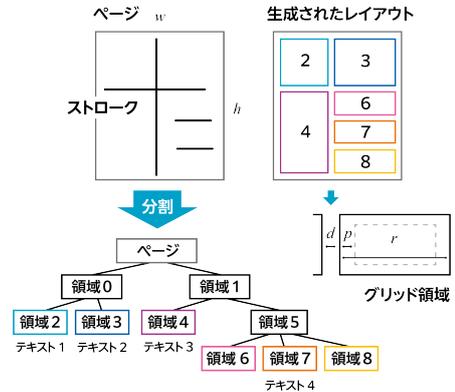


図 5 グリッド生成の過程

ロスプラットフォームとするために Web アプリケーションとして実装した。ユーザは Windows, macOS, iOS 等の様々な OS においてインストール作業も無く本システムを直ぐに利用できる。フロントエンドの実装言語およびフレームワークには TypeScript, React を、バックエンドの実装言語には JavaScript, C# を採用した。フロントエンドの実行環境は Web ブラウザ、バックエンドの実行環境は Node.js, .NET である。

4.2 グリッドの生成

本システムは、ユーザが描画したストロークの座標をリストとして保持し、その情報を基に領域を再帰的に分割してグリッドを生成する。

ストロークからグリッドを生成する過程を述べる (図 5)。まず、幅 w 、高さ h 、パディング p を持つページ領域 (版面) が規定される。続いてリストからページ領域を分割可能なストロークを探索し、そのうちの一つを用いて、ページ領域を水平、垂直方向のいずれかに分割する。生成された領域に対してさらに分割可能なストロークを再帰的に探索し、すべてのストロークが処理されるまで分割を繰り返すことで、最終的なグリッドが生成される。グリッド領域は、比率 r 、間隔 d 、パディング p 、分割方向の情報を持つ。

分割結果は木構造として保持され、各ノードがそれぞれのグリッド領域を表す。このうち、葉ノードは実際に表示される領域であり、必要に応じてテキストと関連付けられる。最終的には、木構造のデータを基に各グリッドの座標を算出して、画面上にレイアウトを表示する。

5. 議論

本システム上で厳密なレイアウトを構築するには、システム内部で使用する組版処理エンジンに対して、禁則やスペーシング処理等の強化を行う必要がある。タイポグラフィのレベルには、レイアウト、版面設計等の巨視的な視点から組版を捉えたマクロ・タイポグラフィと、細密な組版ルールを重視するマイクロ・タイポグラフィ [9] が存在する。テキストの厳密な配置に基づいてレイアウトを行う場合、両者は不可分な関係にあり、前者のみならず後者への対応も求められる。

また本インタフェースは、デザインの4原則のうち、整列、反復の2つを支援するに留まる。したがって、近接やコントラストの原則に則ったレイアウトを簡便に行うインタフェースの検討が要求される。

6. まとめと今後の課題

本研究では、グリッドレイアウトの生成支援を目的とするインタフェースを提案した。また、手書きストロークを主な入力に用いてレイアウト生成を行う提案インタフェースに基づいて、Webアプリケーションとして作動する組版システムを開発した。

今後の展望として、グリッド領域の比率または、グリッド領域が収める字数を固定する等の制約を導入し、新聞組版等の複雑な組版に対しても本システムを適用できるように改善を進める。加えて、DTPソフトウェアのデファクトスタンダードである Adobe InDesign を用いて本システムで生成したデータを編集できるように、出力形式として IDML (InDesign Markup Language) をサポートする予定である。

参考文献

- [1] Müller-Brockmann, J.: *Grid Systems in Graphic Design: A Visual Communication Manual for Graphic Designers, Typographers and Three Dimensional Designers*, Verlag Niggli (1996).
- [2] Williams, R.: *ノンデザイナーズ・デザインブック (第4版)*, マイナビ出版 (2016).
- [3] 工藤強勝: *デザイン解体新書, ワークスコーポレーション* (2006).
- [4] 松田聖大, 塚田浩二: インタラクティブなグリッドレイアウト手法の提案, 第15回一般社団法人情報処理学会シンポジウム インタラクション 2012 論文集, 情報処理学会, pp. 487-490 (2012).
- [5] Atkins, C. B.: Blocked Recursive Image Composition, *Proceedings of the 16th ACM International Conference on Multimedia*, MM '08, New York, NY, USA, Association for Computing Machinery, pp. 821-824 (2008).
- [6] 石橋 賢, 宮田一乗: 対話型グリッドレイアウト生成システム, *日本感性工学会論文誌*, Vol. 13, No. 1, pp. 7-16 (2014).
- [7] Jacobs, C., Li, W., Schrier, E., Barger, D. and Salesin,

- D.: Adaptive Grid-Based Document Layout, *ACM Transactions on Graphics*, Vol. 22, No. 3, pp. 838-847 (2003).
- [8] Callahan, J., Hopkins, D., Weiser, M. and Shneiderman, B.: An Empirical Comparison of Pie vs. Linear Menus, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '88, New York, NY, USA, Association for Computing Machinery, pp. 95-100 (1988).
- [9] 麥倉聖子: マクロ・タイポグラフィとマイクロ・タイポグラフィ (2016). <https://blog.justanotherfoundry.com/2016/05/マクロ・タイポグラフィとマイクロ・タイポグラフィ> (2022/12/18 閲覧) .