

マイクロタスクによるイラスト作成手法の提案

佐々木 孝輔^{1,a)} 平田 章² 井上 智雄^{3,b)}

概要: イラストは、読者を引きつけると同時に文書等の可読性を高めることができるが、だれでも作成できるものではないため、需要が大きいコンテンツの一つである。本研究では、従来のプロのイラストレータによる高品質で高価格のイラストではなく、潜在的ニーズの大きい、より簡単で低価格のイラストの作成を目的とする。そのために、マイクロタスク化したイラスト作成をクラウドソーシングにより実行する、新しいイラスト作成手法を提案する。また本手法を用いたイラスト作成実験を行うことで、手法の有効性とその特性について検討する。実験の結果、提案手法によりイラストが作成できることが示された。

キーワード: クラウドソーシング, マイクロタスク, イラスト作成

Method of Creating Illustrations by Crowd-sourced Microtasks

KOSUKE SASAKI^{1,a)} AKIRA HIRATA² TOMOO INOUE^{3,b)}

Abstract: Illustrations, which attract readers and improve readability of documents, have high demand partly because not everybody can make them in good quality. Conventionally clients request professional creators to draw illustrations and pay rewards. However, there exists potential huge needs for less quality and less expensive illustrations. We propose a new method of creating such illustrations by crowd-sourced microtasks. This paper also describes the experiment to study the feasibility of the proposed method, which proved positive result.

Keywords: Crowdsourcing, Microtask, Creating Illustration

1. はじめに

雑誌, 書籍等, 様々なメディアにおいて読者を惹きつけるものの一つにイラストがある。イラスト制作はこれまで一般的にイラストレータと呼ばれる専門性の高い職に就く者が請け負ってきた。そして比較的高品質高価格なイラストが扱われてきた。しかし近年はその高いコストを抑制しようと、イラストを必要とするクライアントがイラスト

レータやデザイナーなどのクリエイタに対し不透明な契約や不合理な取引をする問題も生じている [1]。文書に適切に図や線などを用いることで可読性が高まることが一般に知られており [2], イラストの需要は大きい [3]。それらのすべてが高品質を必要とするわけではない。例えば個人商店や零細企業のチラシ, 地域コミュニティやNPOの広報紙などにおいては、無料で使用できるイラストをダウンロードできるWebサイトが広まっており [4], [5], 現在イラスト市場が形成されている品質と価格よりもより低品質ながら低価格, ないし無料なイラストにも大きな需要があると考えられる [6]。しかしこのような無料イラストサイトで入手できる既存イラストは種類に限りがあり, 本当に欲しいイラストが得られるとは限らない。つまり従来のイラスト市場よりも, より普及品質と価格帯のイラスト市場が望まれるものの, そのようなイラストの供給は十分なされて

¹ 筑波大学大学院図書館情報メディア研究科
Graduate School of Library, Information and Media Studies,
University of Tsukuba

² 筑波大学情報学群情報メディア創成学類
College of Media Arts, Science and Technology, School of
informatics, University of Tsukuba

³ 筑波大学図書館情報メディア系
Faculty of Library, Information and Media Science, Univer-
sity of Tsukuba

a) ksasaki@slis.tsukuba.ac.jp

b) inoue@slis.tsukuba.ac.jp

いない。

本研究はこの未開拓の市場に供給するようなイラストの作成を対象とする。本稿では、マイクロタスク化したイラスト作成をクラウドソーシングにより実行する新しいイラスト作成手法を提案する。本手法により、従来よりも低品質ではあるが極めて低コストのイラストが供給されることが期待される。また、本手法の実現性を検証し、その特性を検討するために線画イラストを対象として行った実験についても述べ、本手法により十分にイラストが作成可能であることを示す。

2. 関連研究

2.1 複数人でのコンテンツ作成

コンテンツを作成する手法として、複数人が共同で一つの作品を作り上げるシステムが存在する。Wikipedia[7]は誰もが自由に編集ができるインターネット百科事典であり、この全体が二千万人以上の手によって作成されている。また Git[8]はプログラムのソースコードなどの変更履歴を記録、追跡するためのバージョン管理システムで、このシステムを利用し数多くのプログラムが共同で編集、更新されている。これらのサービスは不特定多数の人間によりコンテンツ作成を可能にする一方で、個々人のモチベーションにより作業の量および質が多分な影響を受ける。

また Andre らは、リメリックと呼ばれる五行詩の作成を、互いのことを知らない複数人がオンラインで行う際、同時に作業を行うか連続で作業を行うかで作品の質に違いが生じるかを実験を通じて検証した。その結果、複数人が連続で、すなわち1人がタスクを終了したときに別の1人がそのタスクを受け継いで作業を進める方が、複数人が同時に作業を行うより作成されるリメリックの質が高いことが分かった [9]。この研究より、オンライン上で互いが知人同士ではない作業者同士が1つのコンテンツを作成する際、同時に作業するよりは作業者が各々、それまでのタスクを受け継いでタスクを請け負うことでコンテンツの質が高くなる可能性が示唆される。

2.2 クラウドソーシング

複数人が一つのタスクを行う手法として、クラウドソーシングがある。また、このクラウドソーシングの中でも、1回あたりのタスク処理に要する時間が極めて短くなるようにタスクを細分化して作業者に提示する手法として、マイクロタスクがある。Amazon Mechanical Turk(MTurk)[10]では、マイクロタスクを提示する Requester とそのタスクを実際に実行する Worker が存在し、タスクを完了することで Worker が Requester から所定の報酬を貰い受けることができるようなプラットフォームを提供している。このシステムでは、Requester がタスクを依頼するためには報酬を用意する必要があるが、その報酬は Worker1 人につ

き、時間あたりおよそ2ドルと少ない [11]。David らは、MTurk において報酬は作業者にとって最も重要な要因と述べている [12]。

一方でマイクロタスクの体系によっては、報酬を用意する必要がない場合がある。Vaish らは報酬を支払わないクラウドソーシングとして Twitch と呼ばれるシステムを提案した。これはスマートフォンのアンロック画面において、数秒程度で終わるタスクを提示するものである。このシステムにおけるタスクは、作業者もしくはその身の回りに関する情報を4枚から6枚のタイルから選択するというものだが、このシステムを用いた実験により、報酬を用意しなくともごく短い時間で終了するタスクならば、作業者に大きな負担をかけることなく情報を収集できることを明らかにした [13]。また Winter らは、報酬が作業者に与える影響は、作業の質ではなく作業量であるということ、報酬の有無が異なるタスクを用いた実験によって示している [14]。これらの知見は作業者が、報酬が与えられないボランティアであってもクラウドソーシングを実施できる可能性を示している。

2.3 クラウドソーシングにおける質の担保

クラウドソーシングでは不特定多数の作業者がタスクを行うため、しばしばそのタスクの実行結果の質について議論が上がる。Walter らは文字識別タスクによる実験を通じて、マイクロタスクにおいて悪意のある攻撃が生じうることを示した。そのような可能性を回避するために、作業者に対し提示する情報を制限する、信頼できる作業者の採用を強化する、などといった対策が必要と講じている [15]。

また別の観点から、Sampath らおよび Kittur らは、より作業者が質の高い仕事をこなすためのタスクデザインのあり方を提案した。Sampath らはタスクにおけるユーザインタフェースをより視覚的に認知しやすくなるよう工夫することで [16]、Kittur らはタスクで提示した要求をより正確にこなしていることを確認する質問を用意することで、それぞれタスクの質が向上しうることを明らかにした [17]。さらに Huang らは Noun counting task を通じ、作業者同士の名前や性別、国籍といった個人情報公開し、作業者同士のインタラクションがある場合の方が、ない場合に比べ質の高い作業結果が生じることを明らかにした [18]。マイクロタスクの質は、これらの研究によって示されるとおり、作業者、タスク自体双方の工夫により高めることが可能である。

3. マイクロタスク型クラウドソーシングによるイラスト作成

3.1 提案手法

本研究は1枚の実在の写真からイラスト、特に線画を作成するための手法を提案する。通常、線画を作成する際は

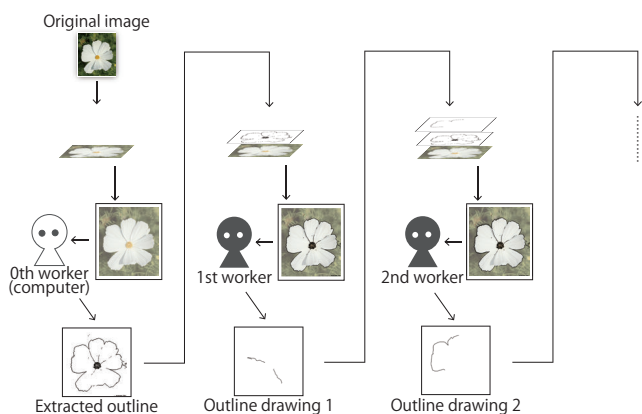


図 1 提案手法

物体の輪郭など、その画像を特徴づける線を描く。線を描くという行為自体は誰にでも行うことができ、また手本となる画像があれば何かしらの物体の線画を描く行為も難しいものではない。そこで、この誰にでもできる画像から線画を描くという行為をマイクロタスクとして多数の作業員(ワーカー)に提示することで、クラウドソーシングによって写真から線画を作成することができる。これにより、クライアントが1枚の線画を入手するためにクリエイタを雇う必要がなくなり、また2.2節で述べたように、クラウドソーシングを用いることで低コストで線画を手に入れることが可能となる。この手法を実現するために、複数のワーカーの作業によって1枚の線画を作成できる仕組みを考える必要がある。

本研究が提案する手法の概念図を図1に示す。このマイクロタスクでは、ワーカー一人ひとりが描いた線画を重ね合わせていくことで、最終的に1枚の線画の完成を目指す。しかし、全てをワーカーに委ねて線画の作成を開始してしまうと、膨大な作業量と人手を要する。そこではじめに、コンピュータを0番目のワーカーとみなして、原図の写真から画像処理技術による輪郭抽出を行い、その画像を0番目のワーカーの作業結果として保存する。次の1番目のワーカーはクラウドソーシングにてタスクを行う人間で、このワーカーには元の画像と、0番目のワーカーの作業結果が重なって提示される。1番目のワーカーは、この重なった画像を元に、線画として不足している部分を描き足す。描き足された画像は1番目のワーカーの作業結果として保存され、2番目のワーカーには、元の画像、0番目の作業結果、1番目の作業結果が重なって提示される。このようにワーカーは、自分より前のワーカーが作業した結果を全て重ねて見ることで、不足している部分を補う形で線画を少しずつ加筆していく。この線画を少しずつ加筆していくタスクがマイクロタスクであり、このマイクロタスクを多数のワーカーがこなすことで、最終的には元の写真画像を除く全ての加筆画像を重ね合わせた画像が1枚の完成した線画となる。このように複数人が共同で1枚の写真から線画を作成することができる。

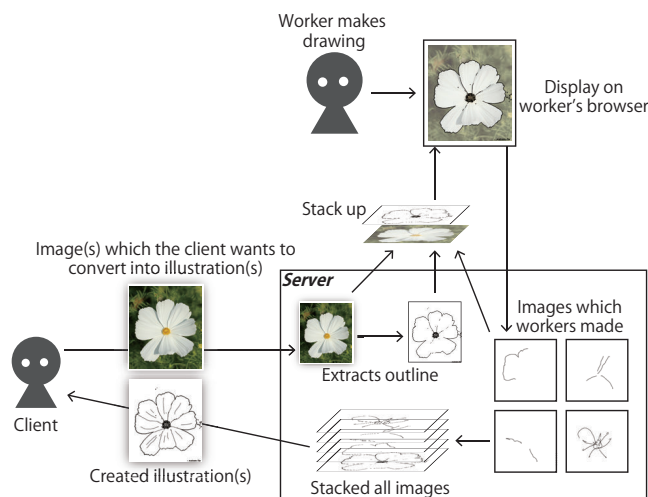


図 2 システムの構成図

3.2 システム設計

提案手法によるイラスト作成システムの構成図を図2に示す。まず、クライアントがイラスト化したい元画像となる写真をシステムに送る。ここで、システムが0番目のワーカーとして輪郭線を自動抽出し、ある程度の線画を作成する。その後、ワーカーがマイクロタスクによって、線画として不十分な部分に加筆していく。ワーカーが描き足した線は別々の画像としてシステムに保存され、次のワーカーに提示する際、それまでの画像に重ねられる。最終的に全ての描画線を重ねることでイラストが完成し、重ねた画像を1枚の画像にした後、クライアントに送られる。

またイラストを作成するために、元となる画像の上から輪郭線をなぞるように線を描くことでイラストの作成を行うことが可能なインタフェースが必要となる。これは、ワーカーの誰もがこのマイクロタスクに参加できるようにするために、絵が描ける描けないに関係なく線画を作成できるようなシステムを提示する必要があるためである。

4. 実装

4.1 モジュール

線画を作成するページを含め Web ベースのシステムを実装した。システム全体は図2より次の4モジュールで構成されている。

- (1) 元画像の受付
クライアントが元画像を、アップロードページを通じてシステムに登録するモジュール。
- (2) 輪郭の自動抽出
登録された画像から、輪郭を自動で抽出し、png 画像として保存するモジュール。
- (3) タスク提示
元画像と輪郭抽出した画像を重ねあわせ、タスクとして提示するモジュール。
- (4) 作成画像の提供

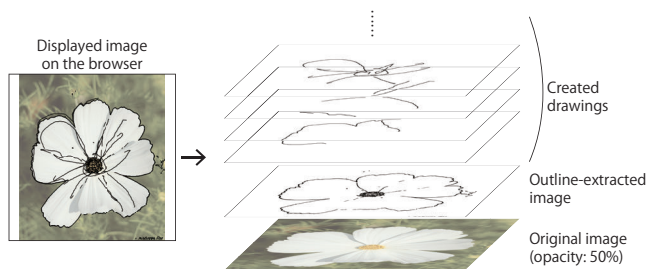


図 3 ワーカーへの画像提示手法

マイクロタスクを経て完成したイラストをクライアントへ提供するモジュール。

モジュール (1) では PHP を利用し、画像のアップロードを受け付ける。アップロードされた画像は固有 ID を付され、システムに保存される。その後、モジュール (2) へと引き渡される。

モジュール (2) では Python, および画像認識ライブラリとして OpenCV を用い、モジュール (1) から受け取った画像の輪郭を抽出する。抽出した画像は元画像を紐付けられ、ワーカーへ提示される。

モジュール (3) では図 3 のように、PHP を用いてシステムに保存された画像セットを重ね合わせて提示する。提示の際は HTML5 の canvas 要素により、画像はレイヤを複数枚重ねることでワーカーに提示される。最下層にクライアントから登録された元画像を不透明度 50% で設置し、その上に輪郭を自動抽出した画像を、さらにその上にワーカーが作成した画像を順に重ね、ワーカーはその一番上のレイヤに線画を描くことになる。これにより、ワーカーは元画像をなぞりつつ、かつそれまでのワーカーが描いてきた線に加筆することで線画を作成していくことができる。描画する機能は JavaScript によって提供される。

モジュール (4) では、モジュール (3) までに作成された画像を全て重ね合わせ、PHP を用いて合成する。その後合成した画像をクライアントへ提示する。

4.2 ワーカーへのインセンティブに対する配慮

2.1 節で述べたように、ワーカーのモチベーションによってシステムへの貢献度、すなわちタスクへの参加度合いが変動してしまうと、コンテンツを物量的に安定して作成することができず、コンテンツの供給がまばらになる。本研究では、ワーカーがモチベーションに左右されることなくマイクロタスクを行うように、図 4 のような想定環境を設定した。

例えばユーザが、Web ブラウザで既存のアップロードサイトを訪問し、何らかのファイルをダウンロードしようとする。ダウンロードのリクエストをサーバに送ると、スパム防止のためのパスワードなどの入力画面、または reCAPTCHA [19] などのマイクロタスクが呼び出され、ユーザはこれを完了しないとファイルをダウンロードすること

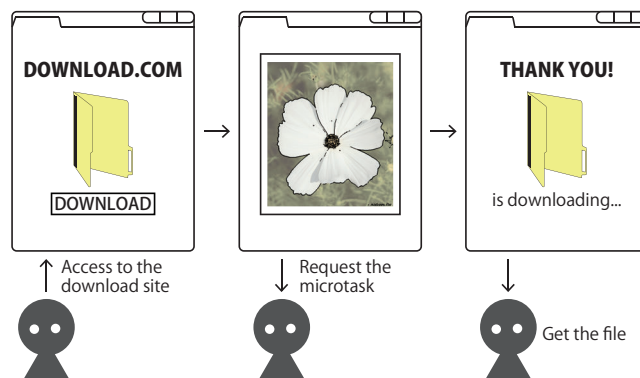


図 4 システムの実行想定環境

ができない。本システムはこのタイミング、すなわちユーザからのリクエストの直後に、パスワードの入力等の代わりに実行する。これによりユーザは必ずこのタスクを実行しなければならない、元々の要求であるファイルをダウンロードすることができないため、モチベーションによらずタスクを行わなければならない。

5. イラスト作成実験

提案するイラスト作成手法は全く新しい手法であるので、イラスト作成のタスクのマイクロタスク化にあたっての注意点、ワーカーにすべき指示などを具体的に検討でき、また、1 ワーカーがどのようにタスクを行うか、複数ワーカーによる作成過程はどのようになるかについて知ることができ、事例を得ることを目的として、以下のようにイラスト作成実験を行った。

5.1 マイクロタスクの作成

イラスト作成の原図となる写真を 10 枚用意した (図 5)。写真は、輪郭がはっきりと読み取れるものと読み取れないものの両方を含み、また被写体や構図も多様であるようにした。

イラスト描画のマイクロタスク化について、は描画量ではなく描画時間を制約した。描画量はその時点でのイラストの完成度による変動が大きく、適切なタスク分割にならないと考えられるためである。これを、ワーカーである参加者が線を描き始めてから一定時間だけ入力可能とする形で実装した。

描画時間は次のような予備実験により決定した。まずは実験者 2 名で、描画時間 2 秒から 20 秒まで 1 秒単位で変更しながら線画の作成作業を試行した。その結果、線画を描くのに適当だと感じる時間を 10 秒以内と判断した。続いて、実験者 2 名および実験参加者 4 名 (男性 3 名, 女性 1 名) で、2 秒から 10 秒まで描画時間を 1 秒単位で変更しながら、10 枚の線画の作成作業を試行した。ボランティアの作業として負担を感じないことと、何がしかの線を描くだけの余裕はあることについて、最適と考える時間を尋ねた。



図 5 実験で使用した画像

その結果, 参加者全員が最適とした秒数の平均は 6.7 秒であったため, 1 マイクロタスクの所要時間を 7 秒とした。

5.2 参加者と実験環境

参加者は 29 名 (男性 25 名, 女性 4 名, 予備実験参加者と重複なし) であった。参加者への報酬はなく, また入力インタフェースとして Wacom 社製ペンタブレット (CTE-650, 描画範囲約 14x22cm) を, ページ表示用 PC として VAIO Pro 11 (表示領域 11.5inch, 表示解像度 1920x1080px) を使用した。また実験は同構内で実施し, 全員椅子に着席し, PC およびペンタブレットを机に設置した状態で実施した。

5.3 手順

ここで本実験で参加者が見る画面を図 6 に示す。画面には, 線画を描くキャンパスの他に「これで完成!」ボタンがある。このボタンは, 線画にこれ以上線を描く場所がなく, 線画として完成されていると判断した際にワーカが押すものである。実験開始前には, この Web ページを提示しながら, 次の内容を参加者に対して説明した。

- これから 10 枚の写真について画像の模写を行ってもらうこと。また完成図として, 図 7 左の写真のような模写を目指すこと
- 1 枚の写真から線画を描く際, 7 秒の制限時間があること。またこの制限時間を過ぎると線を描くことができなくなるということ
- 制限時間内に全体を模写する必要はないので, 落ち着いてゆっくりと, 正確に線を引くようにすること
- 線を描く前に提示された線画が完成されているものだと判断した場合, 「これで完成!」ボタンを押すことで, その画像への描画を終了できること

また説明時に, 参加者はシステムを用いて線を描く練習を同ページにて行った。ここでペンタブレットの操作感, およびシステム上での描画の感覚に慣れた後, 実際のタスクに移行した。

図 8 は実験において一参加者が取り組むタスクのフローチャートである。1 枚ずつランダムに図 5 に示した画像を提示し, 参加者は線画を描く, もしくは「これで完成!」ボタンを押す。なお, 少しでも線を描き始めると「これで完成!」ボタンは消えるため, 参加者は写真 1 枚につき, 線

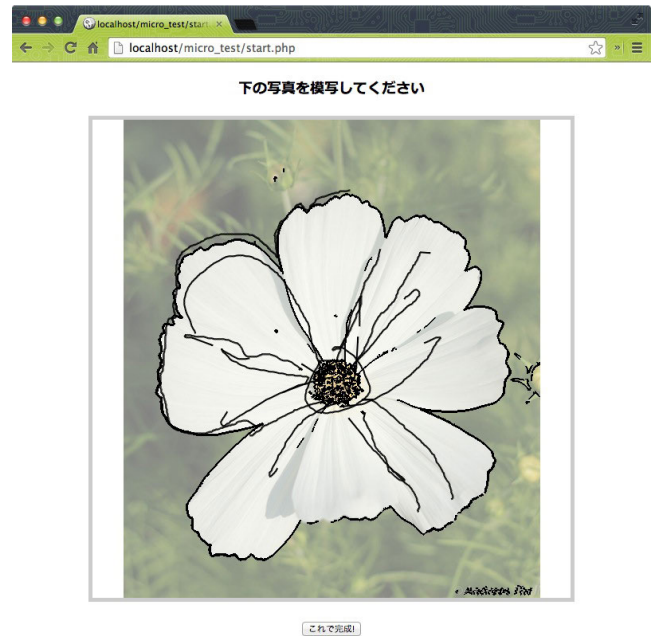


図 6 ワーカに提示される画面

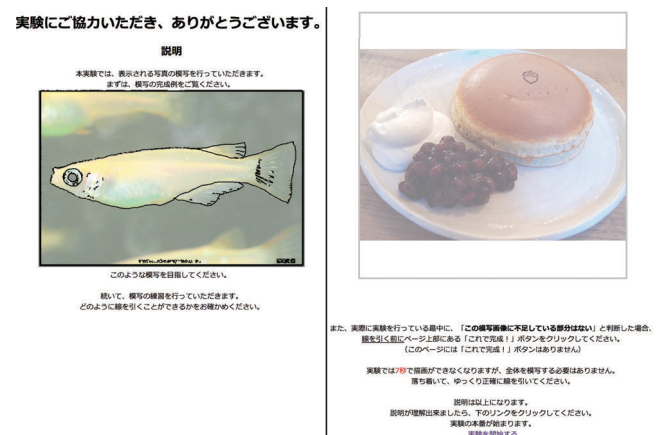


図 7 システムの練習ページ

を描き加えるか完成と判断するか, どちらかしか実行できない。この作業を, 準備した写真画像 10 枚分繰り返した。また 2 人目以降の参加者には, 元の画像と輪郭を抽出した画像の他に, それまでの参加者が描いた線画が全て重ね合わせて提示される。

6. 結果・考察

6.1 描画結果

各ワーカが各画像に対してどのような線を描画したかを, 図 9 に示す。また, これらを重ね合わせた描画の変化, つまり線画イラストが作成されてゆく過程を図 10 に示す。それぞれ, 左上から右下に向かって時系列順に並べた。画像によって枚数が異なるのは, 実験に参加した 29 名のうち, 描画を行わず「これで完成!」ボタンを押し, 線画が完成したと判断したワーカの作業分については表示していないためである。図 9, 10 にある赤線は, ワーカが線画が完

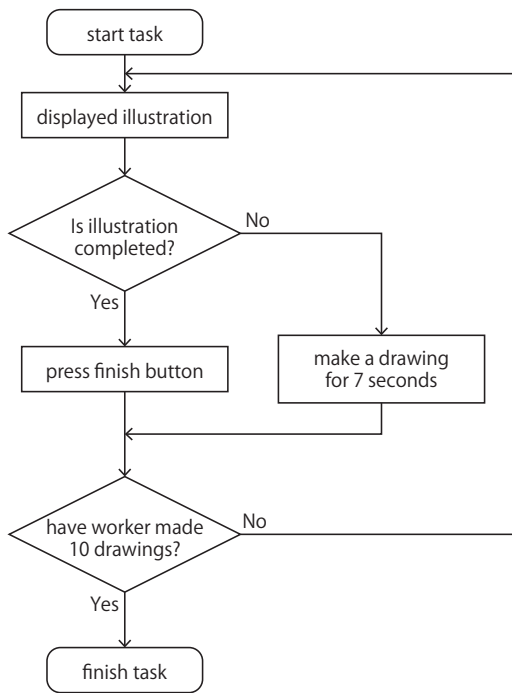


図 8 実験タスクのフローチャート

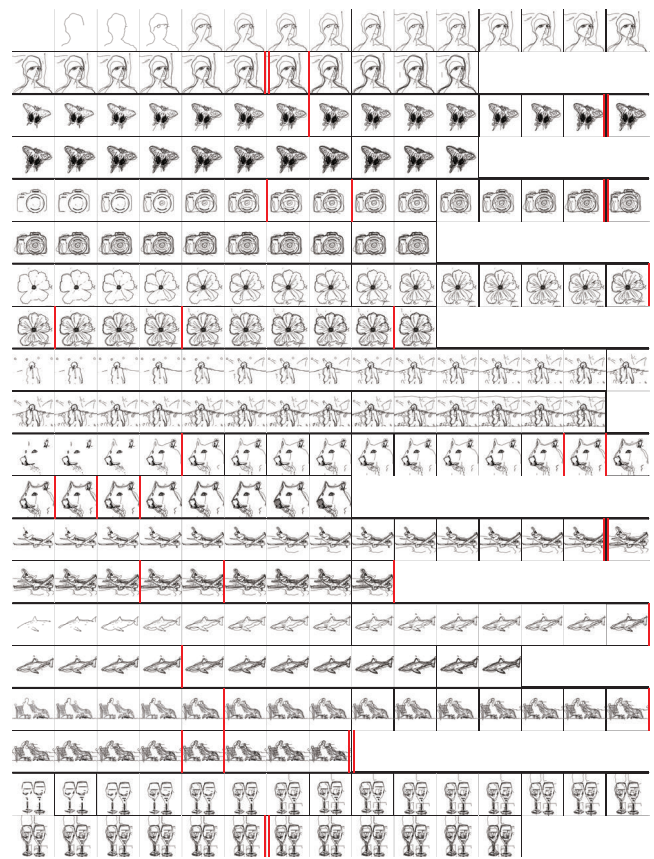


図 10 線画イラストの作成過程

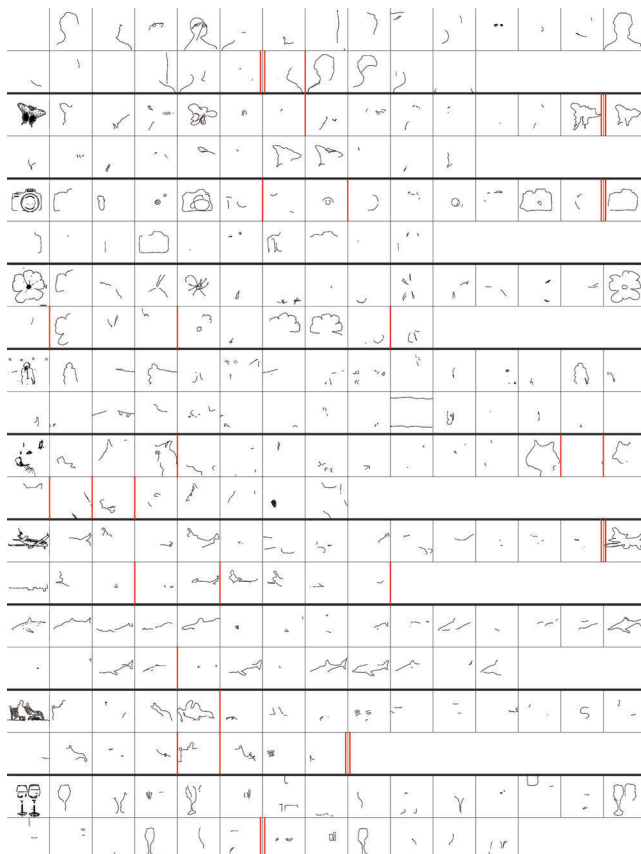


図 9 各ワーカーの描画内容

成したと判断したタイミングを示す。赤線が二重になっている部分では、ワーカーが2人連続で線画が完成したものと判断した。

6.2 イラストの質とワーカ数

図 11 に、それぞれの画像について、実験者 2 名が完成したと判断した時点での線画イラストを示す。左上の画像から右に向かって順に、23 番目、1 番目、21 番目、17 番目、13 番目、18 番目、5 番目、18 番目、10 番目、13 番目に描画が行われた後の画像を完成した線画とした。ばらつきはあるものの、平均しておよそ 14 回程度の描画が行われたときに、線画が完成したと判断されたことが分かる。また、いずれの線画においても輪郭部については複数人が何度も線を描き重ねる傾向にあり、内部に描画される細かい線よりも太く描画されていることが分かる。したがって、十分な多数がタスクを行う場合は、その内の多数が描いた描画線を残すことで、輪郭線だけがはっきりと残るイラストが完成できると考えられる。

また、ぼやけた人の写真のようにコンピュータによる画像処理技術による輪郭の抽出がうまく行えなかった画像についても、作成が進むにつれて輪郭線がはっきりと描かれる様子が認められる。これにより、コンピュータによる輪郭抽出の成否によらず、この手法を用いることで1枚の写真画像から線画の作成は十分可能であることが窺える。

6.3 イラスト作成過程の事例

図 12 は今回の実験で作成した画像のうち、白い犬の描画過程について拡大したものである。図 9, 10 同様、赤線

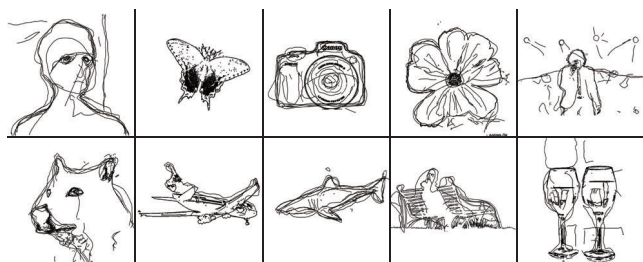


図 11 完成した線画イラスト

はワーカがその時点で線画が完成したと判断したタイミングを示す。

この画像では、0番目のワーカ、すなわちコンピュータによる輪郭の抽出の際、主に犬の体の輪郭部の抽出がうまく行えなかった。また図5からも確認できるように、この画像に関しては人間の目からも輪郭をはっきりと確認することができない。しかし2番目のワーカ以降、その輪郭部に徐々に線が描き足されていく様子が分かる。3番目のワーカの描画終了時点で、1名のワーカが完成の判断を下したが、4番目のワーカ以降、犬の喉の部分、耳の部分などを中心に線が加筆されている。線画を作成するにあたり、輪郭線をなぞるだけではなく線を重ねて描くことで影を描こうとしている様子が窺える。このことから線画を作成する際、まずは輪郭部から描画を始め、ある程度の輪郭が浮き彫りになった後、影となる部分など細部に線を加筆していくことが分かる。この例の場合、3番目のワーカの描画が終了した段階で輪郭がある程度浮き彫りになり、それ以降で細部の描画が始まったといえる。

その後、11番目の画像描画時点までは細部の描画を行っていたワーカが、12番目の画像描画以降、再び輪郭部の加筆を中心に描画を進めている。そのため、この12番目の描画を境に、輪郭線が強調され始めた。実験者が完成と判断したのは18番目の画像だが、これはある程度輪郭線が強調された後の画像であり、この付近で複数のワーカがほぼ連続して完成の判断を下している。このことは線画を作成する際、ワーカはまず輪郭線から描画を始め、ある程度の輪郭線が作成された後に細部の描画に移り、さらに再び輪郭線の描画を行うというように、作業の手順がある程度ステップとしてまとまっている可能性があることを示している。また、そのステップが移るタイミングで、ワーカは完成の判断を下すと考えられる。実際にワーカが完成と判断したものをそのまま線画の完成と判断できるかどうかは、また別に検討する必要がある。

6.4 入力インタフェースによる違い

今回の実験はペンタブレットを使用した実験であったが、実験に参加した29名のうち、21名がペンタブレットに触った経験がなかった。入力インタフェースによって作成されるイラストがどのように変わるかは検討課題である。

7. まとめ

本研究では、マイクロタスク型クラウドソーシングによって、イラストを作成する手法を提案した。手法の実現性を実験的に検討したところ、線画イラスト作成を時間マイクロタスク化しても一般的なイラストであれば14名程度のワーカによって作成可能であることが示された。今回は線画イラストの作成を題材としたが、線画イラストの彩色等についても同様の手法が適用可能であると考えている。さらにこれを基本手法としてイラストの連続であるアニメーションの作成も可能と考えている。

謝辞 本研究の一部は、科学研究費補助金26330218の支援により行われた。

参考文献

- [1] 内閣知的財産戦略本部コンテンツ強化専門調査会：デジタルコンテンツの振興戦略 ― 日本を世界トップクラスのデジタルコンテンツ大国にする ―, 入手先 (<http://www.kantei.go.jp/jp/singi/titeki2/houkoku/180220dezi.pdf>).
- [2] Diana Patterson: Technical writing: lines & spots, *SIGDOC Asterisk J. Comput. Doc.*, Vol.2, No.10, pp.8-10 (1976).
- [3] 株式会社クラウドワークス：相場や需要を押えよう！イラスト業のモノクロとカラーの挿絵の料金, 入手先 (<http://crowdworks.jp/public/jobs/category/27/articles/8754>).
- [4] 無料イラストなら「イラストAC」, 入手先 (<http://www.ac-illustr.com/>).
- [5] イラスト無料素材 イラストボックス, 入手先 (<http://www.illustr-box.jp/>).
- [6] 小学校の広報で使える、無料のイラストを教えてください。入学おめでとう... - Yahoo!知恵袋, 入手先 (http://detail.chiebukuro.yahoo.co.jp/qa/question_detail/q11107922591).
- [7] Wikipedia, the free encyclopedia, 入手先 (<http://en.wikipedia.org/>).
- [8] Git, 入手先 (<http://git-scm.com/>).
- [9] Paul Andre, Robert E. Kraut, Aniket Kittur: Effects of Simultaneous and Sequential Work Structures on Distributed Collaborative Interdependent Tasks, *CHI '14 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp.139-148 (2014).
- [10] Amazon Mechanical Turk, 入手先 (<https://www.mturk.com>).
- [11] Ross Joel, Irani Lilly, Silberman M. Six, Zaldivar Andrew, Tomlinson Bill: Who Are the Crowdworkers?: Shifting Demographics in Mechanical Turk, *CHI '10 Extended Abstracts on Human Factors in Computing Systems*, pp.2863-2872 (2010).
- [12] David Martin, Benjamin V. Hanrahan, Jacki O'Neill, Neha Gupta: Being a turker, *CSCW '14 Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, pp.224-235 (2014).
- [13] Rajan Vaish, Keith Wyngarden, Jingshu Chen, Brandon Cheung, Michael S. Bernstein: Twitch Crowdsourcing: Crowd Contribution in Short Bursts of Time, *CHI '14 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp.3645-3654 (2014).

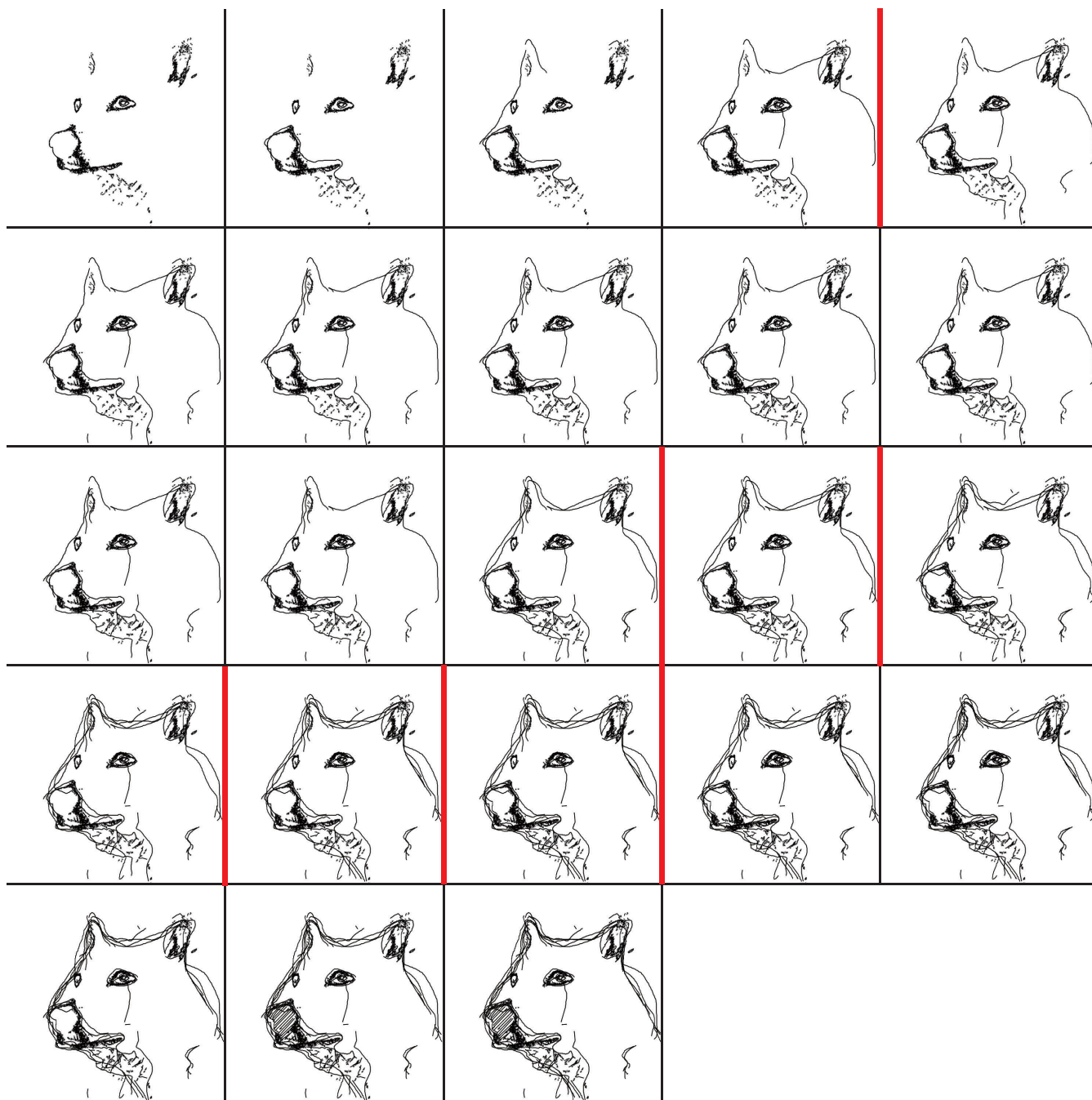


図 12 白い犬の描画過程

- [14] Mason Winter, Watts Duncan J.: Financial Incentives and the “Performance of Crowds”, *SIGKDD Explor. Newsl.*, Vol.11, No.2, pp.100-108 (2009).
- [15] Walter S. Lasecki, Jaime Teevan, Ece Kamar: Information Extraction and Manipulation Threats in Crowd-Powered Systems, *CSCW '14 Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, pp.248-256 (2014).
- [16] Harini Alagarai Sampath, Rajeev Rajeshuni, Bipin Indurkha: Cognitively Inspired Task Design to Improve User Performance on Crowdsourcing Platforms, *CHI '14 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp.3665-3674 (2014).
- [17] Aniket Kittur, Ed H. Chi, Bongwon Suh: Crowdsourcing User Studies With Mechanical Turk, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp.453-456 (2008).
- [18] Shih-Wen Huang, Wai-Tat Fu: Don't hide in the crowd!: increasing social transparency between peer workers improves crowdsourcing outcomes, *CHI '13 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp.621-630 (2013).
- [19] recaptcha, 入手先 (<http://www.google.com/recaptcha/intro/>).
- [20] Khai N. Truong, Thariq Shihpar, Daniel J. Wigdor: Slide to X: unlocking the potential of smartphone unlocking, *CHI '14 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp.3635-3644 (2014).