

マイクロタスクによるイラスト生成の研究

筑波大学

図書館情報メディア研究科

2016年3月

佐々木 孝輔

目次

第1章	はじめに	1
第2章	関連研究	3
2.1	複数のワーカによるタスクの実行と報酬の有無	3
2.2	クラウドソーシングを利用した複雑なタスクの実行	3
2.3	クラウドソーシングにおける共同創作タスク	4
第3章	マイクロタスク型クラウドソーシングによる線画イラスト生成 [8]	5
3.1	提案手法	5
3.2	システム設計	6
3.3	実装	6
第4章	提案手法の実現可能性および有効性の検証	9
4.1	線画イラスト生成実験 [8]	9
4.1.1	マイクロタスクの時間の設定	9
4.1.2	条件の設定	9
4.1.3	参加者と実験環境	10
4.1.4	手続き	11
4.1.5	完成画像の判定	12
4.2	線画イラスト作成手法の特性 [9]	12
4.2.1	実験条件	13
4.2.2	参加者と実験環境	14
4.2.3	手続き	14
4.2.4	完成画像の判定と各条件間での第三者比較	14
	完成画像の判定	14
	複数条件と単独条件に関する比較	15
	原図の提示位置に関する比較	15
4.2.5	データ収集	15
第5章	結果	19
5.1	線画イラスト生成実験 [8]	19
5.1.1	描画結果	19
5.1.2	完成画像	19

5.2	線画イラスト作成手法の特性調査実験 [9]	20
5.2.1	描画結果	20
5.2.2	複数条件での完成画像	20
5.2.3	複数条件と単独条件との完成度の差	20
5.2.4	原図重ね置き条件と原図隣置き条件での比較	20
5.2.5	筆数と描画時間	21
第6章	考察 [8, 9]	35
6.1	タスク補助機能の有効性	35
6.2	複数条件と単独条件の比較	35
6.3	原図の提示位置に関する比較	36
6.4	タスクの所要時間	36
6.5	初期の描画が及ぼす影響	37
6.6	ペンタブレットについて	37
6.7	マイクロタスクの時間制限の効果	37
6.8	タスクの実行結果の質と不真面目なクラウドワーカへの対応	38
6.9	実用イラスト生成に向けて	38
6.10	本手法を用いることによるコストの削減	39
第7章	まとめ	40
	謝辞	41
	参考文献	42

目次

3.1	提案手法	6
3.2	0番目のワーカを含めた作業の流れ	7
3.3	システムの構成図	8
4.1	実験で使った画像	10
4.2	実験の様子	11
4.3	(左)システムの説明・練習ページ / (右)ワーカに提示される画面	12
4.4	実験タスクのフローチャート	17
4.5	原図を描画領域の隣に置いたインタフェース	18
5.1	各ワーカの描画内容 (輪郭抽出なし条件): 赤枠は評価者によって完成と判定された画像を示す	22
5.2	各ワーカの描画内容 (輪郭抽出あり条件): 赤枠は評価者によって完成と判定された画像を示す	23
5.3	線画イラストの生成過程 (輪郭抽出なし条件): 赤枠は評価者によって完成と判定された画像を示す	24
5.4	線画イラストの生成過程 (輪郭抽出あり条件): 赤枠は評価者によって完成と判定された画像を示す	25
5.5	完成と判定された線画イラスト (上: 輪郭抽出なし条件, 下: 輪郭抽出あり条件)	26
5.6	複数重ね条件 (第1グループ)での描画の様子: 上段が各ワーカの描画内容, 下段が線画イラストの作成過程	27
5.7	複数重ね条件 (第2グループ)での描画の様子: 上段が各ワーカの描画内容, 下段が線画イラストの作成過程	28
5.8	複数隣置き条件 (第1グループ)での描画の様子: 上段が各ワーカの描画内容, 下段が線画イラストの作成過程	29
5.9	複数隣置き条件 (第2グループ)での描画の様子: 上段が各ワーカの描画内容, 下段が線画イラストの作成過程	30
5.10	単独重ね条件 (第1グループ)で作成された線画イラスト	31
5.11	単独重ね条件 (第2グループ)で作成された線画イラスト	32
5.12	単独隣置き (第1グループ)で作成された線画イラスト	33
5.13	単独隣置き (第2グループ)で作成された線画イラスト	34

5.14	複数条件での完成画像：左側が第1グループ，右側が第2グループで，左列が原図重ね置き条件，右列が原図隣置き条件での完成画像	34
6.1	多様性の例. 左：原図，中央：ヒレの形が原図より丸い例，右：体が原図より細い例	36

第1章 はじめに

雑誌，書籍，広報誌等，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。これまでイラストを必要とするクライアントは，一般的にイラストレーターと呼ばれる専門性の高い職に就く者に構図や配色などのイラストに対する要求を伝え，イラストの作成を依頼した。そして比較的高価格なイラストが扱われてきた。

また文書に，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 [1]，イラストの需要は大きい [2]。しかしそれらのすべてがイラストレーターが作成する高価格なもの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，Webサイトから無料で入手できるイラストを使用することも多い [3, 4, 5]。昨今このような無料イラストサイトが広まる一方で，それらのサイトで入手できる既存イラストには種類に限りがあり，構図などの面で，クライアントが本当に欲しいイラストを得ることができるとは限らない。すなわち，イラストレーターと専属契約を結んだり，アウトソーシング，クラウドソーシングなどで契約を締結したりすることで入手できる高価格なイラストでもなく，種類に限りのある無料で入手可能なイラストでもない，クライアントの要求を満たしつつも価格を抑えたイラストを入手する方法が現状では無い。

本研究では，そのようなクライアントの要求を満たしながらも，イラストレーターと契約を結ぶようなコストがかからず，イラストの専門家が作成に携わることがなくてもイラストを作成できる手法を提案する。本手法では，描画能力に関わらずどんな人でもイラストを作成できるプラットフォームが必要となる。また，作業員1人にイラスト作成の作業を託すことは，その作業員に対して大きな負担をかけてしまうため避ける必要がある。

作業員の負担を軽減する方法として，本研究ではタスク全体を細分化して複数の作業員に提示するマイクロタスクの手法を用いる。マイクロタスクは通常多くの作業員を必要とするので，マイクロタスクを既存のクラウドソーシングサイト等に登録することでタスクの遂行が可能となる。一方で既存のクラウドソーシングでは，コンペティション方式などでイラスト作成の技能を持つワーカーにイラスト作成を依頼するタスクが多く存在する [6]。本研究では従来の方式とは異なり，イラスト作成の技能に依らず，どんな作業員でもタスクに従事できる手法を提案する。

イラストを作成するためには通常，線画作成や彩色など，複数の工程を必要とする。そこで本研究では，イラストの土台となり被写体を型どる「線画」が，そのまま1枚のイラストとなる「線画イラスト」の生成手法に着目した。また絵を描く，という行為の中でも模写に焦点を当て，どんな作業員でも線画イラストの作成が可能になる手法を用意する。模写では写真などの原図を忠実に描き写すことになるが，さらに作業の難易度を下げるために，原図

をなぞる形で線を描くインタフェースを提示する。加えて1人あたりの作業負荷を低く抑えるために、作業員1人分の作業時間を制限することで、1枚の線画イラストを作成するというタスクをマイクロタスク化する。本研究では、このマイクロタスクをクラウドソーシングにより実行する線画イラスト生成手法を提案する [7, 8].

本稿ではまず、本手法の実現可能性を検証するために、線画イラストを実際に作成した実験についてを述べ、本手法によって線画イラストの生成が可能であることを示す。また、この実験を通じて観察された、本手法を用いることで生成される線画イラストの性質についても考察する。次に本手法によって作成される線画イラストについて、その線画イラストの特徴と、より詳細な手法の性質を検討するための実験について述べ、マイクロタスクを用いて線画イラストを生成する場合でも、1人で線画イラストを作成する従来手法と遜色のない完成度の線画イラストを生成できることが示唆されたことを示す。また、提案手法のように線画イラストの元となる原図を描画領域に重ねて提示する場合と、原図を描画領域に重ねずに提示する新しい原図の提示方法を比較し、新たな原図提示方法により、描画される線に多様性が生まれ、原図とは多少離れた線画イラストが生成できることが確認されたことを述べる [9].

第2章 関連研究

2.1 複数のワーカによるタスクの実行と報酬の有無

マイクロタスク型のクラウドソーシングのプラットフォームを提供する既存のサービスとして Amazon Mechanical Turk (Mturk) [10] がある。Mturk では、マイクロタスクを提示する Requester とそのタスクを実行する 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。一方で、複数のユーザがボランティアとして参加する、分散コンピューティング技術を利用した SETI@home がある。ユーザはソフトウェアをダウンロードし、そのプログラムを実行させるだけで、従来スーパーコンピュータが解析してきたデータの一部の計算処理を行うことができる。1999 年から開始されたこのプロジェクトは、ユーザに対する報酬が一切ないにも関わらず、現在も稼働し続けている [11]。

ユーザがワーカとしてタスクを行う場合について、報酬の有無による違いを検討した報告がある。Mturk 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 Worker 一人につき、一時間あたりおよそ 6 ドルと少ない [12, 13]。Martin らは、報酬は Worker にとって最も重要な要因と述べているが [14]、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、ワーカ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、数秒程度で終わるタスクを提示する。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、ワーカに大きな負担をかけることなく情報を収集できることが明らかになった [15]。また Winter らは、報酬がワーカに与える影響は、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 [16]。

2.2 クラウドソーシングを利用した複雑なタスクの実行

マイクロタスク型クラウドソーシングでは一般に、画像のラベリングやデザインのフィードバックなどといった、単純で独立したタスクを行う場合が多い。Kittur らは、複雑で相互依存性のあるマイクロタスクを実行できるよう、分散コンピューティング技術の概念に基づき、タスクをサブタスクに分類したフレームワークを示した [17]。

このサブタスクの概念を利用したクラウドソーシングとして、Ambati らは 3 段階の翻訳ワークフローを提案した。1 段階目では、単語もしくは句の翻訳を行い、2 段階目では 1 段階目の

結果から、翻訳する文の言語が非母語であるバイリンガルが翻訳を行う。そして3段階目で、2段階目までの結果を見ながら、翻訳する文の言語が母語であるワーカが翻訳を行う。この3段階に分けた翻訳手法は、既存の手法よりコストを削減できたことを明らかにしている [18]。また、Soylent と呼ばれる文章作成アプリケーションでは、文章を要求に応じてより短くしたり、校正したりすることができる。ユーザが入力し選択したテキストは、Mturk のワーカに表示される。Mturk のワーカははじめに、表示されたテキストの間違ひを見つける。その後、見つかった間違ひを含むテキストに対し、修正もしくは書き直しを行う。修正や書き直しの案は Mturk のワーカによって何例か提示され、さらに別のワーカが、修正案の中で最も適切なものを選び、修正内容として適用する。Mturk のワーカによって修正が適用され、変更された文章は、逐次ユーザの画面に表示される [19]。

2.3 クラウドソーシングにおける共同創作タスク

クラウドソーシングを用いて、複数のワーカが共同で実行する創作タスクについての報告が挙がっている。Yu らは、ワーカがデザイン案をスケッチし、スケッチ案どうしを組み合わせるという方法で新しいデザインを作成する手法を提案した。最初に1段階目として、提案システムのワーカがデザイン案のスケッチを行い、作成されたスケッチを Mturk のワーカが評価する。次に2段階目として提案システムのワーカは、評価された1段階目のデザイン案を組み合わせ、新しいデザイン案のスケッチを行う。2段階目のデザイン案も Mturk のワーカによって評価され、同様に3段階目もスケッチを行う。1段階目と3段階目のデザイン案を比較した結果、創造性とオリジナル性の面で、提案手法は有効であることが確かめられた [20]。

また、Kawashima らはアメリカ合衆国の100ドル紙幣を1万人のクラウドワーカによって再描画するタスクを実行した。この再描画タスクでは、100ドル紙幣を1万のエリアに分割し、Mturk で募集したワーカが、割り当てられたエリアを専用のオンラインツールを用いて描画する。1万人のワーカにはそれぞれ1セントが報酬として与えられた。またワーカには、この描画タスクが100ドル紙幣を再描画するというタスクの一部であることを伝えなかったが、結果として100ドル紙幣の再描画は完成した [21, 22]。

第3章 マイクロタスク型クラウドソーシングによる線画イラスト生成[8]

3.1 提案手法

本研究は1枚の写真(原図)から、線画イラストを生成するための手法を提案する。通常、線画を作成する際は物体の輪郭など、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輪郭線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線を描くという行為をマイクロタスクとして多数の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イラストを生成できる。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枚の線画イラストを生成できる仕組みを考える必要がある。

提案手法の概念図を図3.1に示す。このマイクロタスクでは、ワーカー一人一人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。各ワーカーには原図と線を描く描画領域が重ねて提示される。そして1番目のワーカーから順に原図の輪郭線を模写するように線を描いていく。2番目以降のワーカーには、それまでのワーカーが描いてきた線が全て原図に重ねられ、その様子を見ながら線画イラストとして不足している線を描き加えていく。この線を描き加える作業がマイクロタスクとしてワーカーに提示され、このタスクを多数のワーカーがこなすことで、最終的に全ての描画線を重ねて結合した画像が、原図の線画イラストとして完成する。

また、線画を描く作業を全てワーカーに委ねて生成を開始してしまうと、膨大な作業量と人手を要する。そこで図3.2に示すように、コンピュータを0番目のワーカーとみなして原図に対し画像処理技術による輪郭抽出を行い、抽出した線を0番目のワーカーの描画線として原図に重ねて提示する。これにより、線を描画するワーカー数を削減できると見込まれる[7, 8]。

一方で2.3節で述べたように、クラウドソーシングによる共同創作タスクによる研究がある。Yuらの研究は、Mturkによってデザイン案の評価を受けているものの、実際にデザインを作成するのは1人である。本研究はYuらとの研究とは異なり、1枚の線画イラストを複数人が共同で作成する手法について検証する。また、Kawashimaらの研究はマイクロタスクによって1枚の100ドル紙幣を複数人で再描画するタスクであるが、1人で再描画を行った際との比較検討を行っていない。本研究では、マイクロタスクによる手法と従来通り1人で線画イラストを作成する手法について、その違いも検討する。

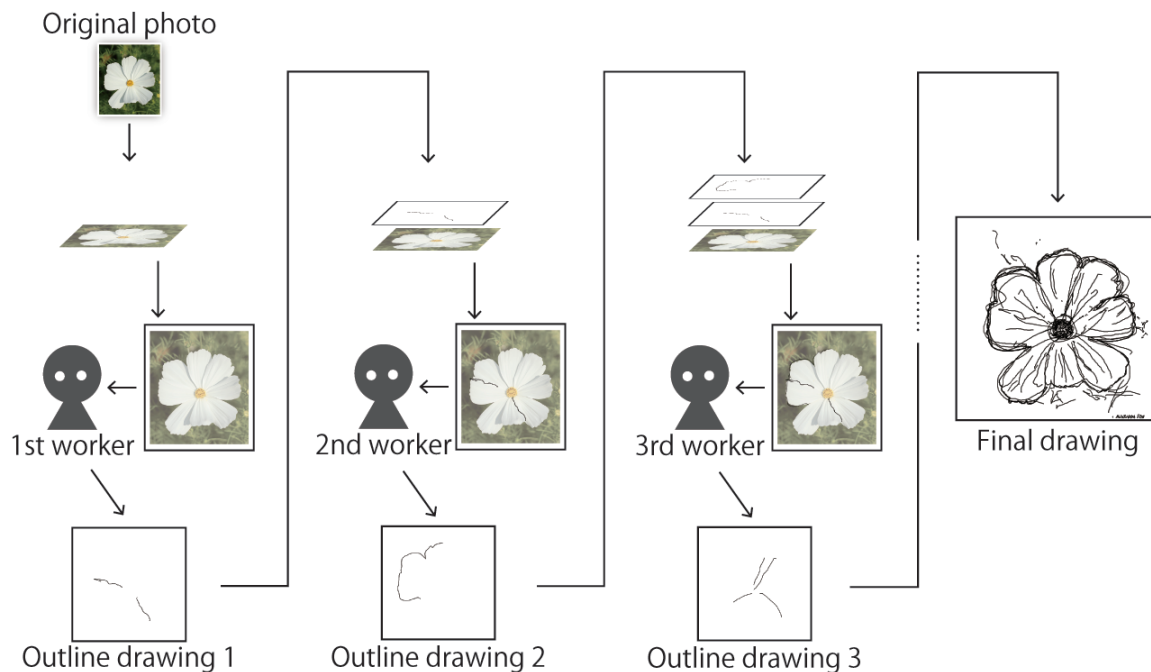


図 3.1: 提案手法

3.2 システム設計

提案手法による線画イラスト生成システムの構成を図 3.3 に示す。まず、クライアントが線画イラスト化したい原図となる写真をシステムに登録する。ここで、システムが 0 番目のワーカとして輪郭線の自動抽出を試みる。その後、ワーカがマイクロタスクによって、線画イラストとして不十分な部位に加筆していく。ワーカ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ていくが、次のワーカにタスクを提示する際に、原図とそれまでのワーカの画像は全て重ねて表示される。最終的に全ての描画線を重ねて結合することで線画イラストが完成する。

3.3 実装

提案手法を Web ベースのシステムとして実装した。システム全体は次の 4 モジュールで構成されている。

1. 原図の受付
2. 輪郭の自動抽出
3. タスク提示

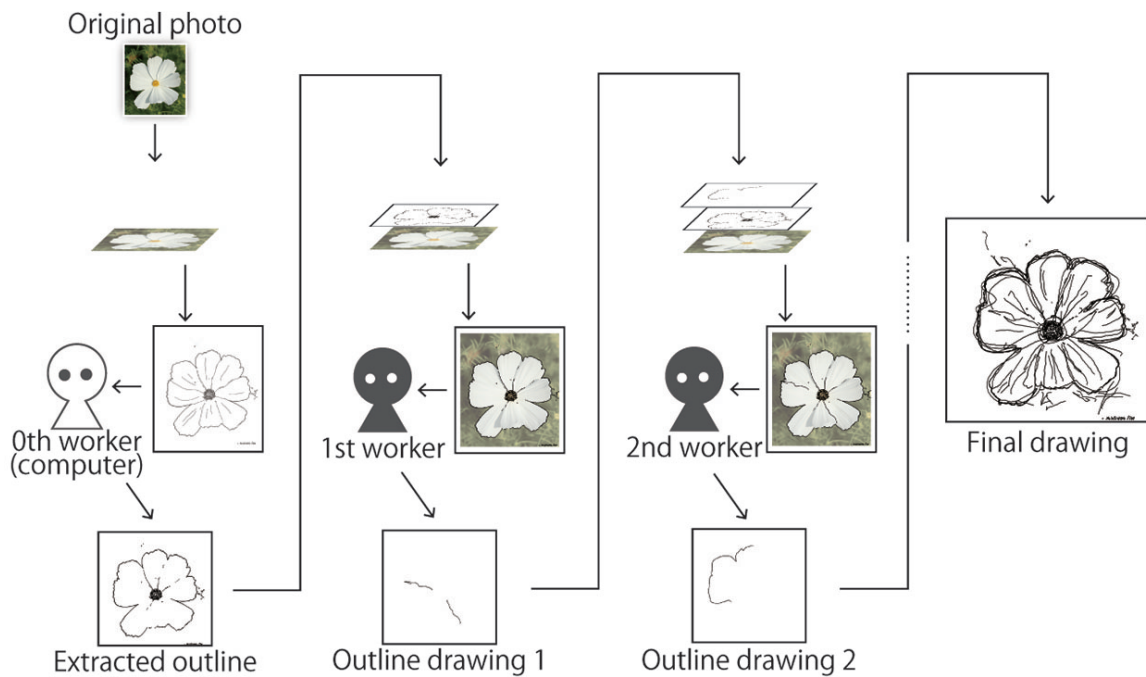


図 3.2: 0 番目のワーカを含めた作業の流れ

4. 生成画像の提供

モジュール (1) では PHP を利用し、クライアントから原図を受け付ける。アップロードされた画像は保存され、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の OpenCV を用い、モジュール (1) から受け取った原図の輪郭を抽出する。抽出した画像は元画像と紐付けられ、ワーカへ提示される。

モジュール (3) では PHP を用いてマイクロタスクを提示する。提示の際は HTML5 の canvas 要素により、画像が複数のレイヤに表示される。最下層にクライアントからアップロードされた原図を透明度 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が作成した画像を順に重ねる。ワーカが一番上のレイヤに線画を描く。そのため、ワーカは原図をなぞりつつ、かつそれまでのワーカが描いてきた線に加筆することで線を描画できる。描画機能は JavaScript が提供する。またワーカが描画を終了後、その描画線を保存する。

モジュール (4) では、モジュール (3) までに作成された描画線を全て重ね合わせ、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。

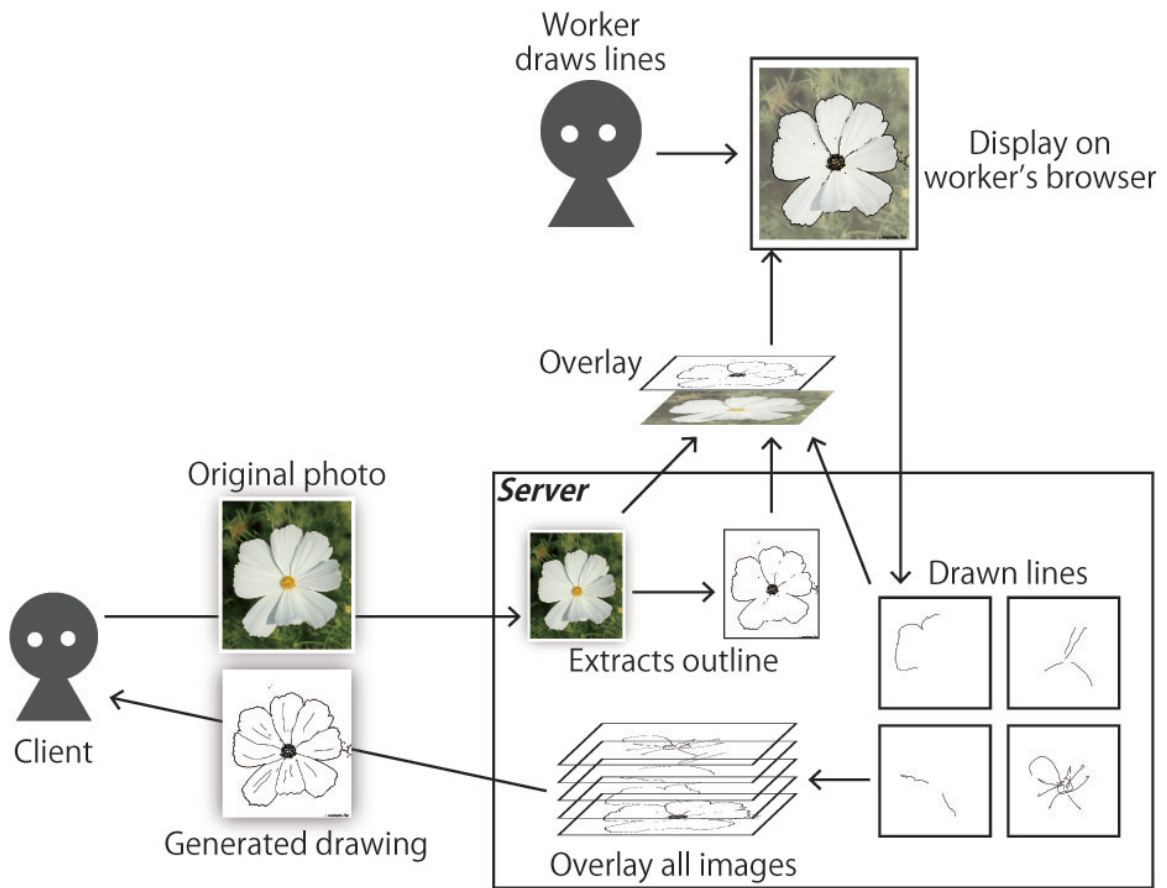


図 3.3: システムの構成図

第4章 提案手法の実現可能性および有効性の 検証

1章で述べたように、本研究では提案手法について実現可能性と有効性を検証するための実験として、実施した2つの実験について述べる。

4.1 線画イラスト生成実験 [8]

本手法を用いて線画イラストを完成できるかどうか、手法の実現可能性を確認するために、線画イラストの生成実験を行った。さらに3.1節で述べたように、提案手法を改善するための、タスクの補助機能についても検討した。

4.1.1 マイクロタスクの時間の設定

本手法ではマイクロタスクを各ワーカに提示する。マイクロタスクでは、1人あたりの作業量は少なくする必要がある。本手法における作業量とは描画する線の総量、すなわち描画量となるが、描画量はその時点での線画イラストの完成度による変動が大きい。そのため実際に描画しなければならない線の分量がワーカによって変わってしまい、ワーカ全員に同等のタスク量を振り分けられず、適切なタスク分割にならないと考えられる。そこで本手法のマイクロタスクでは、描画量ではなく描画時間を制約した。これを、ワーカ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。本実験について事前知識を持たない筑波大学の大学生および大学院生である予備実験参加者4名（男性3名、女性1名）が、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枚の線画の生成作業を試行した。その後、マイクロタスクの作業として負担を感じず、かつ何がしかの線を描くだけの余裕があることを考慮し、線の描画に最適と考える時間を尋ねた。その結果、参加者全員が最適と考えた秒数の平均は6.8秒（標準偏差1.3秒）であったため、1タスクの所要時間を7秒とした。

4.1.2 条件の設定

本手法の実現可能性を検討するために、コンピュータによる原図の輪郭抽出がないシステムを用い、1番目のワーカから線画の描画を始めるタスクを用意する。このタスクを実施する

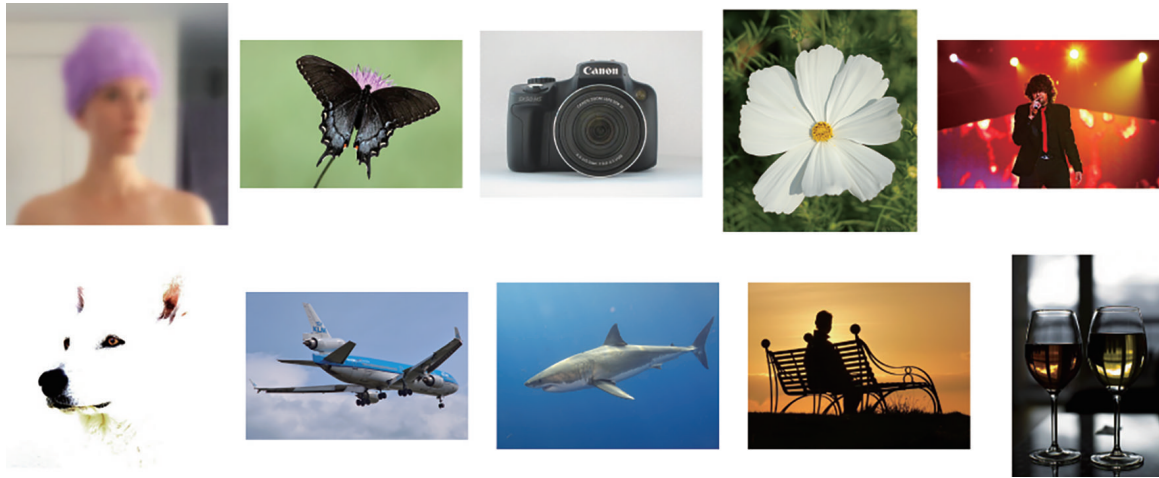


図 4.1: 実験で使用した画像

ことで、複数人のワーカによって1枚の線画イラストが完成することを確認する。また、タスクの補助機能の有効性を検証するために、コンピュータによる輪郭抽出を用いて、0番目のワーカから線画の生成を始めるタスクも用意する。コンピュータによる原図の輪郭抽出がない場合と、輪郭抽出がある場合を比較することで、タスク補助機能の有効性を検討できる。

以上より、線画イラストの生成実験では、コンピュータによる輪郭抽出のない「輪郭抽出なし条件」と、輪郭抽出のある「輪郭抽出あり条件」の2条件について実験を行った。また、共に原図として使用した画像は図 4.1 に示す 10 枚である。これらは、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選定した。

4.1.3 参加者と実験環境

線画イラスト生成実験は、4.1.1 節における予備実験に参加しておらず、本実験について事前知識を持たない筑波大学の大学生、大学院生、および教職員に対して、大学構内で参加者を募った。輪郭抽出なし条件では 27 名（男性 20 名、女性 7 名）、輪郭抽出あり条件では 29 名（男性 25 名、女性 4 名）がワーカとして実験に参加し、条件間で参加者の重複はなかった。また参加者への報酬はなかった。

また入力インターフェースとして 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。また図 4.2 のように、実験では全員椅子に着席し、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。なお、入力インターフェースにペンタブレットを使用したのは、線画を描くというタスクにおいて、直感的な操作が可能であると判断したためである。ペンタブレットを触ったことがない実験参加者に配慮して、実験開始前には十分にペンタブレットを用いた描画の練習を行った。



図 4.2: 実験の様子

4.1.4 手続き

実験開始前に図 4.3 左に示す、説明用のページを参加者に提示した。この画面を表示しながら、参加者に次の内容を説明した。

- これから 10 枚の写真の模写を行ってもらうこと。また完成図として、説明ページ左に示される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
- 写真の一部のみを描画すればよく、急ぐ必要はないこと
- 線を描く前に、提示された画像が線画イラストとして完成されていると判断した場合、実験画面（図 4.3 右）中の「これで完成!」ボタンを押すこと

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 4.3: (左) システムの説明・練習ページ / (右) ワーカーに提示される画面

図 4.4 は実験において 1 人の参加者が取り組むタスクのフローチャートである。1 枚ずつランダムな順に図 4.1 に示した画像が提示され、参加者は線を描くか、「これで完成!」ボタンを押すかした。

本実験では、提案手法の可能性と限界を知るうえで重要な、原図の違いによる生成されるイラストの違いや生成の可否について検証を行う。また提案手法を補助する手法として、コンピュータによる原図の輪郭抽出の利用可能性を検討しており、この機能の有無の比較を行う。以上の目的を達成するため、原図の種類間での参加者の参加順序は変更していない。

また、10 枚全ての画像の描画終了後、参加者に対しコメントを聴取した。

4.1.5 完成画像の判定

実験終了後に、予備実験及び線画イラスト生成実験に参加しておらず、また本研究の事前知識を持たない筑波大学の大学生および大学院生 5 名（男性 3 名、女性 2 名）が、実験で描画されたすべての画像を見ながら、完成画像を判定した。判定は 5 名が独立で行い、完成画像が何回目に描画されたものかを確認した。その回数の平均値 n （小数点以下を四捨五入）を算出し、 n 回目に描画された画像を完成画像とした。なお、4.1.4 節に記したように、実験参加者も提示された線画イラストについて完成か否かの判断をしたが、これは生成過程におけるものであるため、別途このような判定をさせた。

4.2 線画イラスト作成手法の特性 [9]

続いて、より詳細な手法の性質を検討する特性調査実験を行った。まず第一に、本手法による線画イラストと 1 人による線画イラストを比較する。第二に、原図をワーカーが線を描画

する描画領域に重ねて提示する場合と、原図を描画領域に重ねずに提示する場合を比較する。以上の2点を調査するために、次のように実験の条件を設定した。

4.2.1 実験条件

本手法による線画イラストと、1人による線画イラストを比較するために、2つの条件を設定した。まずはマイクロタスクで複数人のワーカが1枚の線画イラストを作成する複数条件、もう1つは、同じシステムを用いて1人のワーカが1枚の線画イラストを作成する単独条件である。複数条件では、4.1.1節によるマイクロタスクの時間の設定に則り、ワーカ1人につき7秒の制限時間を与え、ワーカはその時間内で1枚の原図に対し線を描画する。またワーカが、提示された画像が既に線画イラストとして完成していると判断した場合、「これで完成!」ボタンを押すことでその画像への描画を取りやめることができる [7, 8]。一方単独条件では、ワーカには原図と空白の描画領域が提示される。複数条件でワーカに対し与えられた1枚7秒という制限時間は、単独条件では廃止し、ワーカは自分が完成と判断するまで線を描画する。

次に、原図を描画領域に重ねて提示するか、重ねずに提示するかによって、作成される線画イラストの違いを検討するために、2つの条件を設定した。まずは原図を描画領域に重ねて提示して線画イラストの作成を行う原図重ね置き条件であり、もう1つは原図を描画領域には重ねず、描画領域の隣に提示して線画イラストの作成を行う原図隣置き条件である。原図重ね置き条件では、図 4.3 右に示したように、ワーカは原図をなぞる形で線画を描画できる。一方原図隣置き条件では、図 4.5 のように描画領域の隣に置くインタフェースをワーカに提示した。ワーカは原図を横に見ながら線を描画することになり、原図重ね置き条件のように、原図をなぞって線を描くことはできない。

以上より、次の4条件をランダムな順に参加者に提示する被験者内計画で実施した。

1. 複数条件かつ原図重ね置き条件 (複数重ね条件)
2. 複数条件かつ原図隣置き条件 (複数隣置き条件)
3. 単独条件かつ原図重ね置き条件 (単独重ね条件)
4. 単独条件かつ原図隣置き条件 (単独隣置き条件)

いずれの条件においても、提示する原図は図 4.5 左側にあるサメの写真1枚を使用した。複数条件で生成される線画イラストの数を増やすため、この特性調査実験を、別の参加者群で2回実施した。

ここで、提案手法により複数人で線画イラストを作成した場合と、1人のワーカが線画イラストを作成した場合の比較では複数重ね条件と単独重ね条件、および複数隣置き条件と単独隣置き条件の結果を、原図の提示位置に関する比較では複数重ね条件と複数隣置き条件、および単独重ね条件と単独隣置き条件の結果を、それぞれ比較する。

4.2.2 参加者と実験環境

線画イラスト作成の特性調査実験は1回目（第1グループ）、2回目（第2グループ）ともに20名ずつがワーカとして参加した。第1グループは男性13名、女性7名、平均年齢は23.0歳、年齢の標準偏差は4.7歳であった。第2グループは男性16名、女性4名、平均年齢は20.7歳、年齢の標準偏差は1.9歳であった。両グループで参加者の重複はなく、また実験方法に違いはなかった。いずれの参加者へも報酬はなかった。

また実験環境として、4.1.3節で述べたものと同じ環境で実験を行った。

4.2.3 手続き

実験開始前には、図4.3に示される、システムの説明が記載された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。

- これから提示される1枚の写真について、描画領域に写真の模写を4回行ってもらうこと。また完成図として、図4.3左の写真のような模写を目指すこと
- 4回の模写のうち、2回はそれぞれ7秒の制限時間があること。ただし描画は写真の一部のみでよく、急ぐ必要はないこと。また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、ページ上の「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
- 4回の模写のうち、他の2回は制限時間がないので、自分が完成だと思うまで線画を作成すること
- 制限時間がある模写かそうでないかは、4回の描画前にそれぞれ実験者から指示があるので、その指示に従うこと

説明中の4回の模写とは、4.2.1節の4条件を指す。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。タスクでは4条件がランダムな順に提示され、実験者の指示に従い線画イラストの作成タスクを行った。

4.2.4 完成画像の判定と各条件間での第三者比較

完成画像の判定

実験終了後に、複数条件について、各画像の完成画像を判定した。本研究の事前知識を持たない評価者5名（いずれも男性、平均年齢30.0歳、標準偏差9.9歳）が、実験で描画されたすべての画像を見ながら、完成画像を判定した。その後完成画像の描画回数の平均値 n （小数点以下を四捨五入）を算出し、 n 回目に描画された画像を完成画像とした。

複数条件と単独条件に関する比較

本手法を用いて線画イラストを生成する場合と、1人のワーカが線画イラストを作成する場合の違いを検討するために、複数条件と単独条件とで作成された線画イラストの第三者評価を行った。評価は完成画像を判定した評価者とは別の、本研究の事前知識を持たない評価者5名（男性1名、女性4名、平均年齢23.6歳、標準偏差1.2歳）が独立で行った。

まず評価者には、同じグループ、同じ原図の提示位置で作成された複数条件の画像1枚と単独条件の画像1枚が提示される。この2枚のどちらが完成度が高いかを、同時に提示された原図に基づいて比較する。評価者は、「複数条件で生成された線画イラストの方が完成度が高い」「複数条件で生成された線画イラストも単独条件で作成された線画イラストも同じ程度の完成度」「単独条件で作成された線画イラストの方が完成度が高い」の3段階で回答した。これを1試行とし、この試行を同グループ内で作成された20枚の単独条件の画像それぞれについて行った。残りのグループ、残りの原図の提示位置についても同様の評価を行ったため、1人の評価者はこの評価を、合計80試行実施した。なお、実際には評価者に、どの画像がどの条件で生成された画像かは伝えていない。

原図の提示位置に関する比較

原図の提示位置による違いを検討するために、原図重ね置き条件と、原図隣置き条件とで作成された線画イラストの第三者評価を行った。評価には完成画像の判定、および前述の第三者評価のいずれにも参加していない、本研究の事前知識を持たない評価者5名（いずれも男性、平均年齢19.4歳、標準偏差0.8歳）が独立で行った。

まず評価者には、同一の実験参加者が作成した、単独重ね条件での線画イラスト1枚と、単独隣置き条件での線画イラスト1枚が提示される。この2枚のどちらが原図に近いかを、同時に提示された原図に基づいて比較する。評価者は、「原図重ね置き条件で生成された線画イラストの方が原図に近い」「原図隣置き条件で生成された線画イラストの方が原図に近い」の2段階で回答した。これを1試行とし、この試行を40枚の単独条件の画像それぞれについて行った。また、同一のグループで生成された複数条件での線画イラストにも同様の評価を行ったため、1人の評価者はこの評価を、合計42試行実施した。なお前述の評価同様、実際には評価者に、どの画像がどの条件で生成された画像かは伝えていない。

4.2.5 データ収集

この特性調査実験では2グループ、計40名が実験に参加したため、表4.1に示す枚数の線画イラストを収集した。また描画終了後には口頭にて実験の感想を聴取した。単独条件では、描画を終了するまでに要した筆数を記録した。

これに加え、描画に要した筆数と描画時間の関係を調べる実験を行った。線画イラストの特性調査実験に参加していない5名（男性3名、女性2名、平均年齢21.6歳、標準偏差1.5歳）が、単独条件での実験同様、筆数を測定し、さらに描画を行った時間を測定した。

表 4.1: 実験で取得した線画イラストの枚数

	重ね置き条件	隣置き条件
複数条件	2	2
単独条件	40	40

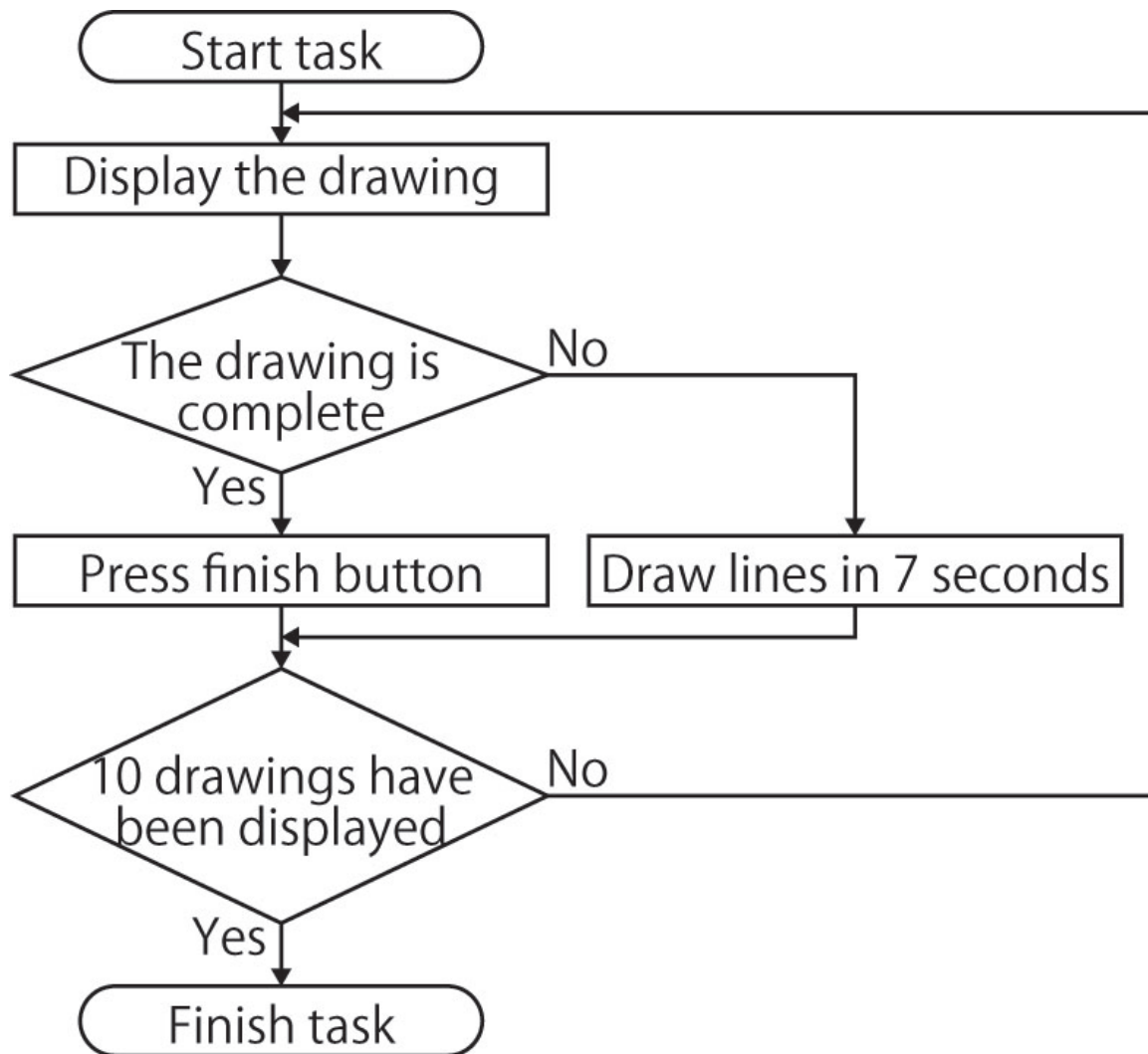
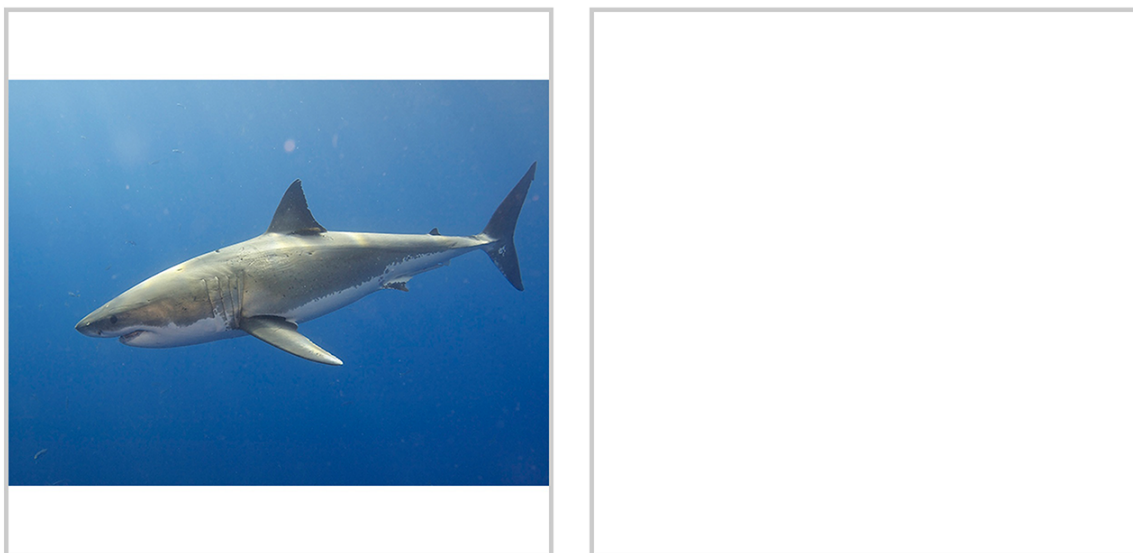


図 4.4: 実験タスクのフローチャート

下の写真を模写してください



これで完成!

図 4.5: 原図を描画領域の隣に置いたインタフェース

第5章 結果

5.1 線画イラスト生成実験 [8]

5.1.1 描画結果

輪郭抽出なし条件と輪郭抽出あり条件のそれぞれで、各ワーカが各原図に対してどのような線を描画したかを、図 5.1, fig:ind1-2 に示す。また、これらを重ね合わせ、線画イラストが生成されてゆく過程を図 5.3, 5.4 に示す。それぞれ、左の列に原図を示し、右側の描画線は左上から右下に向かって順に並べた。画像によって枚数が異なるのは、ワーカが描画を行わず「これで完成!」ボタンを押し、線画が完成したと判断した場合が含まれるためである。

5.1.2 完成画像

図 5.5 に、各条件における各原図の線画について、4.2.4 節で述べた評価者 5 名によって完成と判定された画像を示す。以下に完成と判定された線画イラストについて、5 名の平均値と標準偏差を示す。

輪郭抽出なし条件では 17 回 (3.0), 19 回 (1.8), 17 回 (3.6), 20 回 (3.9), 20 回 (2.4), 19 回 (3.2), 18 回 (2.7), 16 回 (3.0), 18 回 (4.9), 20 回 (3.1), 輪郭抽出あり条件では始めにコンピュータによって抽出された画像を 0 回目として 20 回 (1.6), 8 回 (4.2), 15 回 (7.3), 16 回 (5.3), 21 回 (6.4), 18 回 (4.5), 14 回 (2.5), 18 回 (3.9), 11 回 (7.1), 17 回 (7.9) であった。これらの完成画像は図 5.1, 5.2 および図 5.3, 5.4 の赤枠の画像である。

ばらつきはあるものの、平均して輪郭抽出なし条件ではおよそ 18 回、輪郭抽出あり条件ではおよそ 16 回の描画が行われたときに、線画が完成したと判定された。つまり輪郭抽出なし条件の方が、輪郭抽出あり条件よりも完成と判定された画像の描画回数が多い。

また描画結果から、コンピュータによる輪郭抽出がない場合も、また図 4.1 の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場合も、生成が進むにつれて輪郭線が描かれる様子が見られる。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生成は十分可能であることが分かる。

表 5.1: 複数条件と単独条件とでより完成度が高いと評価された回数

	複数条件	単独条件	どちらも同程度
原図重ね置き条件	148	14	38
原図隣置き条件	86	67	47
合計	234	81	85

5.2 線画イラスト作成手法の特性調査実験 [9]

5.2.1 描画結果

各グループにおける複数条件での描画の作成過程をそれぞれ図 5.6, 5.7, 5.8, 5.9 に示す。各々の上段が各ワーカの描画内容、下段がそれらを重ね合わせた作成過程を示しており、いずれも左上から右に向かって順に並んでいる。各描画線の左上の数字は通し番号である。また図 5.10, 5.11, 5.12, 5.13 は単独条件において全参加者 40 名が作成した線画を示している。

5.2.2 複数条件での完成画像

複数条件で描画された線画について、4.2.4 節で述べた評価者によって完成と判定された画像を、図 5.14 に示す。左側の 2 枚が第 1 グループによって生成された画像であり、右側の 2 枚が第 2 グループによって生成された線画イラストである。また、それぞれのグループの左側の画像が原図重ね置き条件、右側の画像が原図隣置き条件で生成された線画イラストである。以下に、完成した線画イラストの描画回数と標準偏差を示す。第 1 グループでは、原図重ね置き条件で 8 回 (1.2)、原図隣置き条件で 6 回 (0.6)、第 2 グループでは、原図重ね置き条件で 10 回 (1.3)、原図隣置き条件で 12 回 (1.7) であり、平均すると 9.0 回 (2.7) の描画で線画イラストが完成することが分かった。

5.2.3 複数条件と単独条件との完成度の差

複数条件と単独条件における線画イラストの完成度比較評価の結果を表 5.1 に示す。この結果から、原図の提示位置によらず、複数によって生成される線画イラストの完成度は、従来手法のように 1 人で作成する線画イラストと遜色がないことが示唆された。

5.2.4 原図重ね置き条件と原図隣置き条件での比較

原図の提示位置による違いについて、単独条件で同じ人物が描いた線画イラストと複数条件で同じワーカ群が生成した線画イラストについて、原図重ね置き条件と原図隣置き条件のどちらがより原図に近い線画イラストとなったかを評価した。単独条件で作成された 40 枚と

表 5.2: 描画の筆数と所要時間. 上: 原図重ね置き条件, 下: 原図隣置き条件

原図重ね置き条件 参加者 ID	所要時間 (秒)	所要筆数 (筆)	1 筆あたりの描画時間 (秒)
1	51.3	9	5.7
2	84.1	20	4.2
3	47.6	7	6.8
4	36.1	2	18.0
5	60.8	9	6.8
合計/平均	279.9	47	6.0

原図隣置き条件 参加者 ID	所要時間 (秒)	所要筆数 (筆)	1 筆あたりの描画時間 (秒)
1	98.0	10	9.8
2	168.3	49	3.4
3	65.8	14	4.7
4	91.8	19	4.8
5	90.7	12	7.6
合計/平均	514.6	104	4.9

複数条件で生成された 2 枚, あわせて 42 枚の線画イラストのうち, 原図を描画領域に重ねて置いた場合の方が原図に近いと判定されたのは 154 回, 隣に置いた方が原図に近いと判定されたのは 56 回であった. この結果から, 原図を描画領域に重ねると, 原図を描画領域の隣に置いた場合に比べ, より原図に忠実な線画イラストが作成されることが分かった.

5.2.5 筆数と描画時間

描画の際の筆数と描画時間について, 4.2.5 節で述べた実験の結果を表 5.2 に記す. また, 全ての所要時間と所要筆数から 1 筆あたりの描画時間を算出すると 5.3 秒となった.



図 5.1: 各ワークの描画内容 (輪郭抽出なし条件): 赤枠は評価者によって完成と判定された画像を示す

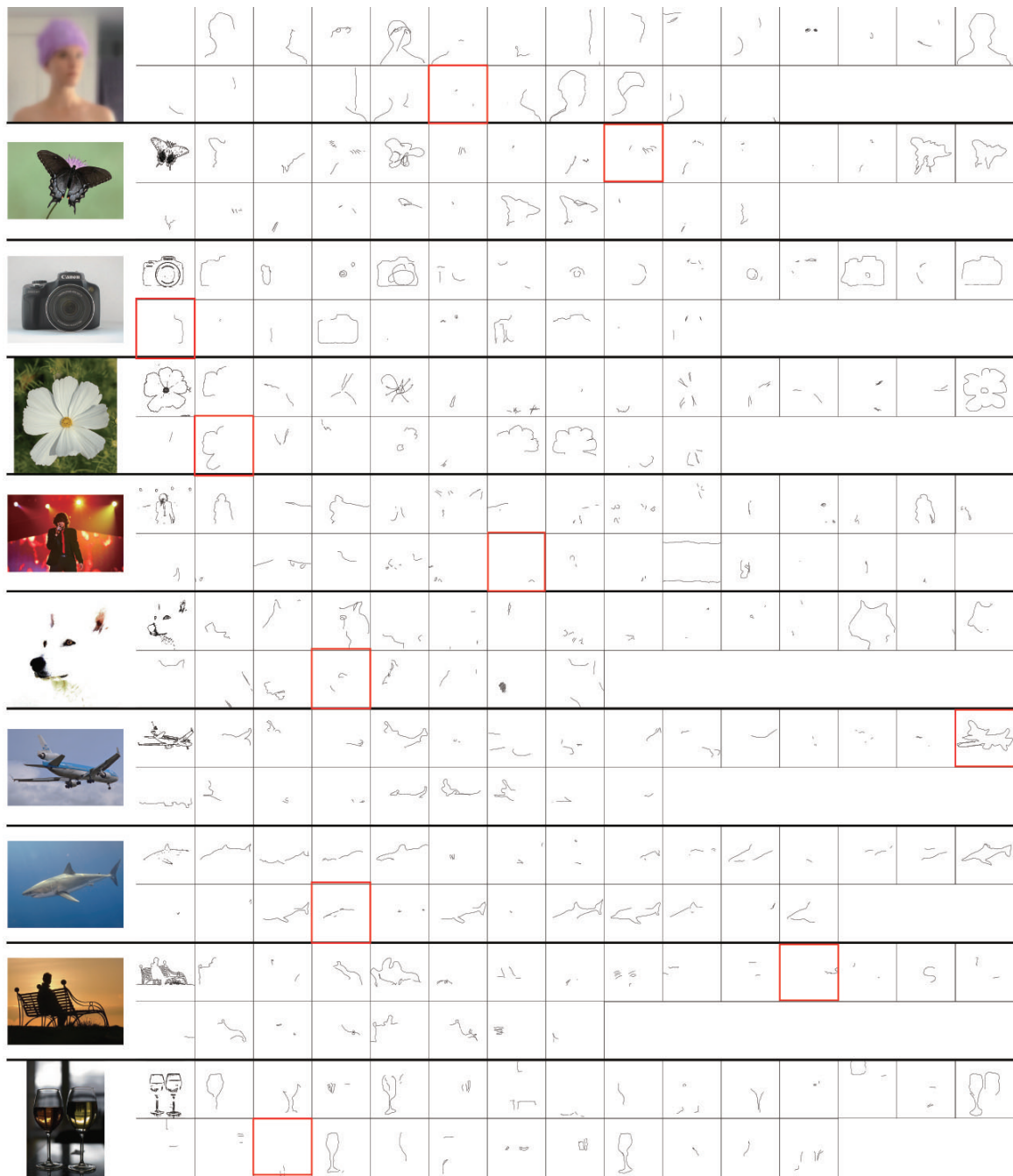


図 5.2: 各ワーカの描画内容 (輪郭抽出あり条件): 赤枠は評価者によって完成と判定された画像を示す



図 5.3: 線画イラストの生成過程 (輪郭抽出なし条件): 赤枠は評価者によって完成と判定された画像を示す

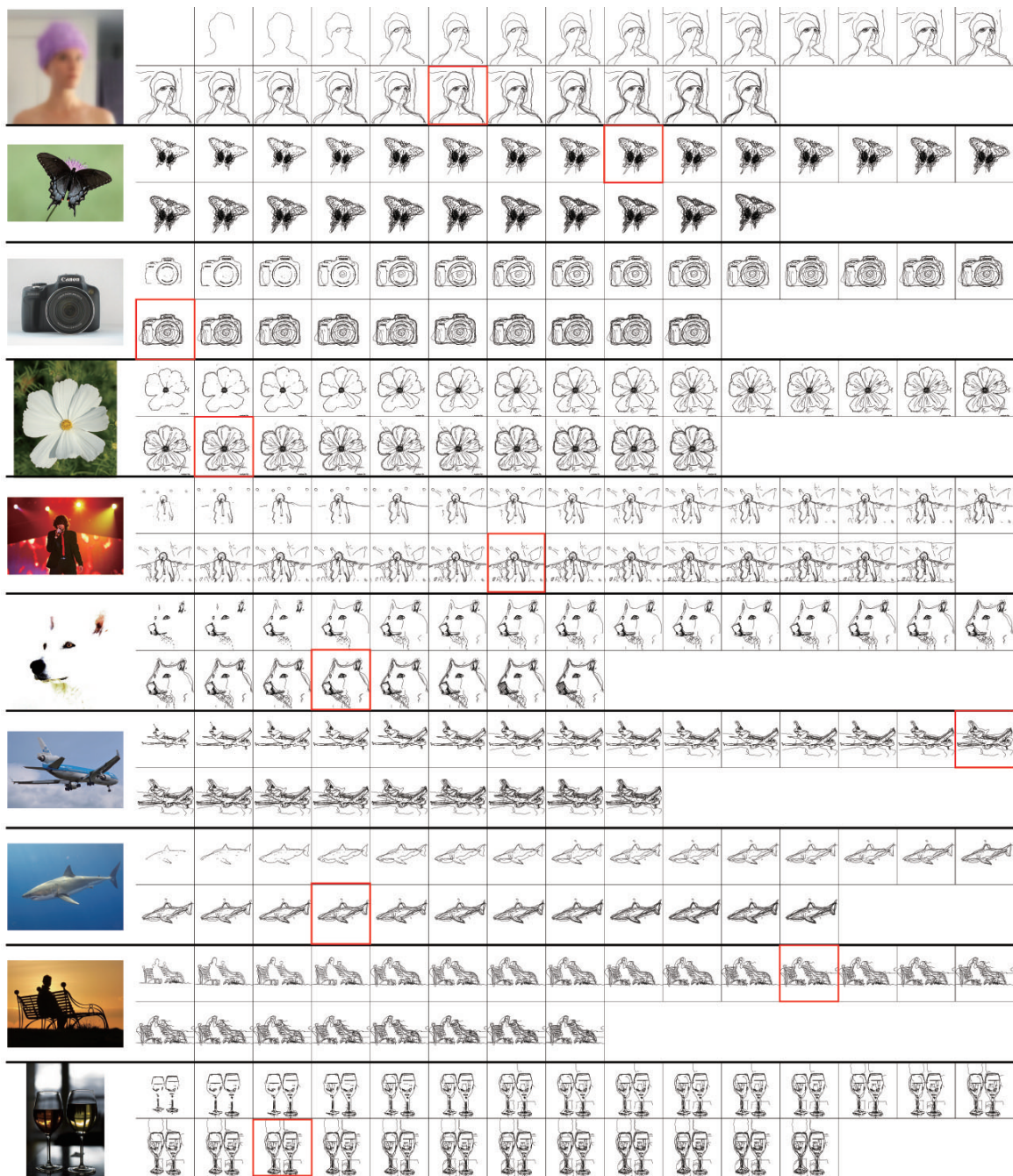


図 5.4: 線画イラストの生成過程 (輪郭抽出あり条件): 赤枠は評価者によって完成と判定された画像を示す

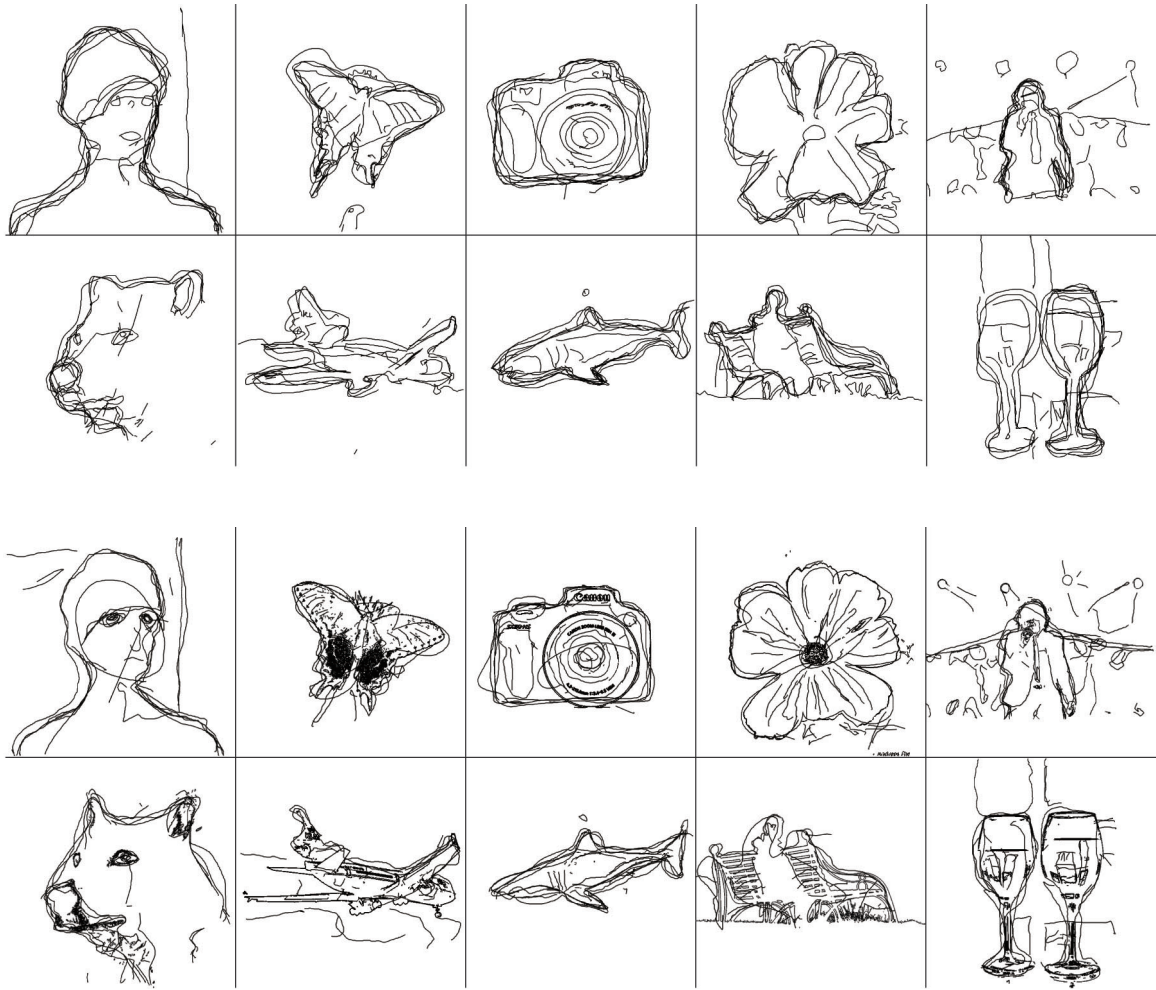


図 5.5: 完成と判定された線画イラスト (上：輪郭抽出なし条件, 下：輪郭抽出あり条件)

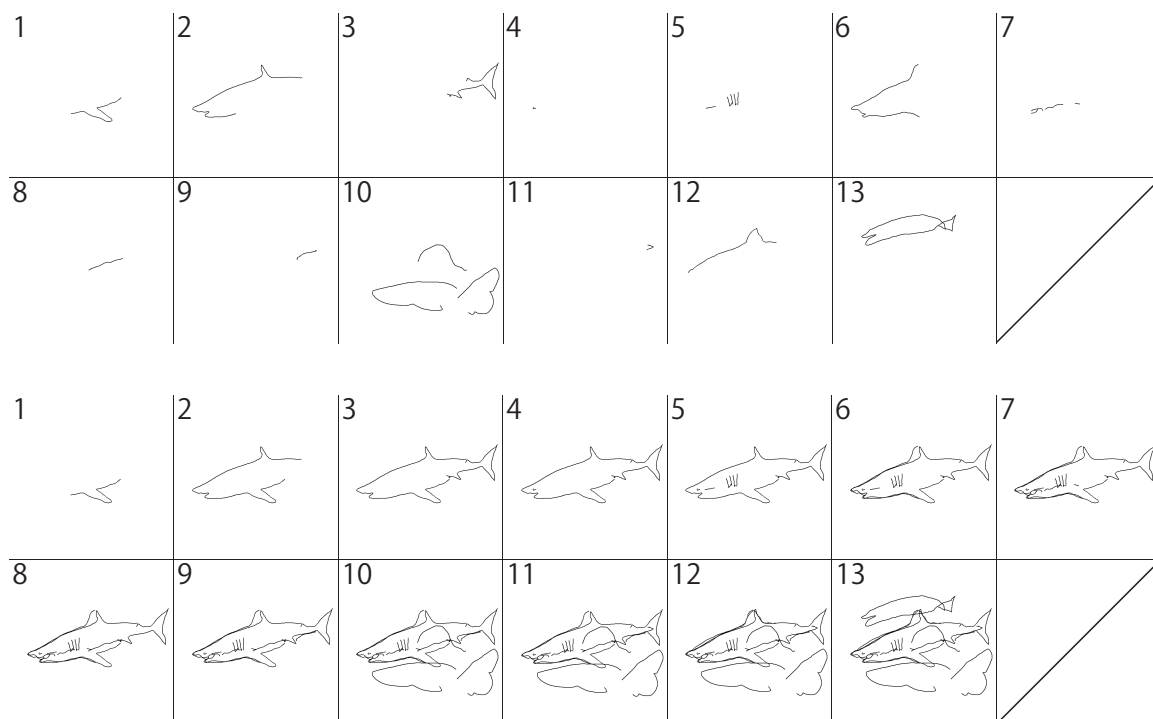


図 5.6: 複数重ね条件（第 1 グループ）での描画の様子：上段が各ワークの描画内容，下段が線画イラストの作成過程

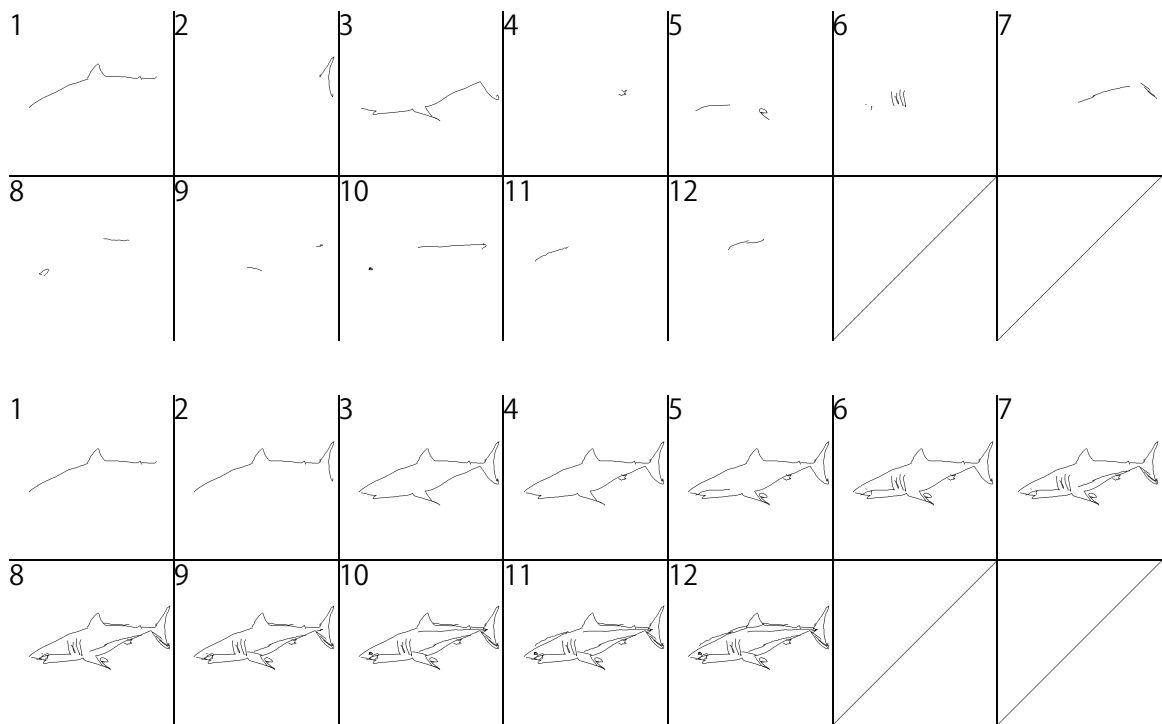


図 5.7: 複数重ね条件（第 2 グループ）での描画の様子：上段が各ワークの描画内容，下段が線画イラストの作成過程

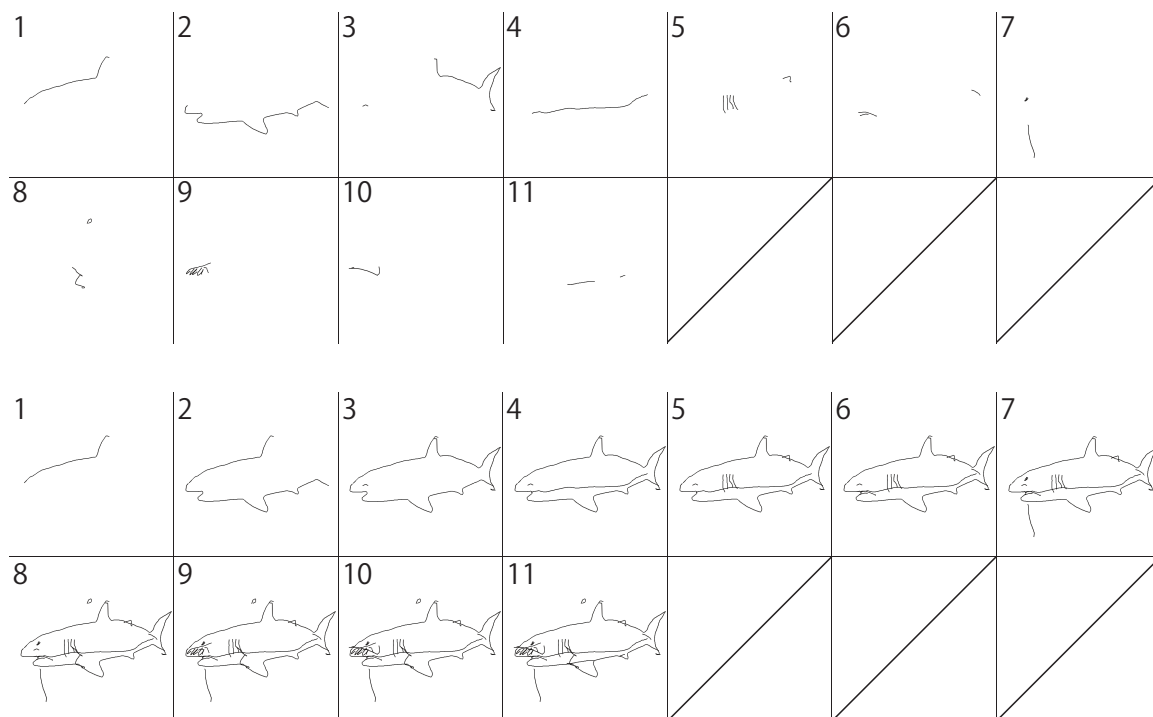


図 5.8: 複数隣置き条件 (第 1 グループ) での描画の様子: 上段が各ワークの描画内容, 下段が線画イラストの作成過程

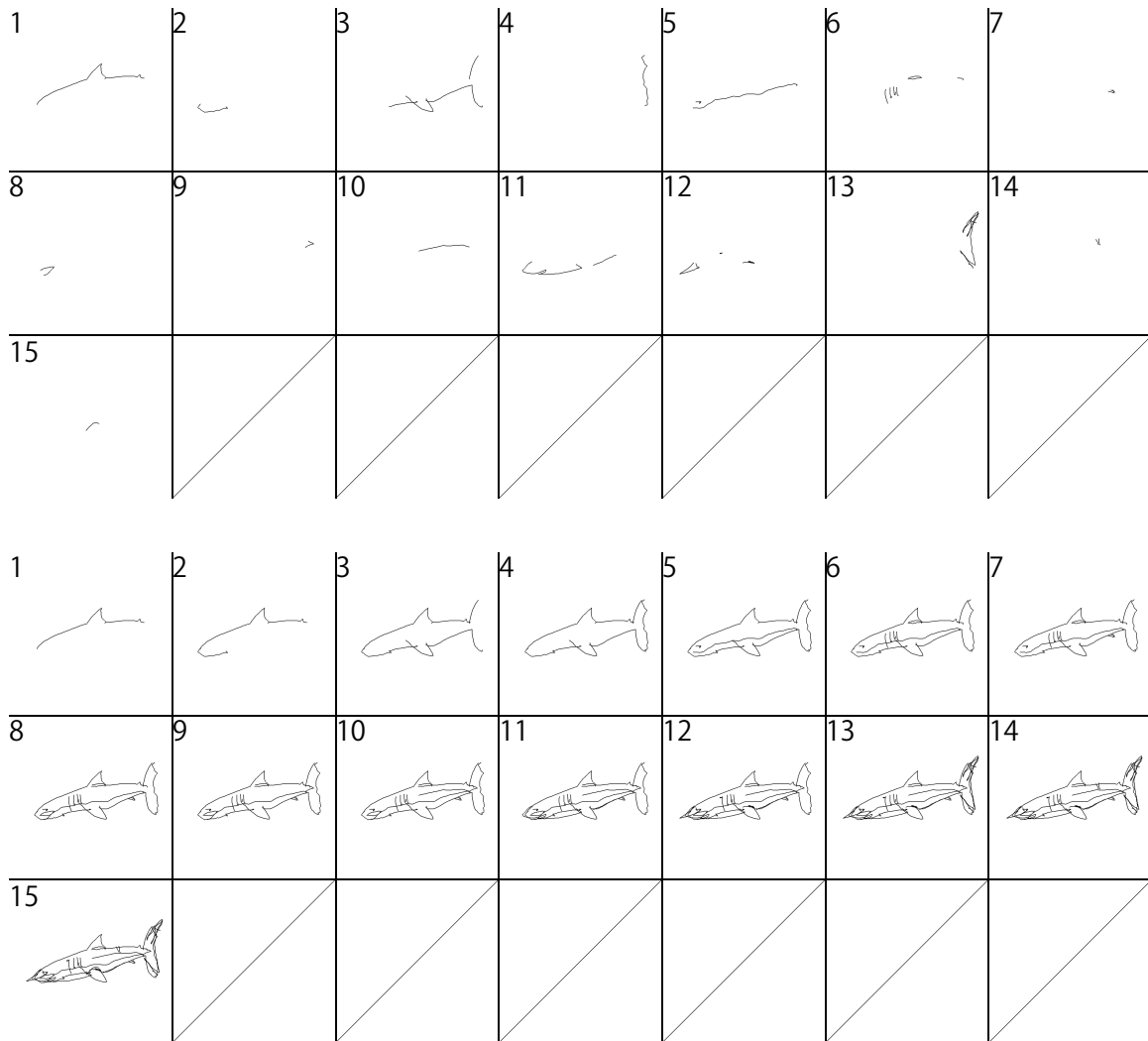


図 5.9: 複数隣置き条件 (第 2 グループ) での描画の様子: 上段が各ワーカの描画内容, 下段が線画イラストの作成過程

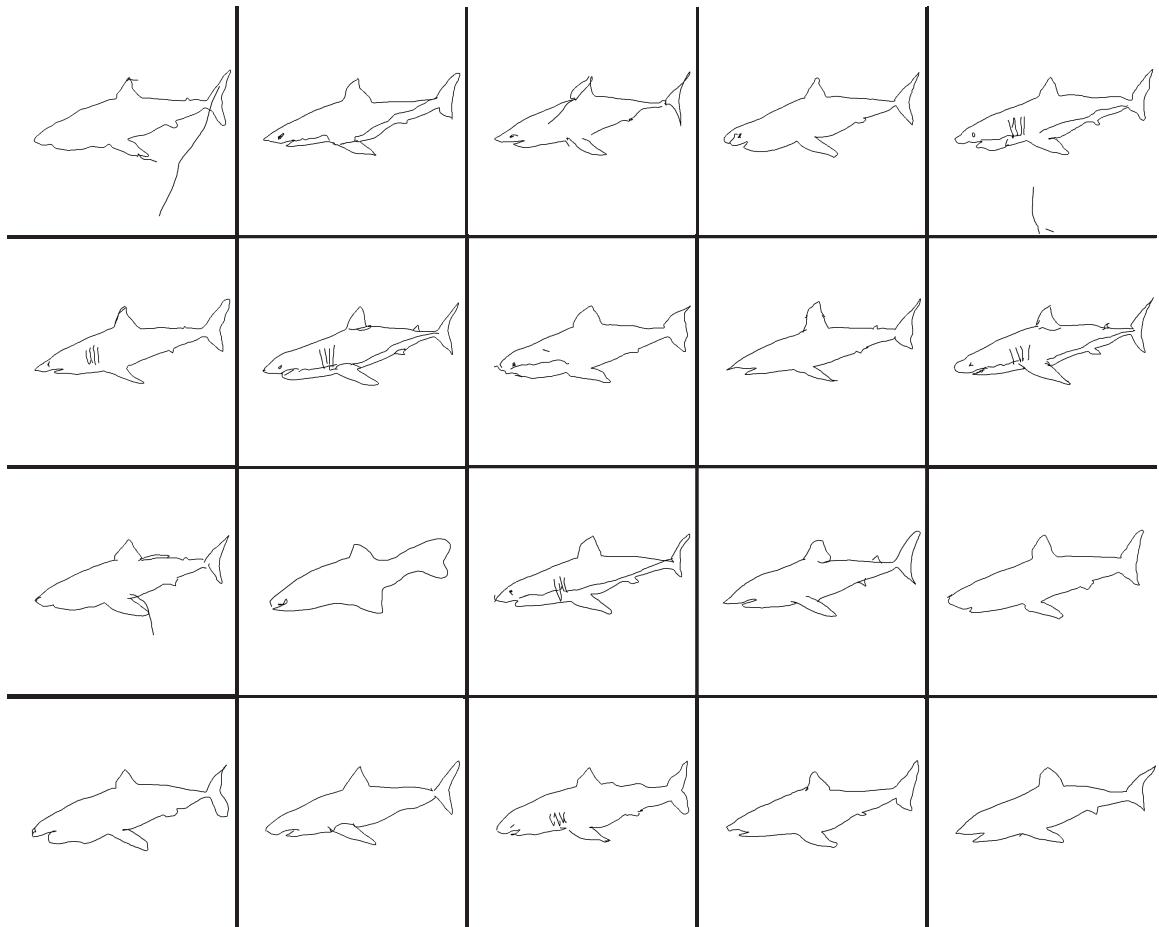


図 5.10: 単独重ね条件 (第 1 グループ) で作成された線画イラスト

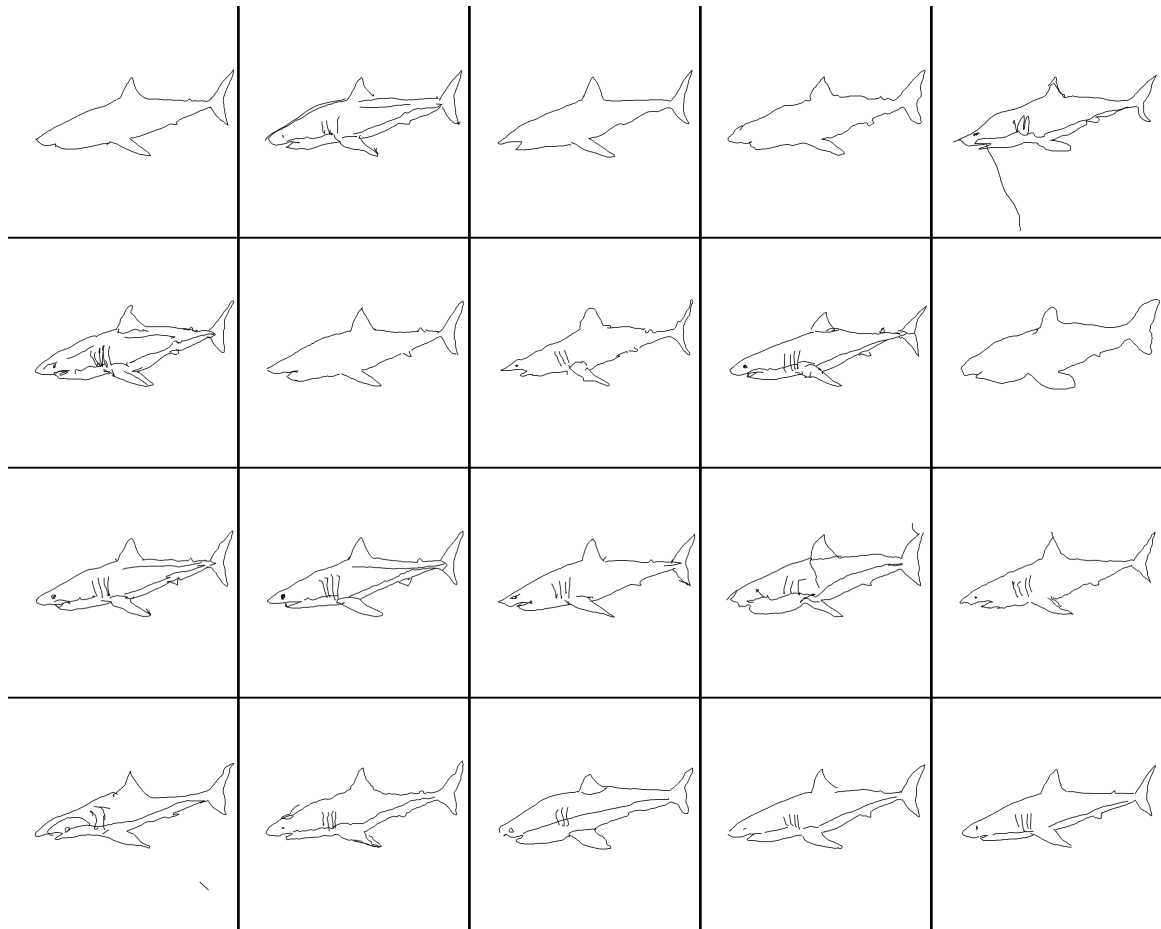


図 5.11: 単独重ね条件 (第 2 グループ) で作成された線画イラスト

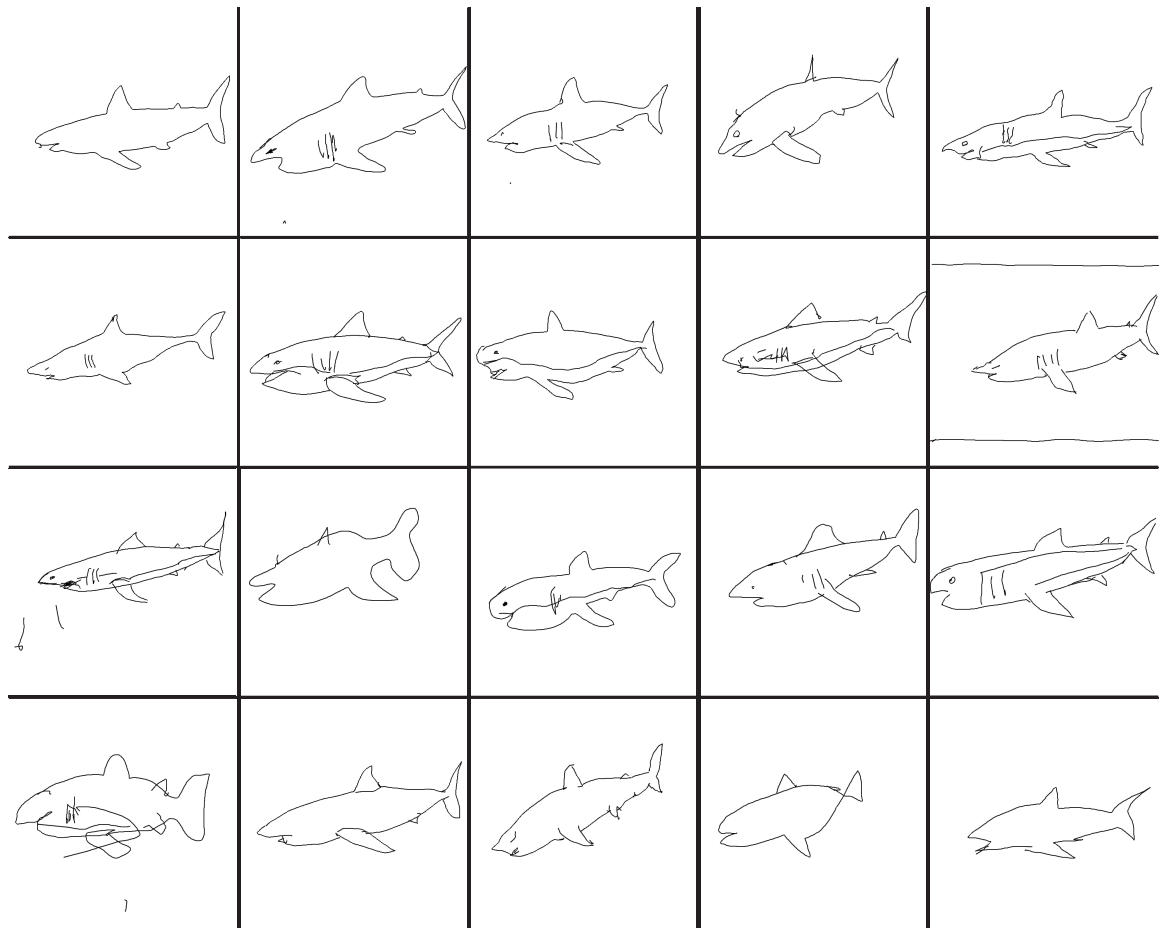


図 5.12: 単独隣置き (第 1 グループ) で作成された線画イラスト

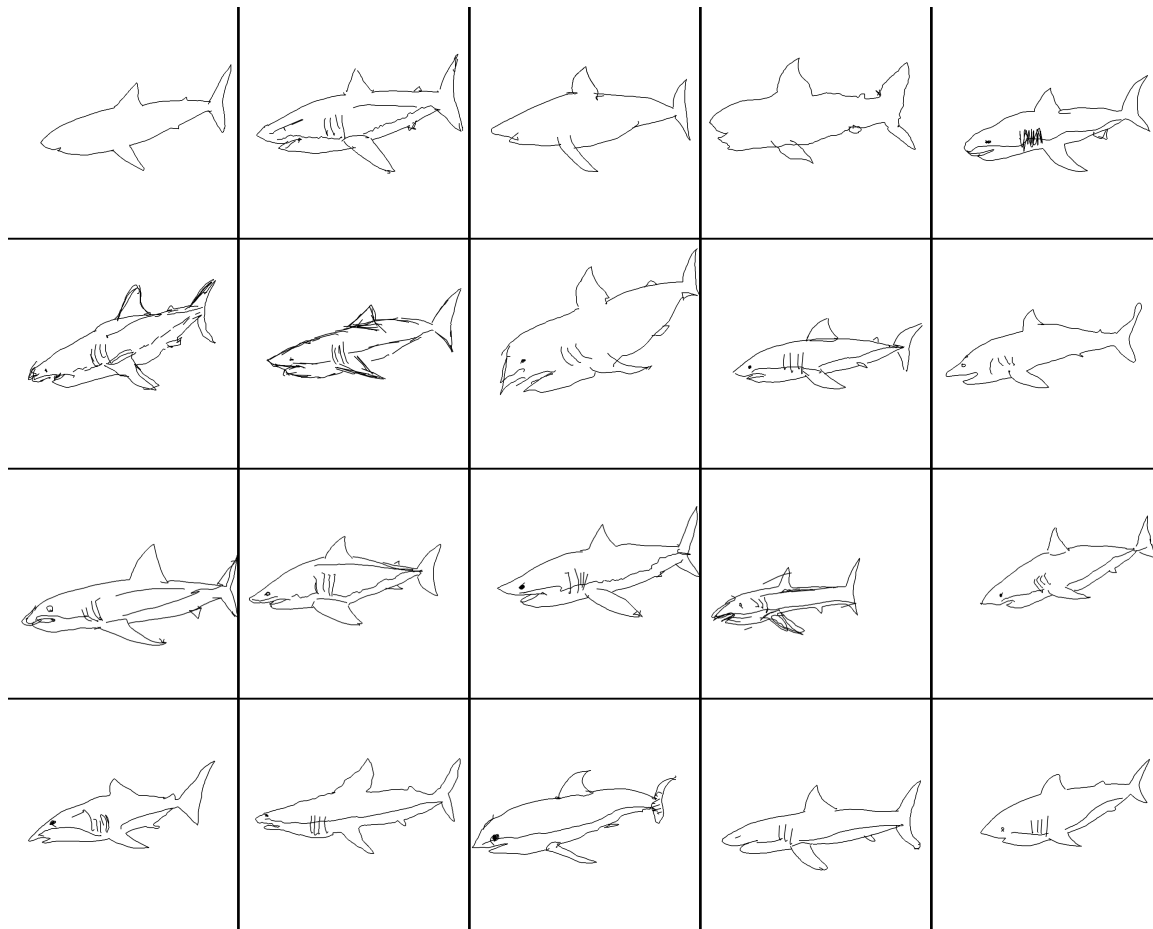


図 5.13: 単独隣置き (第 2 グループ) で作成された線画イラスト


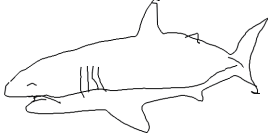


GROUP 1		GROUP 2	
overlaid	Adjoined	overlaid	Adjoined
			

図 5.14: 複数条件での完成画像：左側が第 1 グループ，右側が第 2 グループで，左列が原図重ね置き条件，右列が原図隣置き条件での完成画像

第6章 考察[8, 9]

6.1 タスク補助機能の有効性

タスク補助機能の有効性について検討するために、コンピュータによる輪郭抽出の有無が、線画イラスト生成にどのように影響を及ぼしたかを検討する。まず5.1.2節から、輪郭抽出なし条件と輪郭抽出あり条件で、完成までに要した描画の平均回数に差が見られた。また、10枚の画像中7枚について、完成までに要した描画の回数が、輪郭抽出あり条件の方が少なかった。この結果から、タスクの補助機能は、完成までに必要な描画の回数を削減できる可能性があることが示唆される。ただし、線画イラストを描くのが上手なワーカが一方の条件に偏っている可能性を考慮すると、試行数を増やすなど、より詳細な検討が望ましい。

また図5.5を見ると、輪郭抽出がある場合の方が、ワーカによって描かれた線のうち揺らいだ線が少ないように見える。このことは、輪郭抽出によって提示された原図に対してより忠実な線画を描くことができる可能性を示している。このように原図を模写するとき、描画線をより原図に近づける手法は他にも提案されている[23, 24]。逆に抽出画像がない場合、原図に対して生成される線画の忠実さが低下するが、その分生成される線画のバリエーションを増加させることが期待できる。今後の課題としては、原図をなぞるようなインタフェースだけではなく、描画領域の隣に原図を表示するインタフェースではどのような線画が生成されるかを調査することを考えている。

6.2 複数条件と単独条件の比較

図5.10, 5.11, 5.12, 5.13に見られるように、特性調査実験では実験参加者に時間制限がないことを伝え、参加者が完成と判断するまで線画を描画するよう伝えたにも関わらず、単独条件において輪郭線だけを描き完成とした参加者がいた。一方で、内部まで詳細な線を描く参加者もいた。複数条件では、単独条件で輪郭しか描かなかった人と内部の線まで描いた人の個人差が吸収された線画イラストが完成し、単独条件による線画イラストよりも完成度が高いことが示唆された。

なお、ワーカ集団が異なる場合など、状況によって生成される線画イラストには違いが生じ得るため、詳細な検討にはさらなる試行が望ましい。

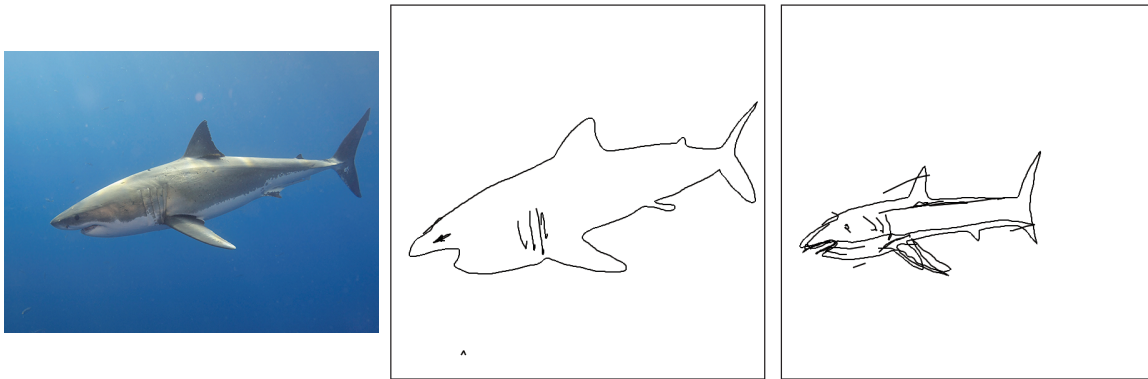


図 6.1: 多様性の例。左：原図，中央：ヒレの形が原図より丸い例，右：体が原図より細い例

6.3 原図の提示位置に関する比較

5.2.4 節では，原図の提示位置による違いを比較した。その結果，原図が描画領域の隣に提示された場合，描画領域に重ねて提示する場合に比べ，原図とは少しはなれた線画イラストが生成されることが分かった。原図とはなれるというのは，例えば図 6.1 に示されるように，サメの線画イラストを生成する場合において，原図よりもヒレの形が丸みを帯びたり，少し細身の身体が描かれたり，といったような違いである。つまり本手法を用いることで，同じ一枚の原図を用いたとしても，全く同じような物体の線画イラストが生成されるだけでなく，描画される線に多様性のある，複数の線画イラストが生成される可能性が示された。この原図提示方法により，線画イラストに対するクライアントの多様な要求に対応できる可能性がある。

また，特性調査実験後に参加者から聴取したコメントの中には，「原図が重ねてある状態で書くよりも，隣に置いてある方が描くのが難しい」という旨のコメントがあったが，今後の検討課題である。

6.4 タスクの所要時間

4.2.5 節で述べたように，単独条件では，線画イラストを完成させるまでに要した筆数を記録した。各グループ，原図の提示位置によらず全ての描画結果の筆数から，1 枚の線画イラストを作成するにあたり平均 12.4 筆（最小 1 筆，最大 85 筆，標準偏差 11.8 筆）要することが分かった。また 5.2.5 節の結果より，単独条件では 1 筆あたり 5.3 秒の描画時間がかかる。この結果より単独条件では，1 枚の線画イラストを作成する場合，およそ 65.4 秒かかると分かる。

一方でマイクロタスクを用いた本手法の場合，5.2.2 節の結果より，線画イラストが完成するのは平均して 9.0 回の描画後となる。そして本手法では，1 タスクあたり 7 秒の描画時間がかかる。この結果より複数条件では，1 枚の線画イラストを生成する場合，およそ 63.0 秒かかると分かる。

以上の結果より、従来通り1人で線画イラストを作成する場合と、マイクロタスクを用いて複数人で線画イラストを生成する場合、1枚の線画イラストが完成するまでにかかる時間はどちらもほぼ同程度になると考えられる。

6.5 初期の描画が及ぼす影響

図5.6, 5.7, 5.8, 5.9を見ると、複数条件では原図の提示位置、およびグループによらず、いずれも3番目のワーカまでに輪郭部の描画が終わっていることが確認できる。それぞれ3番目の描画ででき上がった輪郭線は大きく修正されることなく、いずれも最後まで描画が進んでいる。例えば、図5.8, 5.9では、第1グループはやや太めの、第2グループはやや細めのサメを、それぞれ3番目のワーカまでが描き、その後その概形を崩すことなく描画を進めていった。このことから、線画作成の序盤の方ででき上がった輪郭線は、線画の概形をほぼ決定する可能性が窺える。

6.6 ペンタブレットについて

線画イラストの生成実験では、ワーカの入力インターフェースとしてペンタブレットを使用した。これは4.1.3節で述べたように、専用のペンを持ってタブレットに描くという行為が、実際に紙とペンを用いて絵を描く行為に似ており、より直感的にタスクを行うことができるかと判断したためである。しかし実際は、両条件合わせて56名の実験参加者のうち41名がこの実験までにペンタブレットを触った経験がなかった。また本研究で提案する手法においては、絵を描くスキルが無いワーカでもタスクを行えるようにするのが一つの狙いであり、そのようなワーカがペンタブレットを保有しているとは限らない。しかし今回の実験では、タスク開始前に十分な練習時間を設けたため、ペンタブレットの操作不慣れによる、実験結果への影響はなかったと考える。

6.7 マイクロタスクの時間制限の効果

本研究では、線画イラストを生成するためのマイクロタスクは、1人あたり7秒が最適な時間と判断し、実験では1タスクあたり7秒の時間制限を設けて線の描画を行った。結果として、1名の実験参加者を除き、7秒という時間制限に何かしらの不満を覚えた参加者はいなかった。このため、マイクロタスクの時間設定は適切であったと考える。なお、7秒という短い制限時間を設けることで、線画イラストの作成という難しいタスクが行いやすくなる効果を期待していたが、これに関するコメントを実験参加者から得ることはできなかった。

6.8 タスクの実行結果の質と不真面目なクラウドワーカへの対応

タスクの実行結果の質について Kittur らおよび Sampath 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。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 [25]、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するよう工夫することで、それぞれタスクの質が向上しうることを明らかにした [26]。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を公開し、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 [27]。

特に本研究が提案する手法では、ワーカの中に 1 人でも不真面目なワーカが存在すると、生成される線画の質が大きく低下してしまうおそれがある。しかしこれらの知見から、そのような不真面目なワーカにも対応することができると考えられる。例えば、タスクの前後に Kittur らの示唆に則って、手本となる簡単な線をなぞるようにして描画する予備タスクを実施する。このとき、その線があまりにも手本の線から逸脱していたり、描画速度が速すぎたりした場合は、実際の線画イラスト生成タスクにおいてもそのような線を描く可能性が高くなるため、そのようなワーカが描いた線は保存しないようにする手法をとることができる。この場合、不真面目なワーカの他、真面目にタスクに取り組もうとしているワーカに対しても、そのワーカのタスクの実行環境等により正確な線を描けない場合は、描画結果が保存されないことになる。しかしこれは、生成される線画イラストの質を向上させるという点において有用である。

また Sampath らの知見から、線画として不足している部分をシステムが判断し、その場所をワーカに提示することで、ワーカが描くべき線の場所を容易に把握できるようにする方法も考えられる。適切な箇所に線を描画することで、線画イラストの質を高めることができると考えられる。Huang らの知見からは、個人を特定できない程度の情報を予めタスク開始前に登録してもらうことで、それまで線画の作成に携わってきたワーカの情報をシステムが表示し、作業の質を高める可能性がある。いずれも線画の質を向上させ、かつ同時に不真面目なワーカを廃することが可能になると考えられるが、これらの施策がどの程度作業に影響を及ぼすかは検討課題である。

6.9 実用イラスト生成に向けて

本論文の意義は、実用イラストの生成ではなく、クラウドソーシングによるイラスト生成のための基礎的手法を提案した点にある。そしてその提案手法によって、その質には改善の余地はあるものの実際に線画イラストが生成できることを示した点にある。

したがって、本実験で得られた線画イラストは多くの細かい線が重なったものが多く、一般に提供されている明瞭な輪郭線を持つ実用イラストやカラーイラストと異なる。本研究は、提案手法を核としつつ、このような実用イラストの生成を目標とするが、そのためには提案手法に加えて様々な工夫を追加、組み合わせることが必要であり、これらは今後進める予定である。具体的には、今後実用レベルのイラストを生成するために次の事項を検討している。イラストをカラー化するためには彩色タスクが必要である。この彩色タスクは提案手法と同様

にマイクロタスク化することで実現可能と考えている。現在の線画イラストでは細かい描線が数多く見られるため、これに画像処理を施すことにより、明瞭で主要な線だけによるイラストに変換することを考えている。なお一方で、絵コンテやラフ画、下描きなどに使われるような、必ずしも精密な線を必要としない線画イラストも存在し、これらについては今回生成されたような線画イラストをそのまま用いることも可能と思われる。

線画イラストの完成判定については、別のクラウドワーカーによってイラストの完成判定をするマイクロタスクを実行することが考えられる。

6.10 本手法を用いることによるコストの削減

1章で述べたように、本手法を用いることで、従来イラストを作成する方法に比べ、金銭的、または時間的、作業負荷的なコストを削減できると考える。雑誌等で用いられるカットイラストでは、イラストレータに作成を依頼する場合、モノクロのイラストでおよそ三千円から一万円の金額がかかると言われている [2]。一方で本手法のようにマイクロタスクをクラウドソーシングにより実行する場合、仮に Mturk でタスクを依頼する場合と同等の金銭コストがかかるとした場合、ワーカー1人に対して1時間6ドル程度報酬として支払うことになる [12, 13]。本手法におけるワーカーの拘束時間を1タスクあたり1分と仮定すると、金銭コストは1タスクあたりワーカー1人につき0.10ドルとなる。また、今後システムの拡張が進み、1枚の線画イラストを生成するために複数のマイクロタスクを挟むことになっても、ワーカー1人あたり支払う報酬額が0.10ドル程度ならば、金銭コストは本手法の方が、イラストレータに作成を依頼する場合に比べて低くなると見込まれる。一方で、時間的および作業負荷的なコストに関しても、マイクロタスクでは1回あたりの作業量は数十秒で済む。また、本手法では誰でも線画イラストの生成に参加できるため、作業負荷も、一般的に1人でイラストを作成する場合と比べると低いと考えられる。

しかし既存のイラスト作成手法では、イラストレータが時間をかけてイラストを作成し、高品質なイラストを提供する。それに対して現状、本手法で生成できる線画イラストは、実験結果から見られるように、線が歪であったり無駄な線があったりするなど、生成される線画イラストの質が異なる。この完成とされる線画イラストの質に差がある以上、一概に各種コストが低いから良い、と述べることはできない。現状では、本手法の特徴に合った線画イラストのニーズを見つけることで、従来手法よりコストが低い本手法を生かすことができると考えられる。

第7章 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、特に線画イラストを生成する手法を提案した。本手法では描画能力を問わない多数のワーカが少しずつ線を描画していくことで、1枚の写真から線画を生成する。手法の実現性を実験的に検討したところ、線画作成を時間によってマイクロタスク化しても、一般的な線画であれば生成可能であることが示された。特にコンピュータによる原図の輪郭抽出がない場合は、輪郭抽出がある場合に比べ、完成までに要する描画の回数が増える可能性が示された。また輪郭抽出がない場合、生成される線画の線に揺らぎが多く、生成される画像のバリエーションを増加させる可能性があることがわかった。

続いて、本手法によって生成される線画イラストの特徴と、より詳細な手法の性質を検討するための特性調査実験を実施した。まず、提案手法による線画イラストと1人による線画イラストを比較した結果、提案手法によって1人による場合と遜色のない線画を得られることが示唆され、提案手法の有効性が確認された。続いて、提案手法のように原図を描画領域に重ねて提示する場合と、原図を描画領域に重ねずに提示する場合を比較した結果、提案手法とは異なり、原図を描画領域に重ねずに提示することで、生成される線画イラストの多様性を生むことが見て取れた。さらに提案手法において、ワーカによって初期に描かれた線画は、大きく修正されることなく、線画の作成が進む様子が観察された。生成される線画イラストの質にはなお課題は残るが、これらの知見を生かすことで、クライアントの要求により合致した線画イラストの作成につながると見込まれる。

謝辞

本研究を進めるにあたり、指導教員の井上智雄先生には、研究テーマの決定から実験の進行、論文の執筆に至るまで多くの時間を割いて熟議していただき、研究活動に必要な多くの助言を頂きました。また研究活動において不明な点、疑問に感じた点についても親身になって相談に乗っていただき、私自身大きく成長することができました。心より御礼申し上げます。また森田ひろみ先生にはご多忙の中、研究副指導をお引受けいただきました。ここに感謝申し上げます。

加えて本研究の実験を行うにあたり、平田章氏及び何瓣氏に尽力賜りました。特に平田氏には実験に使用する画像素材の決定から実験の進行、論文に使用する画像の作成まで、様々な場面でお手伝い頂きました。お二方の力添えなしに本論文は完成できませんでした。深く御礼申し上げます。

最後になりましたが、実験にご協力いただきました参加者の皆様、およびこれまで様々な面で支えていただいた井上研究室の皆様に、感謝の意を述べさせていただきます。本当にありがとうございました。

参考文献

- [1] Diana Patterson: Technical writing: lines & spots, *SIGDOC Asterisk J. Comput. Doc.*, Vol.2, No.10, pp.8-10 (1976).
- [2] 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, <http://crowdworks.jp/public/jobs/category/27/articles/8754>.
- [3] 無料イラストなら「イラストAC」, <http://www.ac-illustr.com/>.
- [4] イラスト無料素材 イラストボックス, <http://www.illustr-box.jp/>.
- [5] 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, http://detail.chiebukuro.yahoo.co.jp/qa/question_detail/q111107922591.
- [6] イラスト作成のコンペなら1万で30案集まる【クラウドワークス】, <http://crowdworks.jp/public/jobs/category/27>
- [7] Kosuke Sasaki, Akira Hirata, Tomoo Inoue: Method of Generating a Drawing by Crowdsourced Microtasks. In *Proceedings of the 18th ACM Conference Companion on Computer Supported Cooperative Work & Social Computing*, pp.61-64 (2015).
- [8] 佐々木孝輔, 平田章, 井上智雄: マイクロタスクによる線画イラスト生成手法, 情報処理学会論文誌, Vol.57, No.1 (2016) (in press).
- [9] 佐々木孝輔, 平田章, 井上智雄: マイクロタスク型線画イラスト生成手法によるイラストの特徴, 情報処理学会論文誌デジタルコンテンツ, Vol.4, No.1 (2016) (in press).
- [10] Amazon Mechanical Turk, <https://www.mturk.com>.
- [11] SETI@home, <http://setiathome.ssl.berkeley.edu/>.
- [12] Niloufar Salehi, Lilly C. Irani, Michael S. Bernstein, Ali Alkhatib, Eva Ogbe, Kristy Milland, and Clickhappier, We Are Dynamo: Overcoming Stalling and Friction in Collective Action for Crowd Workers, In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pp.1621-1630 (2015).

- [13] Fair payment - WeAreDynamo Wiki, http://wiki.wearedynamo.org/index.php?title=Fair_payment
- [14] David Martin, Benjamin V. Hanrahan, Jacki O’Neill, Neha Gupta: Being a turker, *In Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, pp.224-235 (2014).
- [15] Rajan Vaish, Keith Wyngarden, Jingshu Chen, Brandon Cheung, Michael S. Bernstein: Twitch Crowdsourcing: Crowd Contribution in Short Bursts of Time, *In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp.3645-3654 (2014).
- [16] Mason Winter, Watts Duncan J.: Financial Incentives and the “Performance of Crowds”, *SIGKDD Explor. Newsl.*, Vol.11, No.2, pp.100-108 (2009).
- [17] Aniket Kittur, Boris Smus, Susheel Khamkar, Robert E. Kraut: CrowdForge: crowdsourcing complex work, *In Proceedings of the 24th annual ACM symposium on User interface software and technology*, pp.43-52 (2011).
- [18] Vamshi Ambati, Stephan Vogel, Jaime Carbonell: Collaborative workflow for crowdsourcing translation, *In Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, pp.1191-1194 (2012).
- [19] Michael S. Bernstein, Greg Little, Robert C. Miller, Björn Hartmann, Mark S. Ackerman, David R. Karger, David Crowell, Katrina Panovich: SoyLent: a word processor with a crowd inside, *In Proceedings of the 23rd annual ACM symposium on User interface software and technology*, pp.313-322 (2010).
- [20] Lixiu Yu, Jeffrey V. Nickerson: An internet-scale idea generation system, *ACM Trans. Interact. Intell. Syst.* Vol.3, No.1, pp.2:1-2:24 (2013).
- [21] Takashi Kawashima, Aaron Koblin: Ten Thousand Cents, *In ACM SIGGRAPH ASIA 2008 artgallery: emerging technologies*, pp.18-18 (2008).
- [22] Ten Thousand Cents,
<http://www.tenthousandcents.com/>.
- [23] Yong Jae Lee, C. Lawrence Zitnick, Michael F. Cohen.: ShadowDraw: real-time user guidance for freehand drawing, *In Proceedings of ACM SIGGRAPH 2011*, Vol.30, Issue 4, Article No.27, pp.27:1-27:10 (2011).
- [24] Alex Limpaecher, Nicolas Feltman, Adrien Treuille, Michael Cohen: Real-time drawing assistance through crowdsourcing, *In Proceedings of ACM SIGGRAPH 2013*, Vol.32, Issue 4, Article No.54, pp.54:1-54:10 (2013).

- [25] Aniket Kittur, Ed H. Chi, Bongwon Suh: Crowdsourcing User Studies With Mechanical Turk, *In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp.453-456 (2008).
- [26] Harini Alagarai Sampath, Rajeev Rajeshuni, Bipin Indurkha: Cognitively Inspired Task Design to Improve User Performance on Crowdsourcing Platforms, *In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp.3665-3674 (2014).
- [27] Shih-Wen Huang, Wai-Tat Fu: Don't hide in the crowd!: increasing social transparency between peer workers improves crowdsourcing outcomes, *In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp.621-630 (2013).