

Web 情報を用いた竜巻経路推定支援のためのクラウドソーシング技術開発の試み

丹治 寛佳^{1,a)} 森嶋 厚行^{1,b)} 井ノ口 宗成^{2,c)} 北川 博之^{1,d)}

受付日 2013年6月21日, 採録日 2013年10月8日

概要: 災害時における災害状況の迅速な把握は重要であるが, 必ずしも容易ではない. なぜなら, 災害状況把握の機械化は難しく, 災害状況把握を行うための人材の確保も困難であるためである. このような問題を解決する有効なアプローチとして, クラウドソーシングに大きな期待が寄せられている. 我々は, 災害発生時に災害の写真等が数多く Web 上に投稿されることに着目し, クラウドソーシングによってこれらの写真から災害状況を把握する手法の可能性を検討した. 特に, 2012 年 5 月 6 日に茨城県つくば市北条で発生した竜巻の経路推定を題材に, それをクラウドソーシングによって行うためのマイクロタスクを設計し, 実験を行った. 本論文の貢献は, 1 つ 1 つは単純な作業であるような複数のタスクを組み合わせて, より大きな問題を全体として解くという複雑なクラウドソーシングの事例を提供し, 本アプローチが災害状況分野において決して不可能ではないことを示すことである. 本研究を通じて, 災害の実データを利用した竜巻経路推定という複雑なクラウドソーシングの実現に関して, その可能性と注意点等に関するいくつかの知見が得られた. 本論文では, 開発した手法と実験結果, および得られた知見について報告する.

キーワード: クラウドソーシング, ヒューマンコンピューテーション, 災害情報, 竜巻

An Attempt to Develop Crowdsourcing Techniques for Identifying Tornado's Paths Based on Web Data

HIROYOSHI TANJI^{1,a)} ATSUYUKI MORISHIMA^{1,b)} MUNENARI INOBUCHI^{2,c)} HIROYUKI KITAGAWA^{1,d)}

Received: June 21, 2013, Accepted: October 8, 2013

Abstract: Quick understanding of disaster situations is important but not easy, because automation is not trivial and recruiting people for gathering and aggregating related information is difficult. Crowdsourcing is a promising approach for such situations. We examined a crowdsourcing approach by designing microtasks for inferring how a tornado moved based on the uploaded photos of the tornado. In particular, we conducted experiments using a set of photos that were taken in the event of the tornado occurred in Hojo, Tsukuba City, Ibaraki, Japan, on May 6, 2012. The contribution of this paper is to provide a real example of complex crowdsourcing in which many different types of microtasks are combined to achieve a big problem, and shows that the approach is possible in the area of understanding of disaster situations. In this research process, we learned the potential and important points of complex crowdsourcing in the domain. This paper reports the method, our experiments, and the lessons learned.

Keywords: crowdsourcing, human computation, disaster information, tornado

¹ 筑波大学
University of Tsukuba, Tsukuba, Ibaraki 305–8550, Japan
² 新潟大学災害・復興科学研究所
Research Institute for Natural Hazards and Disaster Recovery, Niigata University, Niigata 950–2181, Japan
a) hiroyoshi.tanji.2012b@mlab.info
b) mori@slis.tsukuba.ac.jp
c) inoguchi@gs.niigata-u.ac.jp
d) kitagawa@cs.tsukuba.ac.jp

1. はじめに

災害時における災害状況の迅速な把握は重要であるが, 必ずしも容易ではない [1], [2]. なぜなら, 災害状況把握の機械化は難しく, 災害状況把握を行うための人材の確保も困難であるためである. このような問題を解決する有効な

アプローチとして、クラウドソーシングに大きな期待が寄せられている。クラウドソーシングとは、インターネットを通じて不特定多数の人々（ワーカー）に仕事（タスク）を委託することである。2011年3月11日に起きた東日本大震災においても、Google Person Finder [3] や sinsai.info [4] 等の、クラウドソーシングを用いたサービスが数多く利用され、災害状況把握や被災地の復興に大いに役立った。

我々は、災害発生時に災害の写真等が数多く Web 上に投稿されることに着目し、クラウドソーシングによってこれらの写真から災害状況を把握する手法について検討を行った。特に、2012年5月6日に茨城県つくば市北条で発生した竜巻を題材に、竜巻が写った写真とクラウドソーシングを用いて、竜巻の経路推定を実際に試みた。

本研究では、マイクロタスク型のクラウドソーシングによって竜巻が写った写真から竜巻の経路を推定する手法を提案する。マイクロタスク型のクラウドソーシングでは、短時間で完了する比較的簡単なタスク（マイクロタスク）をワーカーに委託する。詳細は3章で説明するが、提案手法で用いるマイクロタスクは図2から図6に示すようなものである。

本論文の重要な貢献は、クラウドソーシングによる災害状況把握支援の可能性を示す実証例を提供することである。具体的には次のような貢献があると考えている。

複雑なクラウドソーシングの事例の提供。 既存のクラウドソーシングの多くは、個々のタスクの結果の単純な集合が最終的な解であり、タスク設計が容易である。たとえば、画像の文字認識タスクでは、各画像に書かれた文字をテキスト化した結果そのものが最終的な解であるため、単に各画像の文字をテキストとしてワーカーに入力させるタスクを用意すればよい。一方、竜巻が写った写真からの竜巻の経路推定といった問題の解決には、異なる作業を組み合わせた複雑なクラウドソーシングの枠組みを設計する必要がある。本論文では、複数のマイクロタスクの組合せを適切に設計することにより、複雑なクラウドソーシングを実現する事例を提供する。

実データを利用した知見の提供。 本論文では、災害の実データを利用した実験を行い、実際のクラウドソーシングにおける問題点とそれに対処するための知見を得ることができた。詳細は6.4節で説明するが、本論文の知見としては、次のようなものがある。(1) 災害の実データを利用したタスクでは、曖昧な判断が必要なケースが多い。たとえば、図4のタスクは、竜巻の写真が撮影された位置がどの地図要素に対応するかの判断を行うものであるが、各地図要素と個別に比較すると、どちらの地図要素も正しいように感じる場合がある。このような場合、人間は比較することによって、より正確な判断ができるため、比較のタスクが必要であるが、対象の比較をマイクロタスクで行うには、トーナメント方式が有効である。(2) 人間は必ず間違える。

単なるミスというだけではなく、本質的に答えるのが難しいタスクも存在する。たとえば、図5のタスクは、写真に写っている竜巻が地図要素中のどの位置に写っているかを特定するタスクであるが、写真だけからその位置を判定するのは一般に困難である。このような場合、タスクの結果がある程度間違っていることを前提とした仕組みの開発が必要である。

知見に基づく手法の提案。 本論文では、クラウドソーシングのタスクの結果は一定量必ず間違えているということを前提とし、そのうえで、可能性の高い最終的な解の候補を列挙するためのアルゴリズムを提案する。これにより、一定の間違いがあっても、列挙される候補の中に正しい候補が含まれることが保証される。このアプローチは本事例だけに適用可能なものではなく、津波や広域火災の被害範囲推定等、災害情報分野において広い範囲で有効であると考えられる。

本論文の構成は次のとおりである。2章で関連研究を説明する。3章では、竜巻の経路推定支援をクラウドソーシングで行う手法の概要の説明を行う。4章ではマイクロタスク設計の詳細について説明する。5章では、各タスクの結果に間違いが含まれていることを前提として、竜巻経路の候補を列挙する手法を提案する。6章では、実験について説明する。具体的には、2012年5月に茨城県つくば市北条で発生した竜巻の経路推定を行う実験を行った。実験の結果、設計したマイクロタスクが適切であり、また、間違っているものを含むタスクの結果に基づいて列挙した解（竜巻経路）の候補が、正しい解を含んでいることを確認した。6章では、さらに、本研究の結果から得られた知見についても議論する。

2. 関連研究・関連システム

本章では、次の3つについて説明を行う。(1) 災害状況把握に関する研究。(2) クラウドソーシングを用いた災害状況把握に関するシステム。(3) クラウドソーシング処理の効率化に関する研究。

(1) 災害状況把握に関する研究：この例として、新潟県中越沖地震での試み [5] や、緊急地図作成チームにおける効果的な現場型空間情報マッシュアップの実現に向けた研究 [2] がある。文献 [2] の研究では、東日本大震災において、国レベルでの効果的な意思決定を支援するために、災害状況等の地図での可視化による状況認識の統一を試みた。状況認識の統一とは、災害対応に従事する主要人物が、直面している主要課題を明らかにし、課題解決へ向けた方針の決定を行うため、収集・集約した最新の情報を共有することである。最新の情報を表す地図の作成は専門家チームが行うが、作成のために利用する情報としては、災害対応にあたる関係機関が収集した情報だけでなく、個人やグループが Web 上に公開した情報も利用した。個人が Web 上に公開

した情報を活用するという点では本研究も同じであるが、作業そのものはクラウドソーシングされず、専門家チームによって行われた。

(2) クラウドソーシングを用いた災害状況把握に関するシステム：この例として、東日本大震災の際に提供されたサービスである *sinsai.info* [4] がある。このシステムは、ニュージーランド地震の際に活用されたクラウドソーシングツールである、*Ushahidi* [6] を援用することで構築されている。まず、ワーカは東日本大震災に関するデータをメールや Twitter, *sinsai.info* のフォームからレポートとして投稿を行う。その後、投稿されたレポートの内容を「モデレータ」という特別なワーカが確認し、デマや扇動でないかと判断されるとそのレポートが公開される。公開されるレポートは、位置情報等をもとにエリアごとに分けられ、地図上に可視化される。このようにして、被災者等が災害状況の把握を行うことを可能にしている。このシステムで行っているクラウドソーシングは、役立つ情報の提供そのものをクラウドソーシングする単純なクラウドソーシングである。一方、本研究で行うクラウドソーシングは、個々のタスクは単純であるが、異なる種類のタスクの結果を組み合わせて最終的には竜巻の経路推定を行う複雑なクラウドソーシングの事例となっている。

sinsai.info [4] では Open Street Map を利用しており、アップロードされた GPS のログを被災地地図の作成に利用している。これは、参加型センシングの一例と考えられる。参加型センシングは、マイクロタスク型とは異なるクラウドソーシングの形態である。これらとともに組み合わせにより効果的な手法を開発することは興味深い今後の課題であるが、そのためには、参加型センシングのインセンティブを適切に設計する必要があると考えられる [7]。

(3) クラウドソーシング処理の効率化に関する研究：本研究は、クラウドソーシングの処理を人手による操作 (Human-powered operation) を含むデータベース演算としてモデル化し、議論している。これまで、このような処理に関する研究としては、SQL にクラウドソーシングによる人手の操作を埋め込む研究が進められてきた (*CrowdDB* [8], *Qurk* [9], *Deco* [10] 等)。これらの論文では、そのようなデータベース演算を効率化するための手法について提案されている。本論文で議論するような複雑なクラウドソーシングにも、それらの手法を応用することは可能であると考えられる。これらについては重要な今後の課題とする。

3. 提案手法概要

本章では、マイクロタスク型のクラウドソーシングにより、竜巻が写った写真から竜巻の経路を推定する手法の概要を説明する。まず、マイクロタスク型クラウドソーシングについて説明する。次に、提案手法が満たすべき要件を説明し、最後に手法の全体像について説明する。

3.1 マイクロタスク型クラウドソーシング

マイクロタスク型のクラウドソーシングでは、短時間で完了する比較的簡単なタスク (マイクロタスク) の作業をクラウドソーシングする。具体的には、タスクの依頼者がマイクロタスクを、クラウドソーシングプラットフォームが持つタスクプールに登録する。タスクプールに登録されているマイクロタスクは、そのプラットフォームにアクセスするワーカに割り当てられ、作業が行われる。マイクロタスク型のクラウドソーシングプラットフォームの例としては、*Amazon Mechanical Turk* [11] が有名である。また、学術分野においては、*Crowd4U* の開発 [12], [13] が行われている。

3.2 本提案手法が満たすべき要件

本提案手法は次の3つの要件を満たす必要がある。

要件1：タスクの作業時間 マイクロタスクの作業時間は、通常数秒から数分である [14]。したがって、各タスクの作業時間を平均数分以下とする。

要件2：災害エリアに関する知識の仮定 マイクロタスク型クラウドソーシングでは、多くの参加者がいたほうが迅速に問題を解決できる。それに加え、災害エリアの人々が参加できるとは限らない場合への対応を考えると、災害エリアに詳しくない人でも貢献できるようなタスクになるよう設計する必要がある。

要件3：結果の品質 クラウドソーシングでは、必ずしも個々のタスクの結果は正しいとは限らない。したがって、これらの結果に一定の間違いがあっても、最終的には精度の高い結果を計算する仕組みが必要である。

提案手法がこれら3つの要件を満たしているかについては、6.4節で議論を行う。

3.3 設計概要

マイクロタスク型クラウドソーシングによって、竜巻の経路推定を行うためには、この問題を複数のタスクの組合せに分解する必要がある。我々は、本タスク設計に先立っていくつかの予備実験を行い、その結果を参考に、竜巻経路推定支援を、複数のマイクロタスクの組合せと、竜巻経路候補列挙アルゴリズムにより行うことにした (図1)。

提案手法の入出力は次のとおりである。

入力：(1) 竜巻が写った写真の集合 $I = \{i_1, i_2, \dots, i_m\}$. (2) 時刻部分関数 $t : I \rightarrow Time$. (3) 地図 (たとえば Google Maps) をメッシュ状に分割した各部分の集合 $M = \{m_1, m_2, \dots, m_n\}$. 以下では M を分割マップと呼び、各 m_l を地図要素と呼ぶ。

出力：(1) 竜巻経路の候補の集合 $P = \{p_1, p_2, \dots, p_q\}$. (2) 各経路の「もっともらしさ」を表現するスコアを表す関数 $S : P \rightarrow [-\infty, \infty]$.

提案手法の処理は2段階に分けられる。第1フェイズ (各

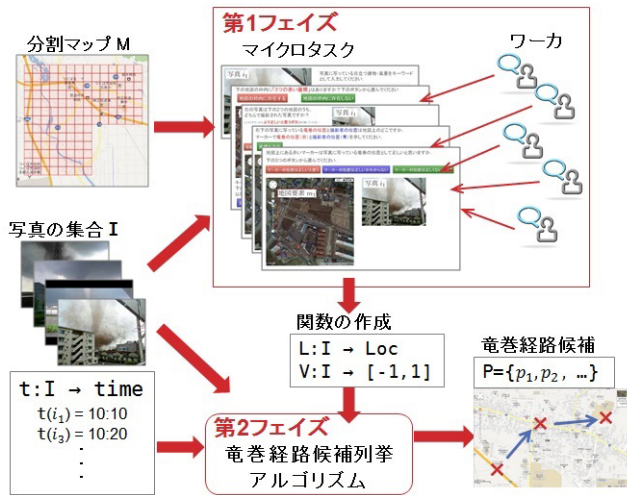


図 1 設計概要
Fig. 1 Design overview.

写真中の竜巻位置の同定)では、5種類のタスクを用いて、各 $i_j \in I$ に写っている竜巻が M 中のどの地図要素のどの座標に存在しているかの推測結果を表す関数 $L: I \rightarrow Loc$ を求める。ここで、 Loc は位置の集合であり、 $(m_l, x, y) \in Loc$ は、地図要素 m_l の座標 (x, y) の位置を表す。これに加えて、第1フェイズではまた、それぞれの推測の賛否度 (皆がその推測に賛成するか否か) を表す関数 $V: I \rightarrow [-1, 1]$ を求める。第2フェイズ (竜巻経路候補列挙) では、 I, t, L, V を用いて、竜巻経路の候補の集合 P と各経路の「もっともらしさ」を表現するスコアを表す関数 $S: P \rightarrow [-\infty, \infty]$ を作成する。

第1フェイズ：写真中の竜巻の位置同定。図2、図3、図4、図5、図6に示すような5種類のマイクロタスクを用いて、写真に写った竜巻の位置を特定する。Task 1~4の入力結果から関数 L を作成し、Task 5の入力結果から関数 V を作成する。

ワーカは、Task 1, Task 2, Task 3, Task 4, Task 5の順番でタスクを行う。まず、Task 1では、各写真 $i_j \in I$ に写っている竜巻以外の目立つ建物や風景をキーワードとしてワーカが写真に付与する。次に、Task 2では、各地図要素 $m_l \in M$ に対して、Task 1で入力されたキーワードの中から m_l に存在するものを付与する。Task 3では、各写真 i_j と同じキーワードが付与された地図要素 (一般に複数存在する) の中から、 i_j の撮影位置として最も適切な地図要素1つを、トーナメント方式で選択する。Task 4では、Task 3の結果から得られた、写真と (撮影位置の) 地図要素の組、および分割マップ全体が提示され、ワーカは写真に写っている竜巻の位置を分割マップに記入する。その結果から、関数 L を作成する。Task 5では、Task 4の各結果に対して、他のワーカが「正しいと思うか否か」を入力し、その結果から、関数 V を作成する。

GPS等によりすでに位置情報が含まれている写真に関



図 2 Task 1. 写真へのキーワード入力
Fig. 2 Example of Task 1.



図 3 Task 2. 地図要素とキーワードの結合
Fig. 3 Example of Task 2.



図 4 Task 3. 写真と地図要素の結合
Fig. 4 Example of Task 3.

しては、第1フェイズのTask 1からTask 3までを省略可能である。しかし、6章で説明する実験で使用した写真にはすべて位置情報が含まれていなかったため、現実には位置情報が含まれていない写真も多いと推察される。

第2フェイズ：竜巻経路候補の列挙。第1フェイズで得られた L と V 、および入力 I と t から、竜巻経路の候補の集合 P を作成する。ここでは、第1フェイズの入力が必ずしも正しくないという状況の下で、竜巻の経路推定を行う必要がある。



図 5 Task 4. 竜巻位置の入力
Fig. 5 Example of Task 4.



図 6 Task 5. 竜巻位置の賛否値の入力
Fig. 6 Example of Task 5.

4. マイクロタスク設計の詳細

4.1 モデル化

本節では、各タスク設計について、リレーショナルデータモデルを用いて詳細に説明する。以下では、下記のリレーションを利用する。

- 写真を格納するリレーション Img 。これは、スキーマ $Img(img)$ を持ち、インスタンスは $\{(i_j)|i_j \in I\}$ である。
- 分割マップを格納するリレーション Map 。これは、スキーマ $Map(map)$ を持ち、インスタンスは $\{(m_l)|m_l \in M\}$ である。

以下の具体例の説明は、図 2～図 6 を用いて行う。これらの図中には、写真 i_1 および地図要素 m_1, m_2 が写っている。 $I = \{i_1\}$ とし、 $M = \{m_1, m_2\}$ とする。

4.1.1 Task 1: 写真へのキーワード入力

Task 1 は、 $Img(img)$ 中の各 img ごとに、その写真に写っている建物や風景を表す $keyword$ を入力するタスクである。結果は $IK(img, keyword)$ に格納する。

ワーカは、図 2 のような Web フォームを用いて Task 1 を行う。ただし、目立つ建物や風景等が写真に写ってい

ない場合は、「目立つ建物等は特にない」ボタンを押す。図 2 では、 $i_1 \in I$ に対して、ワーカは写真に写っている「マンション」「3つの赤い屋根」の2つをキーワードとして入力している。その結果、 IK には、タプル群 $\{(i_1, \text{マンション}), (i_1, \text{3つの赤い屋根})\}$ が挿入される。

ここで、Task 1 において入力されたすべてのキーワード集合を表すリレーション $K(keyword)$ のインスタンスを $K = \{(keyword)|IK(img, keyword)\}$ とする。すなわち、例においては $K = \{(\text{マンション}), (\text{3つの赤い屋根})\}$ となる。

4.1.2 Task 2: 地図要素へのキーワードの付与

Task 2 では、各地図要素 $m_l \in M$ に、Task 1 で入力された $K(keyword)$ 中のキーワード w_k が表す建物や風景が存在するかを判定する。存在する場合には、リレーション $MK(map, keyword)$ にタプル (m_l, w_k) を挿入する。このタスクは次の式で表現できる。

$$MK = \pi_{map, keyword}(Map \bowtie_{map.contains(keyword)} K)$$

ここで、 $map.contains(keyword)$ は、「map 中に $keyword$ が表す物体が存在する」という結合条件であり、Task 2 によって人手で判定する。

ワーカは、図 3 のような Web フォームを用いて各 Task 2 を行う。図中の2つの画像は、同一の地図要素に対応する地図と航空写真である。

図 3 では、組 $(m_1, \text{3つの赤い屋根}) \in Map \times K$ に関する判定をワーカが行っている。「3つの赤い屋根」が表す建物・風景が分割 m_1 中に存在する場合は、ワーカは「地図の枠内に存在する」ボタンを押す、存在しない場合は、「地図の枠内に存在しない」ボタンを押す。図 3 では、「3つの赤い屋根」が m_1 中（地図の赤い枠線の中）に存在するため、ワーカは「地図の枠内に存在する」ボタンを押す。キーワードが不適切（意味不明な場合や明らかに地図に存在しない等）である場合には、ワーカは「見つけることは難しい」ボタンを押す。

4.1.3 Task 3: 写真と分割マップの結合

Task 3 では、Task 1 で生成された IK と Task 2 で生成された MK に関して、 $(i_j, w_k) \in IK \wedge (m_l, w_k) \in MK$ を満たす（すなわち、同じキーワードが付与された）写真と地図要素の組 (i_j, m_l) に対し、 i_j が m_l で撮影されたか否かを判定するタスクである。結果は、リレーション $J(img, map)$ に格納する。

Task 3 をすべて行うことにより、次の式を計算することになる。

$$J = \pi_{img, map} (IK \bowtie_{IK.keyword=MK.keyword \wedge takenAt(img, map)} MK)$$

ここで、 $takenAt(img, map)$ は「 img は map 中で撮影された」という結合条件であり、各 Task 3 で人手により判定

する。

各 Task 3 は次のように生成される。まず、ある写真 i_j と同じキーワードを持つ地図要素の集合を $MapCandidates_j = \{m_l | (i_j, m_l) \in IK \bowtie_{IK.keyword=MK.keyword} MK\}$ とする。次に、 $MapCandidate_j$ の地図要素を用いて、 i_j に対応する地図を選択するためのトーナメントを行う。したがって、各 i_j ごとに、 $|MapCandidates_j|$ 個の Task 3 が行われることになる。

ワーカは、図 4 のような Web フォームを用いて各 Task 3 を行う。各 Task 3 では、ある写真 i_j の撮影位置の候補として、 $MapCandidate_j$ 中の異なる 2 つの地図要素が表示され、撮影位置として正しいのはどちらかであることをワーカが判定する。図 4 では、左の地図要素 m_1 が正しいため、ワーカは「写真は左下の地図内で撮影された」ボタンを押す。どちらの地図要素も正しくないとワーカが判定する場合は、「どちらの地図内でも撮影されていない」ボタンを押す。このようなタスクを、各写真 i_j に対して 1 つ以下の地図要素が定まるまで繰り返す。

4.1.4 Task 4 : 竜巻位置の入力

Task 4 は、Task 3 で入手した写真と地図のペア $(i_j, m_l) \in J(img, map)$ ごとに生成される。各タスクでは、竜巻の写真 i_j と、撮影位置の地図要素 m_l 、および分割マップ全体 M がワーカに提示される。ワーカは、写真の竜巻がどの地図要素 m_r のどの座標 (x_a, y_b) に存在するかを推測して入力する。結果は、関数 $L : I \rightarrow Loc$ を表すリレーション $Location(img, map, x, y)$ に格納する。具体的には、 $L(i_j) = (m_r, x_a, y_b)$ であれば、タプル (i_j, m_r, x_a, y_b) を $Location$ に格納する。

ワーカは、図 5 のような Web フォームを用いて Task 4 を行う。例では、写真 i_1 が、地図中のどの座標に写っているかを、マーカーを任意の場所に動かすことにより入力する。複数のワーカにより同一のタスクを行う場合には、入力された複数の竜巻位置の重心を最終的な「その写真に写っている竜巻の位置」とする（すなわち、 $L(i_j)$ の値とする）。

4.1.5 Task 5 : 竜巻位置の賛否値の入力

Task 5 の目的は、賛否値を表す関数 $V : I \rightarrow [-1, 1]$ を作成することである。そのために、各 Task 5 は各 $(i_j, m_l, x_a, y_b) \in Location$ ごとに生成される。写真 i_j に写る竜巻が、地図要素 m_l の (x_a, y_b) に位置するという推測に対して、賛否を入力する。

ワーカは、図 6 のような Web フォームを用いて Task 5 を行う。画面には、写真 i_j と竜巻位置の推測結果が提示 (m_l 上の位置 (x_a, y_b) にマーカー表示) が提示される。ワーカはそれに対する賛否を入力する。この例では、写真に写った竜巻の位置は、地図上のマーカーの位置で正しいため、ワーカは「マーカーの位置は正しいと思う」ボタンを押す。写真に写った竜巻の位置が地図上のマーカーの位置ではない場合は、「マーカーの位置は正しくないと思う」ボタンを押

し、写真に写った竜巻の位置が地図上のマーカーの位置であるか分からない場合は、「マーカーの位置は正しいか分からない」ボタンを押す。賛否の結果 ($V(i_j)$ の値) は、前からそれぞれ 1, -1, 0 の値となる。

複数のワーカにより同一のタスクを行う場合には、 $V(i_p)$ の値は、全員の賛否の値の平均とする。

5. 竜巻経路の候補列挙

本フェイズでは、写真集合 I と時刻部分関数 t 、およびマイクロタスクの結果によって得られた 2 つの関数 $L : I \rightarrow Loc$ と $V : I \rightarrow [-1, 1]$ から、竜巻経路の候補の集合 $P = \{p_1, p_2, \dots, p_m\}$ および各経路の「もっともらしさ」を表現するスコアを表す関数 $S : P \rightarrow [-\infty, \infty]$ (値が大きい方がもっともらしい) を出力する。ここでのチャレンジは、次の問題に対応しなければならないことである。(1) 必ずしもマイクロタスクの結果は正しくない。(2) 考えられる経路候補の数は非常に多い ($\sum_{k=2}^{|location|} |location| C_k^k!$)。

このため、本論文では、(1) 少数の結果 (位置の推測) を用いた経路候補をまず作成し、(2) その後、貪欲法を用いてより詳細な (多くの結果を用いた) 経路候補を列挙するというアプローチを提案する。次に説明するアルゴリズムでは、(1) において 2 つの結果だけを用いた経路候補をまず作成する。したがって、少なくとも 2 つの結果が正しければ、おおざっぱであっても正しい竜巻経路候補を必ず含むことが保証される。

Algorithm 1 竜巻経路の候補列挙

```

1:  $N = \{n_j | n_j = L(i_j), i_j \in I\}$ ; // 竜巻位置の集合
2:  $seeds = \{[n_i, n_j] | n_i \in N, n_j \in N, n_i \neq n_j\}$ ; // 2 つのノードからなる竜巻経路候補の集合
3:  $P = seeds$ ; // P に結果を格納する
4: while  $seeds \neq \emptyset$  {
5:    $seeds' = \emptyset$ ; // ノード数が 1 多い経路候補を格納する変数
6:   foreach  $p \in seeds$  {
7:      $N' = N - p.nodes()$ ;
8:     if  $N' \neq \emptyset$  {
9:        $p' = ExtendPath(p, N')$ ; // 新しい候補  $p'$  を求める
10:       $seeds' = seeds' \cup \{p'\}$ ;
11:    }
12:  }
13:   $P = P \cup seeds'$ ;
14:   $seeds = seeds'$ ; // 次の候補計算のための seeds に更新
15: }
```

Algorithm 1 は本アプローチによる竜巻経路の候補作成アルゴリズムである。まず、1 行目で、Task 1~5 のマイクロタスクから得られた竜巻の位置を表すノードの集合を N に格納する。2 行目で、 N 中の任意の 2 つのノードのリスト $[n_i, n_j]$ によって表現されるパスを最初の経路候補とし、3

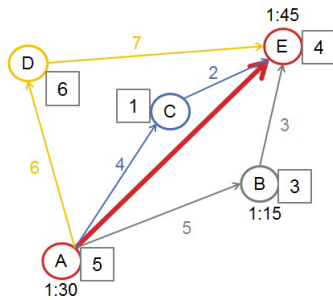


図 7 貪欲法によるパスの選択例

Fig. 7 Choosing a path by a greedy algorithm.

行目で、これらを結果 P に格納する。これが、上記の (1) に対応する。

次に、(2) を行う。 $seeds$ に含まれているすべてのパスに対して、それよりもノード数を 1 つ増やしたパスの集合を作成する。初回は、 $seeds$ は長さ 2 (ノード数が 2) の経路の集合であるため、長さ 3 の経路の集合を作成する。これらの経路は $seeds'$ に格納する。具体的には次のように行われる。まず、 $seeds$ に含まれてる各パス p に対して (6 行目)、 p に含まれない N 中のノードの集合を N' とする (7 行目)。9 行目の $ExtendPath(p, N')$ の詳細は次の段落で説明するが、簡単にいうと、 N' 中のいずれかのノードを p に追加して長さを 1 増やしたすべてのパスの中から、最も「もっともらしい」候補を計算する。新しい候補は $seeds'$ に追加する (10 行目)。 $seeds'$ の計算が終了したら、さらに 1 つ長いパスの計算のために、 $seeds$ を更新する (14 行目)。

$ExtendedPath(p, N')$ の具体的な処理は次のとおりである、まず、「 N' 中のいずれかのノードを p に追加して長さを 1 増やしたすべてのパス」とは、すべての $n' \in N$ に関して、パス $p = [n_1, n_2, \dots, n_q]$ の各 n_i のいずれかの間もしくは両端に n' を配置したものである。したがって、 $|N'| \times |p| + 1$ 個のパスが候補として列挙される。次に、ノードの写真に付与された撮影時間と矛盾している竜巻経路 (パス) を除去し、そのうえで、最も「もっともらしい」候補を選択する。これは、次の式で表すパス p' の評価値 $S(p')$ が最も大きくなるようなパスである。

$$S(p') = \frac{confidence(p)}{distance(p)}$$

ここで、 $confidence(p')$ は、 p' に含まれるノードの賛否値の和であり、 $distance(p')$ は、経路 p' が表現する地図上の長さ (ノード間の距離の和) である。たとえば、図 7 において $p = [A, E]$ のとき、ノード数が 1 多く、写真の撮影時間に矛盾せず、かつ評価値を最大にするパスは $p' = [A, C, E]$ であるため、そのパスが選ばれる。

6. 実験

本実験の目的は、実データを用いた竜巻経路推定の把握支援に対してクラウドソーシングの適用実験を行うことに

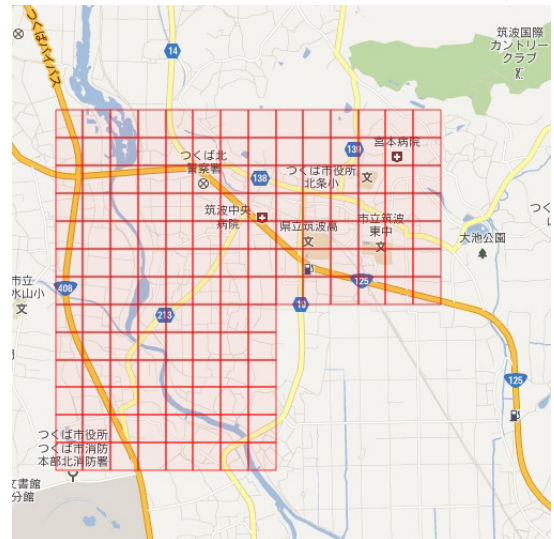


図 8 使用した分割マップ

Fig. 8 Map elements used in the experiment.

より、次にに関する知見を得ることである。(1) 現地に詳しくない大量の人によるマイクロタスク型のクラウドソーシングが、竜巻経路推定の支援という問題に実際に適用可能か否かを明らかにする。(2) 設計したマイクロタスクが、クラウドソーシングのタスクとして適切であるかどうかに関する知見を得る。(3) 本アプローチが適用可能であれば、注意すべき点に関する知見を得る。そうでなければ、なぜそうなのかを明らかにする。

実験の結論としては、(1) に関しては適用可能であり、実用化の可能性が高いことが分かった。(2) に関しては、マイクロタスクに求める要件を満たしていることが確認できた。(3) に関しては、竜巻経路推定に関するタスクの作業を完全に正しく行うことは期待できないこと、したがって、タスクがある程度間違っていることを前提として候補を列挙するアプローチが有効であることが分かった。

本章では、まず、実験について説明し、次にその結果と得られた知見について議論を行う。

6.1 実験方法

対象データ。本実験では、2012 年 5 月 6 日につくば市で発生した竜巻を対象とした。実験には、竜巻が写った写真 21 枚と、地図要素 160 個からなる分割マップを使用した。使用した写真は、Web 上に投稿された竜巻が写った写真の集合の中から、目立つ建物や風景が写っていない写真を取り除いた写真の集合である。使用したいずれの写真にも位置情報は含まれていなかった。使用した分割マップは、GoogleMaps [15] の地図と航空写真を利用し、特に竜巻被害の多かったつくば市北条地区を中心に、図 8 のように地図を 200 m 四方のメッシュ状に分割して作成したものである。利用する地図の範囲は、関係資料 [16] において竜巻の発生状況に示されている「山木・泉・小泉・北条地区」を

含む形に設定した。

利用プラットフォーム. 本実験では, Task 1~Task 4 は専用システムを構築し, Task 5 は Crowd4U を用いて実験を行った。

各タスクを行うワーカーの人数. ワーカーは, 全員が大学生および大学院生である。実験後の調査では, つくば市北条地区に関して詳しい人物はいなかった。同一のタスクを行う人数は, タスクの種類ごとに異なる。本実験の目的は, 竜巻の経路判定支援をクラウドソーシングで可能かどうかに関する検証実験であるため, 固定値を割り当てた。たとえば, 多くのキーワードを求めたいタスク (たとえば Task 1) や, タスク作業を間違えるとただちに品質に直接結びつくタスク (たとえば Task 4, Task 5) には複数の人数を割り当てた。具体的には次のとおりである。(Task 1) 各タスク 3 名。(Task 2) 各タスク 1 名。(Task 3) 「左」もしくは「右」が 2 票を得るまで, もしくは「どちらでもない」が 3 票を得るまで。(Task 4) 各タスク 3 名。(Task 5) 各タスク 10 名。人数の変更によるコスト/品質のトレードオフの追求に関しては本論文のスコop外ではあるが, 重要な点であるため, 今後の課題としたい。

6.2 各タスクの作業結果

タスクの種類ごとに結果を説明する。

Task 1. Task 1 では, 21 個のタスクが生成された。入力されたキーワードの数は 85 個であり, 1 タスク作業あたりの入力されたキーワード数は 1.35 個であった。

Task 2. Task 2 では, 4.1.2 項で説明したように, 不適切と報告されたキーワードに関するタスクは, 結合判定には使わない。今回は, そのように報告されたキーワードが 71 あったため, Task 2 のうち, 実際に結合の判定が行われたタスクは $14 \times 160 = 2240$ 個であった。したがって, Task 2 のタスクの総数は $2240 + 71 = 2311$ 個となる。

Task 3. Task 3 では, 1,277 個のタスクが生成された。Task 3 の結果として生成された, 写真と撮影場所を表す地図要素の組の数 ($J(img, map)$) の数は 7 つであった。そのうち, 撮影位置が地図要素に正しく含まれていたもの (正解) は 5 つ, 間違いは 2 つであった。

Task 4. Task 4 では, 7 つのタスクが生成され, それぞれ各 3 名のワーカーが行った。タスクの結果から得られた竜巻位置を図 9 に示す。赤いマーカは竜巻位置を示しており, 数字は竜巻が写っている写真の ID を示している。Task 3 における撮影場所の同定では正しい結果が 5/7 であったのに対し, Task 4 における正しい竜巻の位置 (撮影場所と異なることに注意) を示しているものは 26 番と 44 番のマーカの 2 個であり, 他の 5 個のマーカの示す竜巻位置は間違いであった。

Task 5. Task 5 では, 7 つのタスクが生成され, 10 人のワーカーがそれらを行った。タスクの結果から得られた竜巻

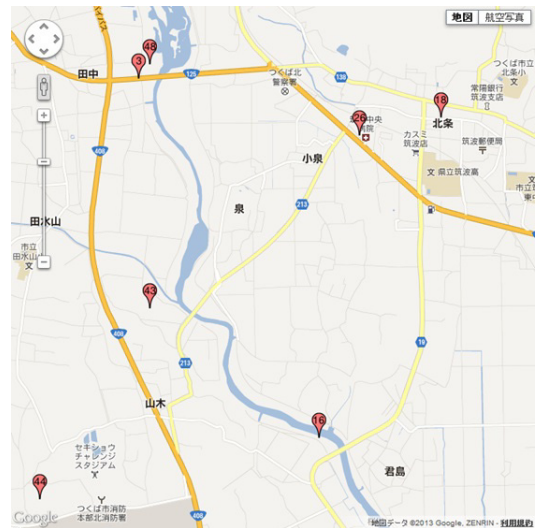


図 9 実験結果: 竜巻位置

Fig. 9 Result: Tornado locations.

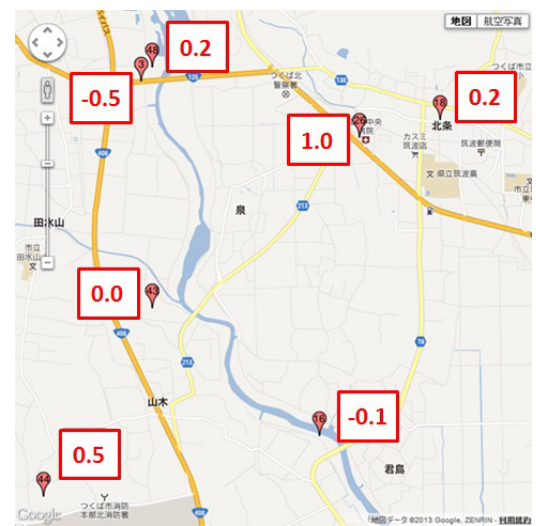


図 10 実験結果: 竜巻位置の賛否値

Fig. 10 Result: Confidence of tornado locations.

位置の賛否値を図 10 に示す。図 10 から分かるように, 正しい竜巻位置 2 個の賛成が多い (賛否値が高い) という結果が得られた。

6.3 竜巻経路推定の結果

マイクロタスクの結果から得られた 7 個の竜巻位置のうち, 対応する写真に撮影時間が付与されていたのは 4 個であった。5 章で説明した竜巻経路推定アルゴリズムは, 56 個の竜巻経路候補を列挙した。それらのうち, 評価値 $S(p)$ 上位 5 個の経路を図 11 に示す。図中の矢印の向きは, 写真の撮影時間に基づき付加したものである。向きが決まらないものは, 双方向矢印として表現している。図 12 には, 評価値が 2 番目に高かった経路候補 (長さ 3) を, 防災科学技術研究所が発表した竜巻被害の概況と分布 [17] に重ねたものである。これから分かるように, 列挙された竜巻候

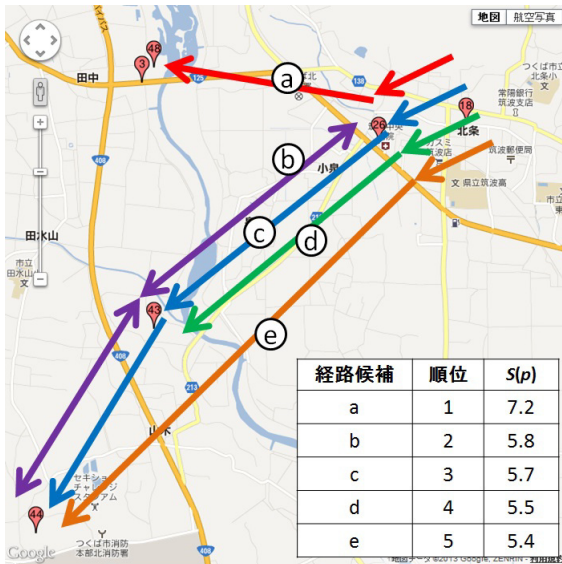


図 11 実験結果：求めた竜巻経路の候補
Fig. 11 Result: Candidates for the tornado path.



図 12 経路候補と竜巻による被害の概況と分布
Fig. 12 Damages caused by the tornado and a path candidate.

補のうち評価値が2番目に高い経路候補が、実際の被害と一致していることが確認できた。

図 11 に列挙されている経路候補のうち4つの候補が防災科研の発表した竜巻の進行方向と逆の方向を示しているが、その理由は、撮影時刻の入っている写真に写る竜巻の位置を、ワーカーが間違っって判断したためである。

6.4 考察

まず、本実験で用いたクラウドソーシング手法が、3.2節で説明した要件を満たしているかについて議論を行う。

要件1：タスクの作業時間。 Tukeyの提唱する外れ値の基準 [18]において、3IQRを基準に(すなわち、極端な外れ値だけを対象に)外れ値を除去した結果を用いて議論する。実行時間ごとのタスク数の分布を図 13, 図 14, 図 15, 図 16, 図 17に示す。各グラフは、外れ値を除外した結果を示したものである。タスクの作業時間の平均について、Task 1は40.8秒、Task 2は3.2秒、Task 3は7.7秒、

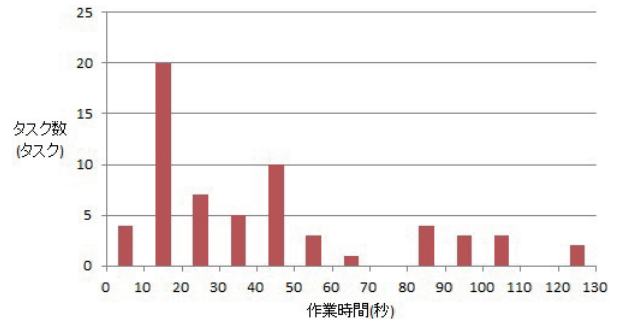


図 13 Task 1の作業時間
Fig. 13 Processing time of Task 1.

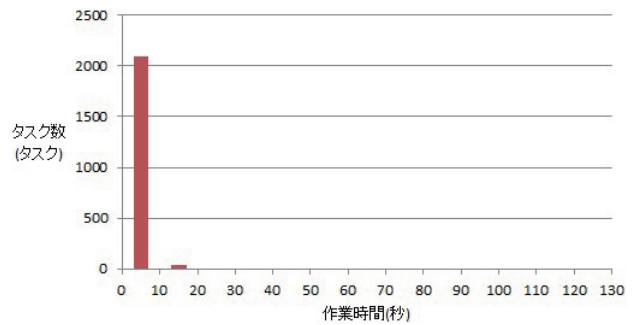


図 14 Task 2の作業時間
Fig. 14 Processing time of Task 2.

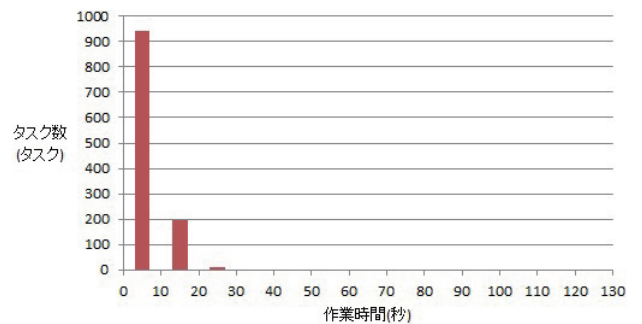


図 15 Task 3の作業時間
Fig. 15 Processing time of Task 3.

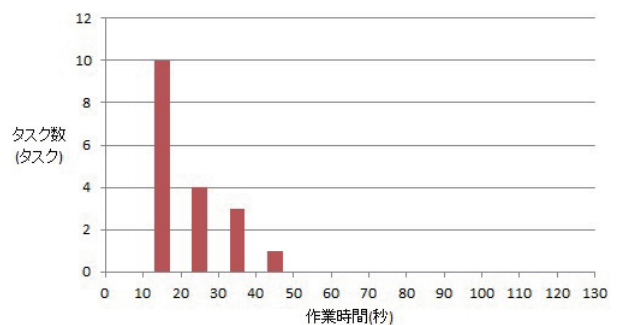


図 16 Task 4の作業時間
Fig. 16 Processing time of Task 4.

Task 4は21.2秒、Task 5は32.7秒であった。これらは、マイクロタスクとして十分短い時間であると考えられる。
要件2：災害エリアに関する知識の仮定。 実験後に確認したところ、実験に参加したワーカーは、全員、北条に関する

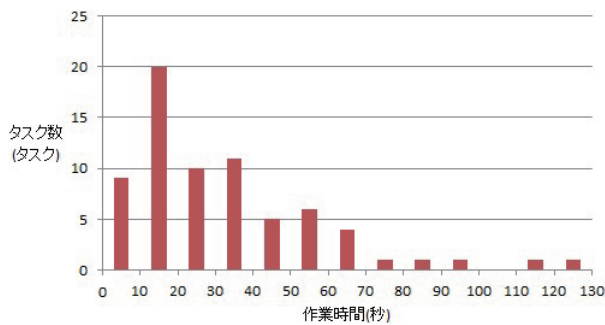


図 17 Task 5 の作業時間

Fig. 17 Processing time of Task 5.

詳しい知識を持っていなかった。

要件 3：結果の品質。 実験の結果、個々のタスクを見ると必ずしも精度の高い結果は得られていないが、“すべての結果が正しいことを仮定せず、候補を列挙する”アプローチにより、正しい竜巻経路を候補の1つとして求めることができた。

以上の結果から、本実験により、クラウドソーシングにより災害情報の把握支援を行うというアプローチが可能であることを示すことができたと考えられる。

次に、個別の知見に関して議論する。(1), (2) は、既知ではあるが本実験により重要性が再確認できた知見である。(3), (4), (5) は、今回の実験で特に得られた知見である。

(1) タスクの種類によっては、結果を完全に正しくすることは困難である。本実験では、正解不正解がはっきりしている Task 3 と Task 4 について、それぞれ 5/7 と 2/7 の正解率であった。理由はいくつか考えられるが、単に人は間違えるという理由だけでなく、そもそも、人が見ても判断困難なケースも多々あるからである。たとえば、Task 4 の結果は、被写体の位置を判定することは、少なくとも単純なタスクでは人手による判断が困難であることを示唆している。したがって、すべてのタスクの結果が正しいとは限らないという前提の仕組みが必要であることが確認された。

(2) 重要なタスクの精度はできるだけ上げるように工夫する。複雑なクラウドソーシングでは、あるタスクの結果が、別のタスクの結果に影響を及ぼす。たとえば、Task 3 では、6.2 節で述べたように、2つの写真に関して正しい撮影位置が得られなかったが、この原因は、Task 1 と Task 2 のタスクで、写真と写真が撮影された地図要素に同じキーワードが付与されず、そもそも Task 3 の判定の候補として採用されなかったからである。したがって、他のタスクへの影響を考慮し、重要なタスクには確認のフェーズを導入する等、データ品質に関する工夫をすべきである。

(3) 複雑なクラウドソーシングでは、結果の利用目的を明記すべきである。6.2 節で述べたように、Task 1 では、ワーカはキーワードを 85 個入力したが、そのうち 71 個のキーワードは Task 2 で不適切とワーカに判定された。このように多くのキーワードが不適切と判定された理由として、

ワーカが Task 1 のタスクと Task 2 のタスクの関係について理解していなかったことがあると考えられる。したがって、提案手法のように相互に関係するタスクをワーカに行ってもらえる場合には、ワーカにタスクの関係について理解させることが重要であると考えられる。

(4) トーナメント方式はマイクロタスク化に有効である。今回の写真と地図のマッチングのような判定が困難なタスクでは、単体での判定は難しいため、どうしても、複数の比較が必要となる。すなわち、写真1枚に対して、地図要素を何枚か見せて、その中で最も写真に近い地図要素を発見する等の工夫が必要である。これを行うタスクの設計は複数案が考えられる。たとえば、すべての候補となる地図要素を見せて、投票を行うといった方法が考えられる。しかし、今回のように候補の地図要素が多い場合（今回の Task 3 では写真1枚あたり最大 117）、短時間でタスクを処理することは困難である。今回、Task 3 が平均 10.3 秒の処理時間で行うことができた理由は、トーナメント方式のタスクを採用したからである。このように、比較評価をマイクロタスクで行うためには、トーナメント方式は有効であることが確認された。

(5) マイクロタスクの結果の正確さを担保できない場合、候補の列挙アプローチは有効である。(1) でも述べたように、マイクロタスクの結果がすべて正しいことを前提とすることは、タスクの種類によっては非現実的である。実際に、本論文で提案した竜巻経路候補列挙アルゴリズムが、正しい竜巻経路を出力できたことから、「タスクの結果に基づく候補列挙」というアプローチは、そのような状況に対応するアプローチとして有効であることが確認できた。

7. まとめと今後の課題

本論文では、クラウドソーシングを用いた災害状況把握を支援する手法として、マイクロタスク型の複雑なクラウドソーシングの適用を検討し、実験を通じて実現可能性の検討と知見の入手を試みた。その結果、本アプローチの適用可能性が十分あることが明らかになり、また、タスクの結果がすべて正しいことを前提とせず、候補を列挙するというアプローチが有効であることが確認できた。

本論文の重要な貢献は、災害状況把握の領域においてマイクロタスク型の複雑なクラウドソーシングが有効なアプローチであることを実験的に示した点であると考えているが、さらに精度が高く、効率の良いタスク設計の研究は今後の重要な課題である。具体的には、別のマイクロタスク設計の検討や比較だけでなく、タスクあたりのワーカの人数の最適化等に関する研究等も重要な課題である。また、本手法は竜巻だけでなく、局所的であり、かつ手がかりが残るような災害の状況把握に応用可能と考えられる。このような災害への適用可能性の検討も含めた、マイクロタスク設計の一般化も今後の課題である。

謝辞 本研究の一部は JST さきがけおよび科研費 (#70338309) の支援による。また、筑波大学とつくば市の連携協定に基づき研究を推進した。

p.688 (1977).

参考文献

[1] Wash, D.W.: *National Incident Management System Principles and Practice*, Jones and Bartlett Publishers (2005).

[2] 井ノ口宗成, 田村圭子, 古屋貴司, 木村玲玖, 林 春男: 緊急地図作成チームにおける効果的な現場型空間情報マッシュアップの実現に向けた提案—平成 23 年東北地方太平洋沖地震を事例として, *地域安全学会論文集*, No.15, pp.219-229 (Nov. 2011).

[3] Google Person Finder, available from (<http://google.org/personfinder/global/home.html>) (accessed 2013-06-20).

[4] sinsai.info: 東日本大震災—みんなでつくる復興支援プラットフォーム, 入手先 (<http://www.sinsai.info>) (参照 2013-06-20).

[5] 浦川 豪, 林 春男, 藤春兼久, 田村圭子, 坂井宏子: 2007 年新潟県中越沖地震発生後の新潟県災害対策本部における状況認識の統一, *地域安全学会論文集*, No.10, pp.531-541 (2008).

[6] Ushahidi, available from (<http://www.ushahidi.com>) (accessed 2013-06-20).

[7] Yang, D., Xue, G., Fang, X. and Tang, J.: Crowdsourcing to Smartphones: Incentive Mechanism Design for Mobile Phone Sensing, *MOBICOM 2012*, pp.173-184 (2012).

[8] Franklin, M.J., Kossmann, D., Kraska, T., Ramesh, S. and Xin, R.: CrowdDB: Answering Queries with Crowdsourcing, *SIGMOD Conference. 2011*, pp.61-72 (2011).

[9] Marcus, A., Wu, E., Karger, D., Madden, S. and Miller, R.: Human-powered Sorts and Joins, *PVLDB*, Vol.5, No.1, pp.13-24 (2011).

[10] Park, H., Pang, R., Parameswaran, A.G., Garcia-Molina, H., Polyzotis, N. and Widom, J.: An Overview of the Deco System: Data Model and Query Language; Query Processing and Optimization, *SIGMOD Record*, Vol.41, No.4, pp.22-27 (2012).

[11] Amazon Mechanical Turk, available from (<https://www.mturk.com/mturk/>) (accessed 2013-06-20).

[12] Morishima, A., Shinagawa, N., Mitsuishi, T., Aoki, H. and Fukusumi, S.: CyLog/Crowd4U: A Declarative Platform for Complex Data-centric Crowdsourcing, *PVLDB*, Vol.5, No.12, pp.1918-1921 (2012).

[13] Crowd4U, available from (<http://crowd4u.org>) (accessed 2013-06-20).

[14] Kittur, A., Smus, B., Khamkar, S. and Kraut, R.E.: CrowdForge: Crowdsourcing Complex Work, *UIST 2011*, pp.43-52 (2011).

[15] Google Maps, available from (<https://maps.google.co.jp>) (accessed 2013-06-20).

[16] 気象庁: 第一回竜巻等突風予測情報改善検討会資料 8 市原委員資料, 入手先 (<http://www.jma.go.jp/jma/kishou/known/toppuu/24part1/24-1-shiryo8.pdf>) (参照 2013-06-20).

[17] 防災科学研究所: 2012 年 5 月 6 日につくば市で発生した竜巻災害について, 入手先 (<http://mizu.bosai.go.jp/wiki/wiki.cgi?page=%BF%E5%A1%A6%C5%DA%BA%BD%CB%C9%BA%D2%B8%A6%B5%E6%A5%E6%A5%CB%A5%C3%A5%C8>).

[18] Tukey, J.W.: *Exploratory Data Analysis, 1st edition*,



丹治 寛佳

2013 年筑波大学情報学群情報メディア創成学類卒業。現在、同大学大学院システム情報工学研究科修士課程在学中。データ工学, クラウドソーシングに興味を持つ。日本データベース学会学生会員。



森嶋 厚行 (正会員)

1998 年筑波大学大学院工学研究科修士。博士 (工学)。AT&T 客員研究員等を経て、現在、筑波大学図書館情報メディア系/知的コミュニティ基盤研究センター教授。ACM, IEEE-CS, 電子情報通信学会, 日本データベース学会各会員。現在、日本データベース学会理事, 日本データベース学会論文誌編集委員, 情報処理学会データベースシステム研究会運営委員。情報統合, Web 情報管理, データ指向クラウドソーシング等に興味を持つ。



井ノ口 宗成

2005 年京都大学大学院情報学研究科社会情報学専攻修士課程修了。2008 年同大学院情報学研究科社会情報学専攻博士後期課程単位認定退学。同年同大学院情報学研究科社会情報学専攻博士号 (情報学) 取得。同年新潟大学災害復興科学センター特任助教。2011 年同大学災害・復興科学研究所特任助教。電子情報通信学会, 地域安全学会, 自然災害科学学会, 土木学会, 地理情報システム学会各会員。



北川 博之 (フェロー)

1980年東京大学大学院理学系研究科修士課程修了。日本電気(株)勤務の後、1988年筑波大学電子・情報工学系講師。同助教授を経て、現在、筑波大学システム情報系教授、ならびに計算科学研究センター教授。理学博士(東京大学)。著書「データベースシステム」(昭晃堂)等。情報処理学会フェロー、電子情報通信学会フェロー、日本データベース学会副会長、ACM、IEEE-CS、日本ソフトウェア学会各会員。

(担当編集委員 井上 創造)