

令和 7 年度 特別研究報告書

スマートフォンによる画像・位置情報連携を用いた
災害被害状況の自動マッピングシステム

龍谷大学 先端理工学部 知能情報メディア課程

Y210133 栗原 生吹

指導教員 三好 力 教授

内容梗概

近年、地震・台風・集中豪雨などの自然災害が全国各地で頻発化・激甚化しており、被害状況を迅速かつ正確に把握することが重要な課題となっている。災害発生直後は現場の状況が刻々と変化するため、行政機関や報道機関が全体像を把握するには一定の時間を要する。一方で、スマートフォンの普及により、被災者や市民自身が現場の映像や画像をリアルタイムで発信できるようになった。特に、動画や写真は文章だけでは伝わりにくい被害の深刻さや現場の臨場感を伝える有効な手段である。しかし、SNS などで発信される情報は断片的で位置情報が曖昧な場合も多く、全体の被害状況を体系的に把握するには限界がある。そこで本研究では、「スマートフォンによる画像・位置情報連携を用いた災害被害状況の自動マッピングシステム」を提案する。本システムは、市民がスマートフォンで撮影した被害状況の画像に、GPS による正確な位置情報と被害の種類を付与して報告できる仕組みを構築するものである。収集されたデータはサーバに送信され、Web 地図上に自動的にマッピングされる。マップ上では、被害の種類に応じてマーカーを色分け表示するなど、直感的に状況を把握できるようにする。これにより、行政機関や災害対策本部が現場の被害状況を迅速に共有でき、救助活動や避難誘導などの判断支援に役立てることが期待される。本研究では、位置情報付き画像の投稿アプリと自動マッピング型 Web システムを構築し、両者の連携による市民参加型情報共有が災害対応の迅速化・効率化に与える効果を検証する。

目次

第1章 はじめに	1
1.1 研究背景	1
1.2 研究目的	2
第2章 既存技術	3
2.1 公的機関による災害情報集約システム	3
2.2 市民参加型の災害情報収集	4
2.3 既存技術における問題点	5
第3章 提案手法	6
3.1 システム概要	6
3.2 主要機能	7
第4章 実験	9
4.1 実験目的	9
4.2 実験手順	9
4.3 実験システム	10
第5章 結果	13
5.1 システム動作の検証結果	11
5.2 可視化とインターフェースの有効性	13
第6章 考察	14
6.1 即時性と正確性の両立	14
6.2 視覚的な状況把握の優位性	14
6.3 今後の課題と展望	14
第7章 おわりに	15
謝辞	16
参考文献	17
付録	18

第1章 はじめに

1.1 研究背景

日本は、その地理的・地形的条件から、地震、台風、集中豪雨、火山噴火といった自然災害のリスクに常にさらされている。[1]図1から、日本は世界的に見ても極めて自然災害が多いことが分かる。近年、災害の頻発化・激甚化が顕著となり、私たちの社会生活や経済活動に深刻な影響を与えている。特に、大規模災害発生直後においては、被害状況が広範囲にわたり、かつ同時多発的に発生するため、行政機関や防災機関が全体の被害状況を迅速かつ正確に把握することは極めて困難である。人命救助や避難誘導を効果的に行うためには、どこで、どのような被害が発生しているのかという情報をリアルタイムに近い形で収集し、関係機関や市民の間で共有することが不可欠となる。

一方、スマートフォンの普及により、誰もが、現場の映像・画像を即時に記録・発信できるようになった。実際に、災害時には SNS 等に現場情報が多数投稿されることがある。しかし、これらの情報は位置情報が不明確であったり、情報の信憑性に問題があったりする場合も少なくない。また、情報が SNS のタイムライン上に散在するため、特定の地域や特定の被害状況に関する情報を効率的に検索・集約し、地図上で俯瞰的に把握することは依然として難しい状況にある。したがって、市民からの一次情報を位置情報とともに効率よく集約し、地図ベースで即時に可視化できるシステムが求められる。

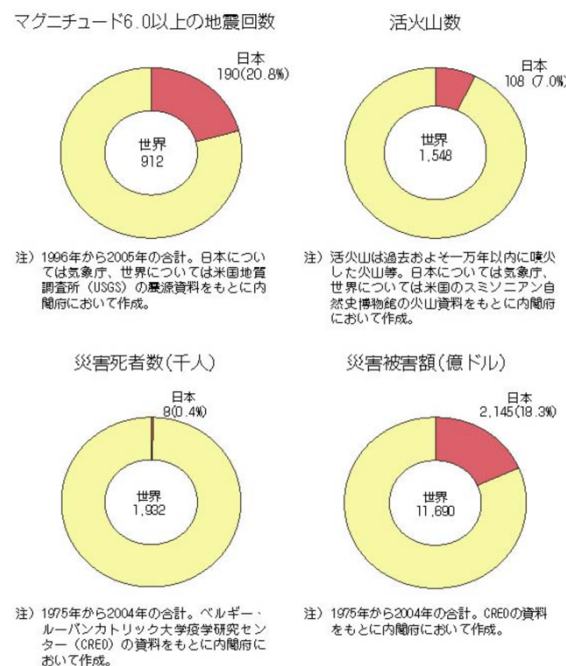


図1 世界の災害に比較する日本の災害

1.2 研究目的

本研究の目的は、災害発生時に市民がスマートフォンを用いて収集・発信する被害状況情報を効率的に集約し、Web 地図上でリアルタイムに可視化するシステムを提案・開発することである。これにより、従来の災害情報収集・共有における課題であった情報の断片性、位置情報の不確実性、集約・可視化の遅延といった問題を解決し、災害対応の迅速化・効率化に貢献することを目指す。

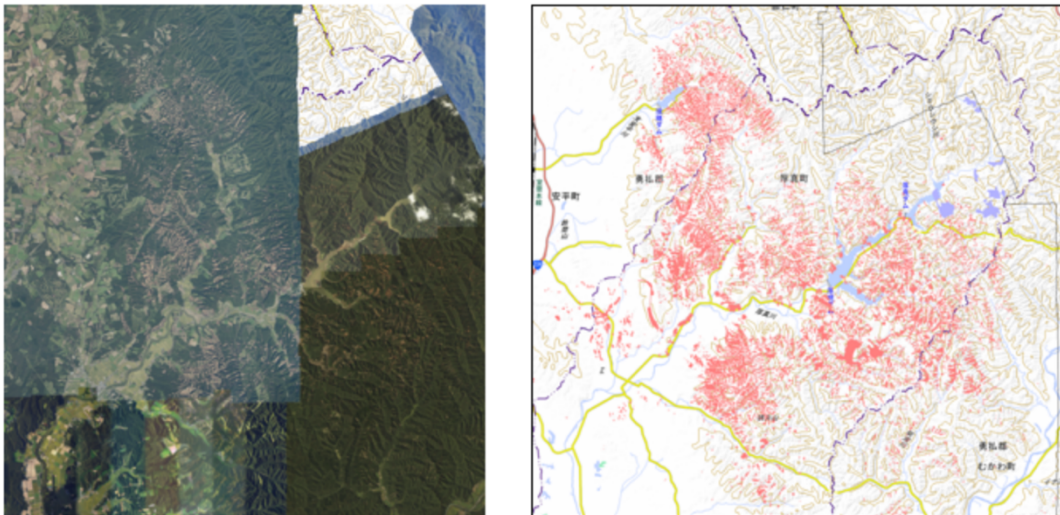
第2章 既存技術

2.1 公的機関による災害情報集約システム

大規模災害時の情報基盤は、国や自治体を確認した事実を集めて配信する「トップダウン型」が中心である。中核技術である GIS は、地図（地理空間）上に被害・避難所などの主題情報を重ね合わせ、可視化と分析を行う仕組みで、意思決定の土台として広く使われている。[2]日本では阪神・淡路大震災以後、GIS 整備が本格化し、地図と属性情報の一体的管理が進んだ。発災直後、国土地理院は被災域の「緊急撮影」を行い、空中写真から簡易オルソ画像を作成して図 2 のように地理院地図で公開する。位置補正された画像は、他の地理情報と重ね合わせて、広域の被害分布や土砂災害痕跡の把握に役立つ。[3]また、総務省が推進する「L アラート（公共情報コモンズ）」は、自治体が発する避難指示や避難所の開設、ライフライン情報などを一元的に集約し、放送・通信・Web などの事業者へ同報配信する仕組みで、公式情報を統一形式で広域に届けられる強みがある。[4]これらの公的システムは、情報の信頼性と網羅性、フォーマットの統一性に優れ、災害対応における「公式な情報基盤」として不可欠な役割を果たしている。

○斜面崩壊・堆積分布図

平成30年北海道胆振東部地震



（左）空中写真 （右）空中写真から判読された斜面崩壊・堆積分布図

図2 空中写真等による災害状況の把握

2.2 市民参加型の災害情報収集

スマートフォンと SNS の普及により、現場にいる市民や被災者が自ら状況を発信する「ボトムアップ型」の情報収集が広がった。テキスト、画像、動画を即時に投稿できる特性は速報性に優れ、細かな被害状況を細かく伝えられる。一方で、投稿の質や位置の明確さにはばらつきがあり、真偽の見極めや重複排除、地域ごとの集約には工夫が要る。近年は市民投稿を地図で集約する「クライシスマッピング」的な機能が広がり、図 3 のような Yahoo!防災速報の「災害マップ」では、利用者が周辺の異常を地図上で共有し、一定条件で異常感知通知も行う。[5]また特務機関 NERV 防災アプリも「クライシスマッピングレイヤー」を導入し、図 4 のように避難所・給水所・通行状況など生活支援情報をユーザー間で可視化している。[6]これらは公的情報を補完し、現場の細部を迅速に伝える新しい実践として位置づけられる。



図 3 Yahoo!防災速報の災害マップ

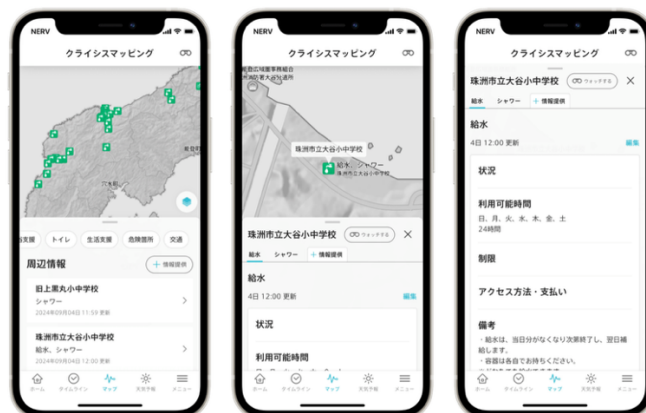


図 4 特務機関 NERV 防災アプリのクライシスマッピングレイヤー

2.3 既存技術における問題点

第一に、公的機関のトップダウン型システムの問題点は、情報が発信されるまでの時間差である。Lアラートや自治体のGISに載る情報は、基本的に職員が現場の状況を認知・確認し、所定の手順で入力してから配信されるため、発災直後のごく短時間には更新が追いつきにくい。国土地理院の航空写真等も広域把握に有効だが、撮影・処理に一定の時間を要する。結果として、初動期に重要な「いま・どこで・どの程度」というミクロな被害の把握をリアルタイムで積み上げていくことが難しい。

第二に、市民参加型のボトムアップ型システムの問題点は、情報の質と構造のばらつき、そして集約性の弱さである。SNSや商用防災アプリは速報性に優れ、細かな現場情報を集めやすい一方、投稿の位置精度や真偽の判定が課題となる。また、サービスごとにデータが分断されやすく、Yahoo!の投稿はYahoo!内、NERVの投稿はNERV内に留まるなど、行政のGIS等とシームレスに連携して一枚の地図に重ねることが難しい。

以上を踏まえ、本研究はトップダウン型の「遅さ」とボトムアップ型の「情報の曖昧さ・散在性」を同時に補い、位置情報付きの一次情報を即時に整理・可視化する枠組みを提案・検証することを目的とする。

第3章 提案手法

3.1 システム概要

本研究では、市民参加による災害状況の迅速な報告と可視化を実現するため、「スマートフォンによる画像・位置情報連携を用いた災害被害状況の自動マッピングシステム」を提案する。既存の公的機関によるトップダウン型システムは情報の信頼性が高い一方で発災直後の速報性に欠け、市民参加型のボトムアップ型システムは速報性に優れる一方で情報の位置精度や質が曖昧で、データが散在しやすいという問題点があった。そこで、現場の一次情報に正確な位置と被害カテゴリを付与し、即座に地図上で共有可能な Web アプリケーションを設計する。

システムは、インストール不要で動作する Web アプリケーションとして構築される。第一の要素は、市民が被害状況を報告するための「投稿インターフェース」である。スマートフォンのブラウザ上で動作し、カメラで撮影した写真・から Exif 情報を用いて緯度・経度を自動的に取得する仕組みを実装する。利用者は「浸水」「火災」「建物倒壊」「道路寸断」などの分類タグを選択するだけで、画像と正確な位置情報を紐付けたレポートを生成できる。

第二の要素は、収集された情報を集約し可視化する「マップビューア」である。Leaflet.js と OpenStreetMap を採用し、投稿されたデータ（位置・カテゴリ・メディア）を即座に同一画面内の地図へプロットする。被害カテゴリごとにマーカーの色を動的に変化させ、クリックすることで現地の写真や動画をポップアップ表示させる機能を備える。これにより、ブラウザ上で情報の入力から可視化までを完結させ、被災状況の分布を直感的に把握できる表示を目指す。

本研究では、これら投稿と閲覧の機能を単一の HTML ファイルとして統合的に開発し、プロトタイプによる実証実験を行う。専用アプリの導入障壁を排除し、URL へのアクセスのみで機能する仕組みとすることで、発災直後の「情報の遅れ」と「情報の散在」を同時に解決し、公的情報を実務的に補完しうる、即時性と信頼性を両立した低コストな市民参加型情報基盤の実現可能性を示す。

3.2 主要機能

本システムは、以下の 4 つの主要機能によって構成され、ユーザーによる情報の生成から共有、可視化までの一連のプロセスを実現する。

(1) スマートフォン連携による現在地・Exif 情報の自動取得機能

本システムは、HTML5 の Geolocation API および JavaScript ライブラリである exif-js を活用し、端末やメディアファイルから位置情報を自動的に抽出する機能を備えている。

- 現在地取得：アプリケーション起動時および動画投稿時に、Geolocation API を用いて端末の現在位置（緯度・経度）を取得し、地図の中心座標を自動的に更新する。これにより、被災現場での即時的な状況確認と投稿作業を支援する。
- Exif 解析：ユーザーがアップロードした画像ファイル（JPEG 形式）に対し、クライアントサイドで Exif メタデータの解析を行う。画像内部に記録された撮影地点の GPS 情報（GPSLatitude, GPSLongitude）を抽出し、度分秒（DMS）形式から十進数（Decimal Degrees）形式へ変換することで、撮影場所への正確なマッピングを実現する。なお、位置情報が付与されていない画像が選択された場合は、投稿をブロックするバリデーション機能を実装し、情報の信頼性を担保している。

(2) 被害カテゴリによる動的分類と色分け表示機能

災害情報の視認性を高めるため、投稿時に選択された被害カテゴリに基づき、地図上のマーカーを動的に色分けして表示する機能を実装した。

- カテゴリ定義：「浸水・冠水」「火災」「建物倒壊」「土砂崩れ」「道路寸断・陥没」「電柱・インフラ被害」「給水所情報」「支援物資・炊き出し」「救助要請」「その他」の計 10 種類の分類タグを定義した。
- 可視化ロジック：各カテゴリに対して固有の色（例：浸水は青、火災は赤）を割り当て、Leaflet.js の描画オプション（fillColor）を動的に変更することで、地図上で被害種別を直感的に識別可能とした。

(3) メディアプレビューを含むインタラクティブな情報表示機能

地図上のマーカーをクリックすることで、詳細情報をポップアップ表示する機能を実装した。

- ポップアップ詳細: カテゴリ名、状況説明テキストに加え、投稿された画像または動画のプレビューをインラインで表示する。動画ファイルの場合は HTML5 の<video>タグを用いてブラウザ上での再生を可能とし、静止画の場合はタグを用いて表示する。
- メディア処理: 選択されたメディアファイルは、URL.createObjectURL()メソッドを用いてブラウザ内部で一時的な Blob URL として生成・参照され、即座にプレビュー表示される仕組みとなっている。

(4) ローカルストレージを用いたデータ永続化機能

本プロトタイプにおけるデータ管理手法として、投稿されたデータ（位置情報、カテゴリ、テキスト）をブラウザのローカルストレージ（Web Storage API）に JSON 形式で保存する機能を実装した。

- データ保持: アプリケーションを再読み込み（リロード）または再起動した際にも、過去に投稿したマーカー情報が保持・復元される仕組みとなっている。
- 管理機能: 蓄積されたデータを一括で削除するリセット機能も備えており、実証実験や訓練などでの繰り返し利用を想定した設計となっている。

第4章 実験

4.1 実験目的

本研究で開発した「スマートフォンによる動画・位置情報連携を用いた災害被害状況の自動マッピングシステム」の有用性と実用性を検証するため、実地におけるフィールド実験を行う。実験の主な目的は以下の2点である。

1. システム機能の検証：スマートフォンのブラウザ上で、位置情報の取得、メディアのアップロード、マッピング、データ保存といった一連の機能が正常に動作することを確認する。
2. 可視化の有効性評価：収集されたデータが地図上でカテゴリ別に色分け表示されることで、災害状況の把握（どこで何が起きているか）が直感的に行えるかを評価する。

4.2 実験手順

本実験では、開発したシステムの有用性を検証するため、実地における動作検証を行った。実験にあたっては、河川の増水による浸水被害、道路の陥没、建物の倒壊など、複数の異なる災害事象が同時多発的に発生したシナリオを想定した。

具体的な手順として、まず実験者はフィールド内を移動しながら、スマートフォンのカメラを用いて対象となる地点や物体の撮影を行った。投稿操作においては、Exif情報が付与された画像データ（JPEG形式）を選択し、システムが画像内部のメタデータから正確に緯度・経度を抽出してマッピングできるかを確認した。あわせて、位置情報が含まれていない画像データのアップロードを試み、システムが「位置情報なし」と判定して投稿を拒否するバリデーション機能が正常に作動するかについても確認を行った。

データの投稿完了後には、マップビューア上での表示確認を行った。具体的には、投稿されたデータが地図上の正しい座標にプロットされているか、選択した被害カテゴリ（浸水、火災、道路寸断など）に応じてマーカーの色が適切に変化しているか、およびマーカーをクリックした際に現地の写真や状況説明を含む詳細情報がポップアップとして正しく展開されるかを評価した。さらに、ブラウザの再読み込み操作を行い、ローカルストレージに保存されたデータが消失することなく地図上に再描画されるかを確認することで、データの永続性についても検証を行った。

4.3 実験システム

本研究の実験のために開発した Web アプリケーションを以下に示す。

図 5 は Web アプリケーションの初期画面である。また、図 6 は、災害画像を投稿する際にカテゴリを選択できる画面である。

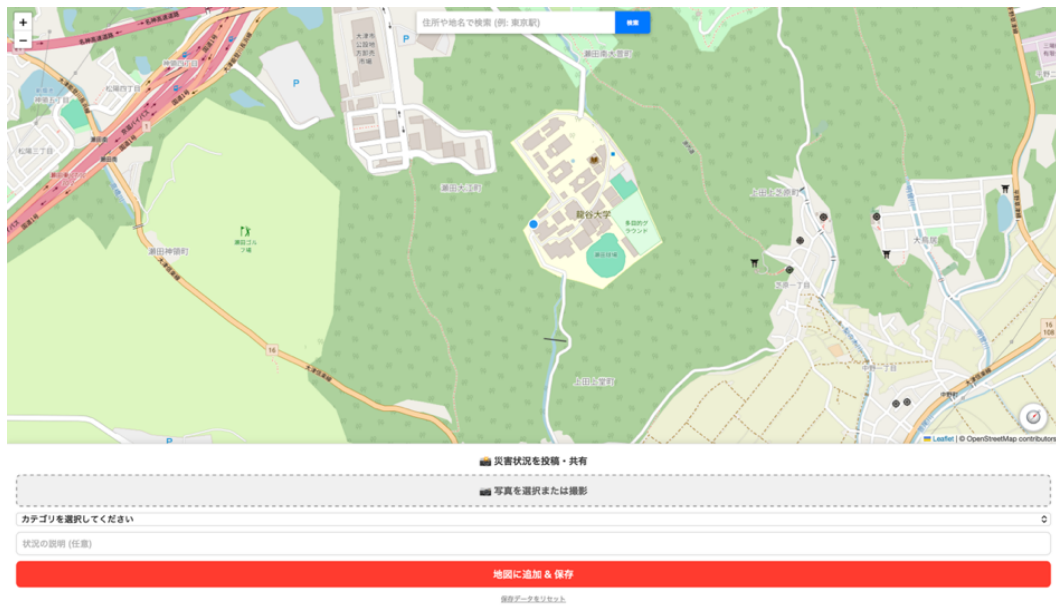


図 5：初期画面

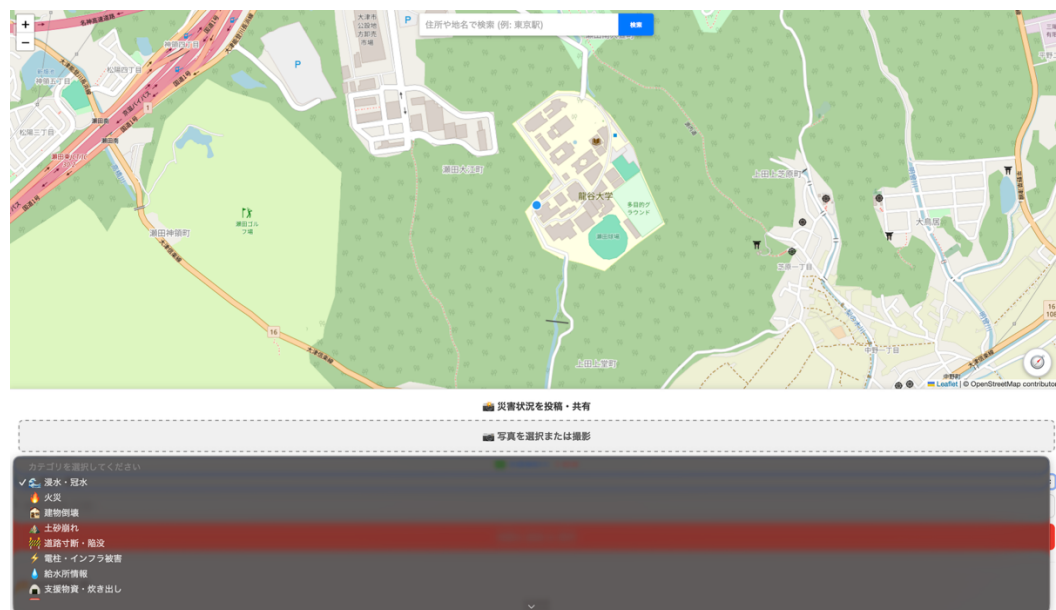


図 6：カテゴリ選択画面

第 5 章 結果

5.1 システム動作の検証結果

本実験において、開発した Web アプリケーションの動作検証を行った結果、以下の挙動が確認された。

第一に、位置情報付き画像データの処理についてである。実験者がスマートフォンで撮影した JPEG 画像を選択し「投稿」操作を行った直後、地図上の該当する撮影地点にマーカーが誤差なくプロットされることを確認した。その結果を図 7 に示す。これにより、システムが画像ファイルから位置情報を正しく読み取り、地図への反映を行えていることが視覚的に実証された。

第二に、位置情報の欠落に対するバリケーション機能の動作である。位置情報が付与されていない画像を選択した結果、画面上に即座に「投稿できません」という警告アラートが表示され、投稿処理が中断されることを確認した。その結果を図 8 に示す。これにより、位置情報を持たないデータの登録を防ぐ機能が正常に作動していることが確認された。

第三に、データの永続性についてである。投稿後にブラウザのリロード（再読み込み）操作を行ったところ、ローカルストレージに保存されていたデータが読み込まれ、地図上のマーカー配置およびカテゴリ・説明文などのテキスト情報が復元されることを確認した。なお、本システムはサーバーへのアップロードを行わないクライアントサイド完結型の仕様であるため、画像データ自体はリロード後に保持されず、位置情報とテキスト情報のみが再表示される挙動となった。

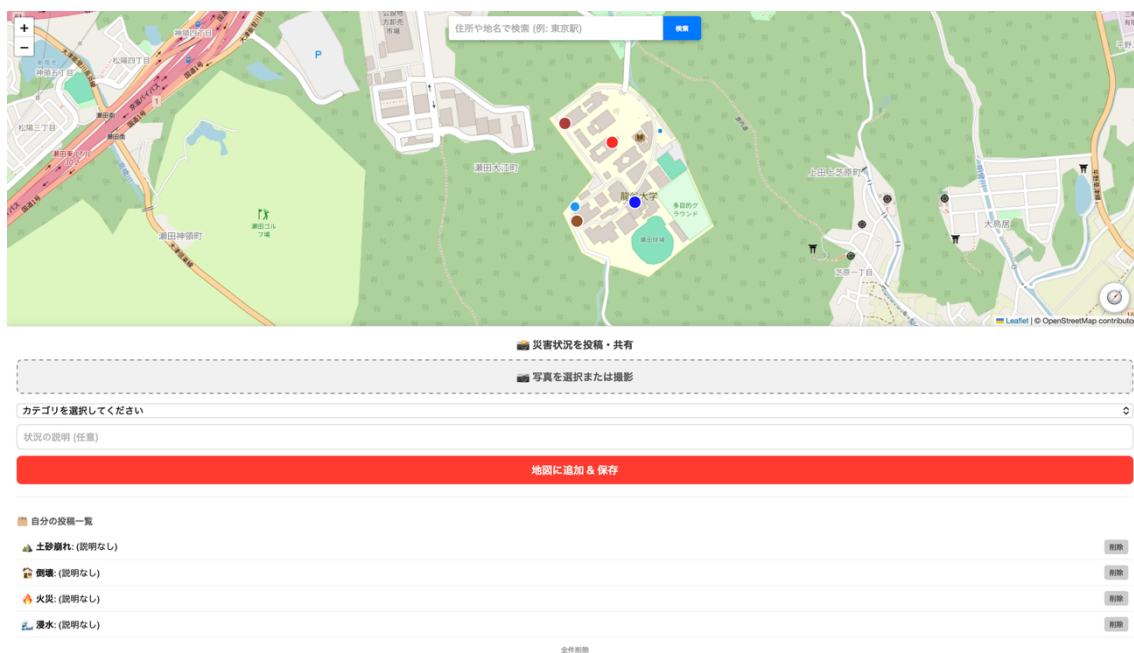


図 7：マッピング表示画面

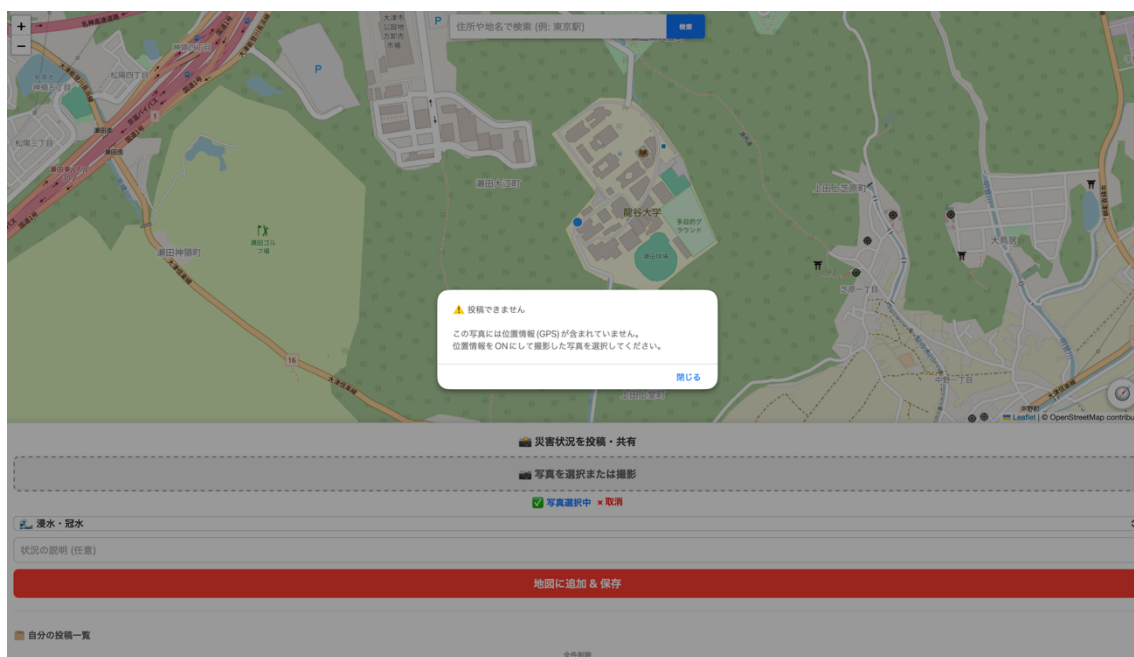


図 8：警告アラート画面

5.2 可視化とインターフェースの有効性

収集された災害情報の可視化およびユーザーインターフェースに関しては、以下の結果が得られた。

地図上の表示においては、投稿時に選択した被害カテゴリ（「浸水」「火災」「道路寸断」など）に基づき、マーカーの色が青・赤・黒などに正しく色分けされて表示された。これにより、地図を俯瞰するだけで、どのエリアでどのような種類の被害が発生しているかを直感的に識別可能であることが確認された。

詳細情報の閲覧においては、地図上のマーカーをクリックすることでポップアップウィンドウが展開され、現地の写真、状況説明テキスト、および正確な緯度・経度情報が表示された。写真はサムネイルとして視認性の高いサイズで表示され、テキスト情報と合わせて現地の状況を詳細に把握することができた。その結果を図9に示す。

また、アプリケーション起動時には Geolocation API によって端末の現在地が取得され、地図の中心が自動的に現在地へ移動することを確認した。さらに、画面上に配置された「現在地ボタン」を押下することで、地図をスクロールさせた後でも即座に現在地へ復帰できる挙動が確認され、フィールドワークにおける操作性の高さが示された。



図9：ポップアップ表示画面

第6章 考察

6.1 即時性と正確性の両立

本システム最大の特徴は、「位置情報の信頼性」をシステム側で担保している点にある。従来の SNS (X など) を用いた情報収集では、投稿テキスト内の地名 (「〇〇駅前」など) に依存するため位置特定が曖昧になりがちであった。しかし本システムでは、画像データの Exif 情報または端末の GPS 情報を必須とすることで、「位置情報がないデータは登録されない」という厳格な仕様を実現した。実験結果からも、投稿された全てのデータが正確な地図上の位置に紐付いていることが確認できた。これにより、災害対応において最も重要となる「正確な場所の特定」という課題を、市民のスマートフォンという身近なデバイスを用いて解決できる可能性が示された。

6.2 視覚的な状況把握の優位性

カテゴリ別の色分け表示は、情報の優先度判断において極めて有効であると考えられる。例えば、人命に関わる「火災 (赤)」や「建物倒壊 (茶)」と、インフラに関連する「道路寸断 (黒)」が色で区別されることで、対策本部は地図を一瞥するだけで対応すべき優先順位を直感的に判断できる。リスト形式のテキスト情報と比較して、地図を用いた可視化は、地理的な被害の広がりや傾向 (例: 川沿いに浸水被害が集中している等) を把握する上で圧倒的な優位性を持つと言える。

6.3 今後の課題と展望

本研究ではプロトタイプとして、ブラウザのローカルストレージを用いたデータ管理を行った。これにより「サーバーレスで即座に導入できる」という利点を示したが、一方で「他者とのリアルタイムな情報共有」までは実装のスコープ外とした。実社会での運用に向けては、投稿データをクラウド上のデータベース (Firebase 等) に送信し、複数のユーザー間で地図情報を同期するバックエンドシステムの構築が次のステップとなる。しかし、フロントエンド (投稿・表示) の仕組み自体は本研究で確立できており、このインターフェースをそのまま活用することで、大規模な災害情報共有システムへと発展させることが可能である。

第7章 おわりに

本研究では、大規模災害発生時における市民参加型の情報収集において、情報の即時性と位置情報の信頼性を両立させることを目的とし、「スマートフォンによる画像・位置情報連携を用いた災害被害状況の自動マッピングシステム」の構築と評価を行った。

提案システムは、専用アプリケーションのインストールを不要とし、Web ブラウザ上で動作する軽量なプラットフォームとして設計した。最大の特徴は、投稿される画像データから Exif 情報を自動解析し、正確な撮影地点を抽出して地図上にマッピングする機能である。また、位置情報が含まれていないデータの投稿を制限するバリデーション機能を実装することで、従来の SNS 等を用いた情報収集で課題となっていた位置情報の不確かさを、システム側で排除する仕組みを確立した。

実証実験の結果、本システムを用いることで、専門的な知識を持たないユーザーであっても、簡単な操作で正確な位置情報付きの被害レポートを作成できることが確認された。また、収集されたデータは被害カテゴリごとに色分けされて地図上に可視化され、被災状況の地理的な分布や優先順位を直感的に把握する上で高い有効性を示した。特に、現場の写真を地図上のマーカーと直接紐付けて表示するインターフェースは、テキスト情報のみでは伝わりにくい現地の詳細な状況を、迅速かつ正確に共有する手段として機能することが実証された。

一方で、本研究で開発したプロトタイプはクライアントサイド完結型の構成であり、投稿データの永続化と他者とのリアルタイム共有機能の実装には至らなかった。今後は、本研究で確立したフロントエンドの仕組みをクラウド上のデータベースと連携させ、複数ユーザー間での情報同期を実現することが課題となる。

結論として、本研究で提案したシステムは、スマートフォンの機能を活用することで、情報の信頼性と視覚的な分かりやすさを兼ね備えた災害情報収集基盤となり得るものであり、今後の防災・減災活動を支援する有効なツールとして発展する可能性を示した。

謝辞

今回の研究を行うにあたり、ご指導,ご助言して頂いた三好力教授に厚く感謝申し上げます。

参考文献

[1] 内閣府 防災省のページ

<https://www.bousai.go.jp/kaigirep/hakusho/h18/bousai2006/html/honmon/hm01010101.htm>

[2] 国土交通省 国土地理院：GIS とは・・・

<https://www.gsi.go.jp/GIS/whatisgis.html>

[3] 国土交通省 国土地理院：被災状況を把握し、災害復旧・復興を支援する

<https://www.gsi.go.jp/GSI/BOUSAI/page5.html>

[4] 一般財団法人 マルチメディア振興センター：L-ALERT

<https://www.fmmc.or.jp/commons/>

[5] YAHOO!JAPAN 防災速報：災害マップ

<https://notice.yahoo.co.jp/emg/sokuho/service/userreportmap.html>

[6] 特務機関 NERV 防災

<https://nerv.app>

付録

```
<!DOCTYPE html>
<html>
<head>
<title>災害可視化マップ (GPS 必須版) - 機能追加版</title>
<meta charset="utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no">

<link href="https://unpkg.com/leaflet@1.9.4/dist/leaflet.css" /> rel="stylesheet"
<script src="https://unpkg.com/leaflet@1.9.4/dist/leaflet.js"></script>
<script src="https://cdn.jsdelivr.net/npm/exif-js"></script>

<style>
/* レイアウト */
body { margin: 0; padding: 0; display: flex; flex-direction: column;
height: 100vh; font-family: sans-serif; }
#map { flex-grow: 1; z-index: 1; position: relative; background-color: #eee; }

/* 検索ボックス */
#search-box {
position: absolute; top: 10px; left: 50%; transform: translateX(-50%);
z-index: 1000; width: 90%; max-width: 400px;
display: flex; box-shadow: 0 2px 6px rgba(0,0,0,0.3);
}
#search-input { flex: 1; padding: 10px; border: none; border-radius: 4px 0 0 4px; font-size: 16px; }
#search-btn { padding: 10px 20px; background: #0078ff; color: white; border: none; border-radius: 0 4px 4px 0; font-weight: bold; cursor: pointer; }

/* 現在地に戻るボタン */
#gps-btn {
position: absolute; bottom: 20px; right: 20px; z-index: 1000;
background: white; border: 2px solid #ccc; border-radius: 50%;
width: 50px; height: 50px; cursor: pointer;
box-shadow: 0 2px 6px rgba(0,0,0,0.3);
display: flex; justify-content: center; align-items: center;
font-size: 24px;
}
#gps-btn:active { background: #f0f0f0; }

/* 投稿エリア */
#controls {
padding: 15px; background: #fff; border-top: 1px solid #ddd;
box-shadow: 0 -2px 10px rgba(0,0,0,0.1); z-index: 2;
max-height: 60vh; overflow-y: auto;
}
h3 { margin: 0 0 10px 0; font-size: 16px; color: #333; text-align: center; }
h4 { margin: 15px 0 10px 0; font-size: 14px; color: #555; }

.input-group { margin-bottom: 10px; }
select, input[type="text"] {
width: 100%; padding: 10px; font-size: 16px; border: 1px solid #ccc; border-radius: 8px; box-sizing: border-box;
}

/* ファイル選択ボタン */
.file-upload-btn {
display: block; width: 100%; padding: 12px;
background-color: #f0f0f0; border: 2px dashed #999; border-radius: 8px;
text-align: center; cursor: pointer; color: #555; font-weight: bold;
}
.file-upload-btn:hover { background-color: #e0e0e0; }
#fileInput { display: none; }

/* ファイル取消ボタン */
#cancelFileBtn {
display: none; cursor: pointer; margin-left: 10px; color: red; font-weight: bold;
}

/* 投稿ボタン */
#submit-btn {
width: 100%; padding: 12px; background-color: #ff3b30; color: white;
border: none; border-radius: 8px; font-size: 18px; font-weight: bold; cursor: pointer;
}
#submit-btn:active { opacity: 0.8; }

.popup-media { width: 200px; margin-top: 5px; border-radius: 4px; }

/* 投稿一覧リスト */
#my-posts-list { list-style: none; padding: 0; margin: 0; }
#my-posts-list li {
display: flex; justify-content: space-between; align-items: center;
padding: 8px; border-bottom: 1px solid #eee; font-size: 14px;
}
.post-info { flex-grow: 1; overflow: hidden; white-space: nowrap; text-overflow: ellipsis; margin-right: 10px; }
```

```
.delete-post-btn {
padding: 4px 8px; background-color: #ccc; color: #333; border: none;
border-radius: 4px; cursor: pointer; font-size: 12px; flex-shrink: 0;
}
.delete-post-btn:hover { background-color: #bbb; color: #000; }

</style>
</head>
<body>

<div id="search-box">
<input type="text" id="search-input" placeholder="住所や地名で検索 (例: 東京駅)">
<button id="search-btn" onclick="searchLocation()">検索</button>
</div>

<div id="map">
<button id="gps-btn" onclick="moveToCurrentLocation()">現在地へ移動</button>
</div>

<div id="controls">
<h3>災害状況を投稿・共有</h3>

<div class="input-group">
<label for="fileInput" class="file-upload-btn">
写真を選択または撮影
</label>
<input type="file" id="fileInput" accept="image/jpeg"
onchange="updateFileName()">
<div id="fileNameDisplay" style="text-align: center; font-size: 14px; color: #0078ff; margin-top: 5px; font-weight: bold; display: flex; justify-content: center; align-items: center;">
<span id="fileNameText"></span>
<span id="cancelFileBtn" onclick="cancelFileUpload()">×
取消</span>
</div>
</div>

<div class="input-group">
<select id="categoryInput">
<option value="" disabled selected>カテゴリを選択してください</option>
<option value="flooding">洪水・冠水</option>
<option value="fire">火災</option>
<option value="collapse">建物倒壊</option>
<option value="landslide">土砂崩れ</option>
<option value="road_damage">道路寸断・陥没</option>
<option value="infrastructure_damage">電柱・インフラ被害</option>
<option value="supply_water">給水所情報</option>
<option value="supply_food">支援物資・炊き出し</option>
<option value="rescue">救助要請</option>
<option value="other">その他</option>
</select>
</div>

<div class="input-group">
<input type="text" id="textInput" placeholder="状況の説明 (任意)">
</div>

<button id="submit-btn" onclick="processImage()">地図に追加 & 保存</button>

<div id="my-posts-container" style="margin-top: 20px; border-top: 1px solid #eee; padding-top: 10px;">
<h4>自分の投稿一覧</h4>
<ul id="my-posts-list">
</ul>
<div style="text-align: center; margin-top: 10px;">
<button onclick="clearLocalStorage()" style="font-size: 12px; color: #888; background: none; border: none; text-decoration: underline;">全件削除</button>
</div>
</div>

</script>
// --- 1. 地図の初期化 ---
var map = L.map('map');
L.tileLayer('https://s.tile.openstreetmap.org/{z}/{x}/{y}.png', {
attribution: '© OpenStreetMap contributors'
}).addTo(map);

var localPins = [];
var markersById = {};
var currentLocationMarker = null;

// --- 2. 起動時の処理 ---
window.onload = function() {
loadFromLocalStorage();
moveToCurrentLocation();
};

// --- 現在地へ移動 ---
function moveToCurrentLocation() {
```

```

    if (!navigator.geolocation) {
        alert("位置情報が取得できません。デフォルトの場所を表示し
ます。");
        map.setView([35.681, 139.767], 5);
    }

    navigator.geolocation.getCurrentPosition(
        (position) => {
            var lat = position.coords.latitude;
            var lon = position.coords.longitude;
            map.setView([lat, lon], 16);

            if (currentLocationMarker) {
                map.removeLayer(currentLocationMarker);
                currentLocationMarker = L.circleMarker([lat, lon], {
                    radius: 8, color: "white", weight: 2, fillColor:
"#2196F3", fillOpacity: 1
                }).addTo(map).bindPopup("現在地");
            }
        },
        (error) => {
            console.log("現在地取得エラー: " + error.message);
            map.setView([35.681, 139.767], 5);
        }
    );

    // --- 3. 住所検索 ---
    function searchLocation() {
        var query = document.getElementById('search-input').value;
        if (!query) return;

        fetch('https://nominatim.openstreetmap.org/search?format=json&q={qu
ery}')
            .then(response => response.json())
            .then(data => {
                if (data && data.length > 0) {
                    map.setView([data[0].lat, data[0].lon], 16);
                } else {
                    alert("場所が見つかりませんでした");
                }
            })
            .catch(e => alert("検索エラー"));

    }

    // --- 4. 画像処理 ---
    function processImage() {
        var fileInput = document.getElementById('fileInput');
        var file = fileInput.files[0];
        var category = document.getElementById('categoryInput').value;
        var text = document.getElementById('textInput').value;

        if (!file) { alert("画像を選択してください"); return; }
        if (!category) { alert("カテゴリを選択してください"); return; }

        // Exif 読み取り開始
        EXIF.getData(file, function() {
            var latData = EXIF.getTag(this, "GPSLatitude");
            var lonData = EXIF.getTag(this, "GPSLongitude");

            if (latData && lonData) {
                var latRef = EXIF.getTag(this, "GPSLatitudeRef");
                var lonRef = EXIF.getTag(this, "GPSLongitudeRef");

                var lat = convertDMSToDD(latData[0], latData[1],
latData[2], latRef);
                var lon = convertDMSToDD(lonData[0], lonData[1],
lonData[2], lonRef);
                var imageUrl = URL.createObjectURL(file);

                var newPin = {
                    id: Date.now(),
                    lat: lat, lon: lon, category: category, text: text,
media_url: imageUrl
                };

                addPinToMap(newPin);
                saveToLocalStorage(newPin);
                map.setView([lat, lon], 16);

                // フォームリセット
                cancelFileUpload();
                document.getElementById('textInput').value = "";
                document.getElementById('categoryInput').selectedIndex
= 0;

                alert("投稿しました！");
            } else {
                alert("▲ 投稿できません¥n¥n この写真には位置情報
(GPS)が含まれていません。¥n 位置情報を ON にして撮影した写真を選択
してください。");
                cancelFileUpload();
            }
        });
    }

    function convertDMSToDD(degrees, minutes, seconds, direction) {
        var dd = degrees + minutes / 60 + seconds / (60 * 60);
        if (direction == "S" || direction == "W") dd = dd * -1;
        return dd;
    }

    // --- 5. ピン描画 ---
    function addPinToMap(data) {
        var color = "gray";
        if (data.category === "flooding") color = "blue";
        else if (data.category === "fire") color = "red";
        else if (data.category === "collapse") color = "brown";

```

```

        else if (data.category === "landslide") color = "saddlebrown";
        else if (data.category === "road_damage") color = "black";
        else if (data.category === "infrastructure_damage") color =
"purple";
        else if (data.category === "supply_water") color = "cyan";
        else if (data.category === "supply_food") color = "green";
        else if (data.category === "rescue") color = "orange";
        // else if (data.category === "other") color = "gray";

        var pinOptions = { radius: 10, fillColor: color, color: "#fff", weight:
2, opacity: 1, fillOpacity: 0.9 };

        var popupContent = `<b>${getCategoryLabel(data.category)}</b><br>${data.text}<br>`;
        if (data.media_url) {
            if (data.media_url.startsWith('http') ||
data.media_url.startsWith('blob')) {
                popupContent += ``;
            }
        }

        var marker = L.circleMarker([data.lat, data.lon], pinOptions)
            .addTo(map)
            .bindPopup(popupContent);

        if (data.id) {
            markersById[data.id] = marker;
        }
    }

    function getCategoryLabel(key) {
        const labels = {
            "flooding": "🌊 浸水", "fire": "🔥 火災", "collapse": "🏠 倒壊",
            "landslide": "🏔️ 土砂崩れ", "road_damage": "🛣️ 道路寸断",
            "infrastructure_damage": "⚡ インフラ被害", "supply_water":
"💧 給水所",
            "supply_food": "🍱 支援物資", "rescue": "🆘 救助要請",
            "other": "🌐 その他"
        };
        return labels[key] || key;
    }

    // --- 6. 保存と管理 ---
    function saveToLocalStorage(pinData) {
        var saveItem = {
            id: pinData.id,
            lat: pinData.lat,
            lon: pinData.lon,
            category: pinData.category,
            text: pinData.text,
            media_url: ""
        };
        localPins.push(saveItem);
        localStorage.setItem('disasterMapData',
JSON.stringify(localPins));
        renderPostList();
    }

    function loadFromLocalStorage() {
        var dataStr = localStorage.getItem('disasterMapData');
        if (dataStr) {
            localPins = JSON.parse(dataStr);
            localPins.forEach(pin => {
                if (pin.id) pin.id = Date.now() + Math.random();
                addPinToMap(pin)
            });
            renderPostList();
        }
    }

    function renderPostList() {
        var listEl = document.getElementById('my-posts-list');
        listEl.innerHTML = "";

        if (localPins.length === 0) {
            listEl.innerHTML = "<li style='color:#888;'>まだ投稿はあり
ません</li>";
            return;
        }

        localPins.slice().reverse().forEach(pin => {
            var li = document.createElement('li');
            var catLabel = getCategoryLabel(pin.category);
            var textPreview = pin.text ? pin.text : "(説明なし)";

            li.innerHTML = `
                <div class="post-info">
                    <b>${catLabel}</b>: ${textPreview}
                </div>
                <button class="delete-post-btn"
onclick="deletePost(${pin.id})">削除</button>
            `;
            listEl.appendChild(li);
        });
    }

    function deletePost(id) {
        if (!confirm("この投稿を削除しますか？¥n 地図からも削除されま
す。")) return;

        localPins = localPins.filter(pin => pin.id !== id);

        localStorage.setItem('disasterMapData',
JSON.stringify(localPins));

        if (markersById[id]) {
            map.removeLayer(markersById[id]);
            delete markersById[id];
        }
    }

```

```

    }
    renderPostList();
}

function clearLocalStorage() {
    if(confirm("保存されている投稿データを【全て】削除しますか？\n
この操作は取り消せません。")) {
        localStorage.removeItem('disasterMapData');
        localPins = [];
        markersById = {};
        alert("リセットしました。ページをリロードしてください。");
        location.reload();
    }
}

// --- ファイル名表示と取消 ---
function cancelFileUpload() {
    document.getElementById('fileInput').value = "";
    updateFileName();
}

function updateFileName() {
    var input = document.getElementById('fileInput');
    var textDisplay = document.getElementById('fileNameText');
    var cancelBtn = document.getElementById('cancelFileBtn');

    if(input.files.length > 0) {
        // ファイル名は表示せず、選択中であることを示す
        textDisplay.innerHTML = "👍<b>写真選択中</b>";
    } else {
        cancelBtn.style.display = "inline";
        textDisplay.textContent = "";
        cancelBtn.style.display = "none";
    }
}
</script>
</body>
</html>

```