

スマートフォン連携型の土砂災害関連情報TV通知システム におけるユーザビリティ改善の一検討

細川 拓也¹ 徳本 敬祐¹ 小林 真¹ 新 浩一¹ 西 正博¹

概要：集中豪雨による土砂災害では、避難の遅れによって多くの人的被害が生じている。例えば甚大な被害を西日本にもたらした2018年7月の豪雨では避難率は4.6%と低かった。住民の早期避難を促して被害を未然に防ぐために、利用者に配慮した防災情報提供システムの構築が重要である。我々の研究室では、土砂災害による人的被害を軽減するために、危険箇所を監視して撮影する電源自立型の土砂災害モニタリングシステムを構築して、地域住民へリアルタイムな土砂災害危険箇所監視画像を提供するシステムを開発している。さらに、地方自治体からの防災メールをもとにして避難情報と土砂災害危険箇所監視画像をTVモニタに通知する、スマートフォン連携型の土砂災害関連情報TV通知システムを開発してきた。本システムは既存の有線のインターネット回線を有していない家庭でも利用できるという特徴を有する。本研究では、スマートフォン連携型の土砂災害関連情報TV通知システムにおける、TVの情報表示系およびスマートフォンの操作系のユーザビリティを改善した。Python3の標準ライブラリであるTkinterを用いてTVの情報表示系を構築することで、情報表示系の動作を軽量化した。スマートフォンの操作系では、情報機器の取り扱いが不慣れな人でも利用しやすいように、スマートフォン操作系のボタンの配置、機能やデザインを変更した。

1. はじめに

集中豪雨による土砂災害では、避難の遅れによって多くの人的被害が生じている。例えば甚大な被害をもたらした2018年7月の豪雨では避難率は4.6%と低かった[1]。このような人的被害を未然に防ぐためには地域住民の早期避難が必要不可欠である。早期避難の実現のために、住民の早期避難を促す防災情報の提供と、利用者に配慮した情報伝達システムの構築が重要である[2]。災害情報を速やかにかつ広範囲に伝搬させ地域住民と共有する手法として、スマートフォン等の携帯端末で構成されるモバイルアドホックネットワークを用いた草の根災害伝搬システムが提案されている[3]。

我々の研究室では、自然災害による人的被害を軽減するために、土砂災害危険箇所を監視し撮影することができる電源自立型の土砂災害監視システムを構築している[4]。本システムは2015年5月から運用を開始して、2021年2月現在では広島市内8箇所に設置してある。撮影画像を本研究室のウェブサーバにアップロードしており、誰でもウェブブラウザから確認することが可能となっている。

さらに、情報機器の取り扱いが不慣れで自ら土砂災害関

連情報を収集することが難しい人でも利用できるように、土砂災害危険箇所の画像をTVモニタに出力するシステムを構築している[5]。本システムはスマートフォンと小型PCをBluetoothを用いて連携することで家庭に既存の有線のインターネット回線が整ってなくても利用することができる。さらにスマートフォンをコントローラとして利用して、任意の地点の監視画像を表示させることができる。

しかしながら、TVの情報表示系の動作が重くなるという課題と、スマートフォンでの操作系が扱いにくいという課題があった。TV表示系では、動作を続けるとGUI(Graphical User Interface)の構築に用いているウェブブラウザのメモリ使用量が増大し続けて、動作が重くなっていた。スマートフォン操作系では、コントローラのボタンが記号で構成されていたため、情報機器操作が不慣れな人には扱いにくかった。本研究ではこのシステムのTVでの情報表示系およびスマートフォン上での操作系のユーザビリティの改善を行なった。

2. 土砂災害モニタリングシステム

我々は、これまでに、土砂災害の危険箇所を24時間監視撮影して、撮影画像をウェブページとして地域住民が近隣の危険箇所を閲覧できる土砂災害モニタリングシステムを構築してきた[4][6]。図1に土砂災害モニタリングシ

¹ 広島市立大学大学院情報科学研究科

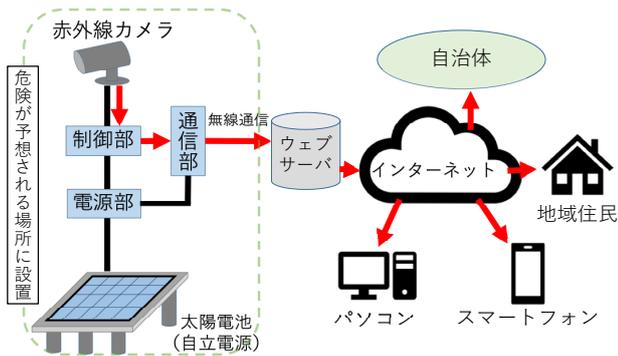


図 1: 土砂災害モニタリングシステムの概要



図 2: 土砂災害モニタリングシステムの外観

システムの概要を示す。本システムは、赤外線カメラ、太陽電池、制御部、電源部、通信部と、撮影画像を公開するウェブサーバから構成される。

図 2 に実際に設置、運用している本システムの外観を示す。撮影に赤外線カメラを用いることで、土砂災害の危険箇所を昼夜、天候問わず撮影可能である。太陽電池を用いることで、商用電源の確保が困難な環境でも 24 時間監視を可能とした。通信には 3G 回線を用いる。3G 回線の通信容量に限りがあるため、撮影時の解像度を、カメラ性能の最大値である 640×480 pixel から 320×240 pixel に下げて撮影している。さらに通信容量を考慮して、撮影アップロードの頻度を 5 秒間隔としている。

システムの制御部として、省電力かつ電力復旧時に自動で起動できる Raspberry Pi を用いている。ポート転送と SSH (Secure Shell) 接続を利用することで、システムにプライベート IP アドレスが割り当てられても遠隔操作できる。また、Raspberry Pi 再起動後に自動再接続する機能を実装しており、画像のアップロードを自動で再開する。SFTP (SSH File Transfer Protocol) を利用し画像情報をウェブサーバに自動的にアップロードする。アップロードした撮影画像はウェブページとして公開して、地域住民はリアルタイムな撮影画像を確認できる。

表 1: 災害情報伝達手段の比較

災害情報伝達手段	TV 通知システム	コミュニティ FM[7]	緊急速報メール [8]
情報伝達形態 (PUSH/PULL)	PUSH PULL	PUSH	PUSH
情報媒体	TV スマートフォン	ラジオ	スマートフォン
情報の形態	画像・文字	音声	文字
情報量	川や山の様子 警戒レベル	1 メッセージ 40~50 秒程度	約 200 文字程度
自動起動	TV が OFF でも自動起動 画面を自動切り替え	自動起動しない	マナーモードでも 警報音

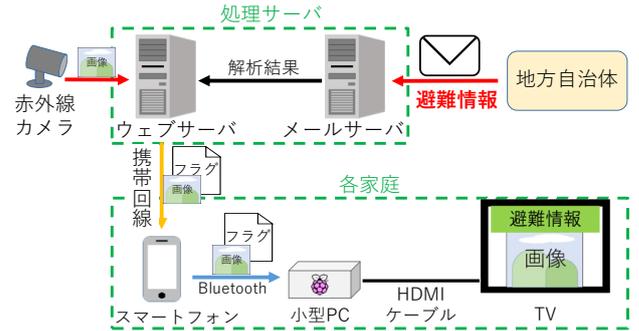


図 3: スマートフォン連携型の土砂災害警戒情報 TV 通知システムの基本構成

3. スマートフォン連携型の土砂災害関連情報 TV 通知システム [5]

情報機器の取り扱いに不慣れな人は、ウェブページに公開された土砂災害危険箇所のリアルタイムな監視画像を確認することが困難である。また家庭に有線のインターネット回線がないと利用できないという課題もある。情報機器の取り扱いに不慣れで、家庭に有線のインターネット回線がない人に向けて、土砂災害情報や監視画像を自動的に TV に表示するスマートフォン連携型の土砂災害関連情報 TV 通知システムを開発している [5]。

現在、日本では災害による被害者を減らすため多くの災害情報伝達システムが開発、提供されている。日本コミュニティ放送協会が運営するコミュニティ FM は、災害発生時に避難所の案内や被災者の安否確認などきめ細やかな災害情報を伝達する。NTT ドコモ、KDDI、ソフトバンクの三社で協調して提供している緊急速報メールは、指定されたエリア内の対応携帯電話に一齐に災害情報をメールで送信する。

表 1 に TV 通知システムとコミュニティ FM、緊急速報メールの比較を示す。コミュニティ FM はユーザが働きかけなくても情報を受け取ることが可能な PUSH 型である。40~50 秒程度の音声メッセージで情報を通知する。緊急速報メールは PUSH 型で、スマートフォンに 200 程度の文字を通知する。TV 通知システムは自動起動して情報を受け取ることが可能な PUSH 型かつ、ユーザが地点を選択して川や山の画像や警戒レベルを取得できる PULL 型である。



図 4: 従来のスマートフォン連携型の土砂災害関連情報 TV 通知システム

表 2: Raspberry Pi の諸元

モデル	Raspberry Pi 3 Model B
CPU	1.2 GHz (クアッドコア)
メモリ	1 GB
Bluetooth	4.1

3.1 システムの全体図

図 3 に本システムの構成を示す。本システムでは、まず地方自治体などから送信されたメールの内容をメールサーバで受信および解析して、避難情報の発令地域とレベルを特定する。メールサーバは解析結果をフラグファイルとしてウェブサーバに送信する。赤外線カメラで取得した監視画像もウェブサーバに送られる。

定期的にスマートフォンがウェブサーバのフラグファイルを確認して、変更があれば Bluetooth 経由で小型 PC に変更内容を送信する。小型 PC がフラグファイルを確認すると HDMI ケーブルで接続されている TV に小型 PC を写しているチャンネルへ切り替えるための信号を出すと同時に、避難レベルに応じた画面を TV モニタに出力する。

図 4 に本システムの外観を示す。TV モニタに表示される一例として南原地区を示す。TV モニタに現在の監視地点の監視画像、あらかじめ撮影しておいた通常時の昼と夜の監視画像、スマートフォンと小型 PC を接続するための QR コード、警報の有無、マップへ遷移するためのボタンが表示されている。

本システムでは、小型 PC として Raspberry Pi 3 を使用している [9]。表 2 に Raspberry Pi 3 の諸元を示す。Raspberry Pi 3 は手のひらに乗る程のサイズの PC である。Raspberry Pi 3 のモデルは Raspberry Pi 3 Model B、CPU は 1.2 GHz クアッドコア、メモリは 1GB である。スマートフォンには Google の提供している Pixel 3 を使用した [10]。オペレーティングシステムは Android 9、プロセッサは Adreno 630、メモリとストレージはそれぞれ 4GB、64GB である。

3.2 スマートフォンアプリケーション

開発フレームワークとして Cordova[11] を用いて本システムのスマートフォンアプリケーションを構築した。

3.2.1 Cordova

Cordova は HTML5、CSS3、JavaScript などの標準的なウェブ開発技術を利用して、クロスプラットフォームにおいて同一

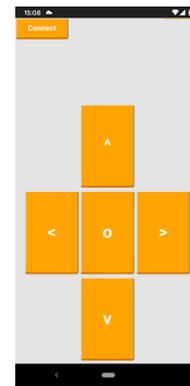


図 5: 従来のスマートフォンアプリケーションのキャプチャ

ソースコードにより開発を可能としたオープンソースモバイル開発フレームワークである。開発環境として Cordova を選択したのは以下の理由からである。

- iOS, Android を対象として同一ソースコードで開発が可能であり、同一アプリを大多数のスマートフォンユーザーが利用可能となる。
- 更新が頻繁に行われており、2019 年 8 月時点で 1 ヶ月以内に更新があったため今後もサポートが続くと考えられる。
- ウェブ開発技術で開発できるため、分かりやすい GUI を容易に開発することが可能である。
- 多数のプラグインが開発されており、機能を追加することが容易である。

また、本研究に必要な機能を実現するために以下のプラグインを使用した。

- (1) cordova-plugin-bluetooth-serial[12]: Bluetooth の SPP での通信をするためのプラグイン。
- (2) phonegap-plugin-barcodescanner[13]: カメラの起動と QR コードの読み取りまで行うプラグイン。

3.2.2 インターフェース

スマートフォンアプリケーションは小型 PC による TV 表示システムのリモコンとして機能する。図 5 にスマートフォンアプリケーションのキャプチャを示す。以下の機能を持つボタンが配置されている。

- (1) Connect キー：押すと TV システムに表示される接続用 QR コードを読み込むためのカメラが起動する QR コードを読み込むと、スマートフォンが TV システムに接続する。
- (2) 上下左右キー：それぞれの方向を表して、押すと TV 表示システム内の要素を移動する。
- (3) 決定キー (○印)：決定を意味する押すと選択している要素に応じて TV 表示システムが動作を起こす。

3.3 小型 PC による TV 表示システム

システムの小型 PC として Raspberry Pi 3 を使用している。Raspberry Pi 3 はスマートフォンからの Bluetooth 通信を待ち受けるための SPP サーバとして機能する。スマートフォンからは画像のバイナリデータとコマンドが送られてくる。受信した画像のバイナリデータとコマンドを UTF-8 でデコードして、

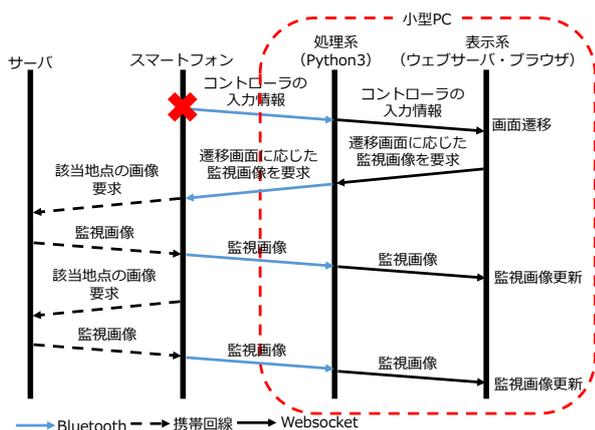


図 6: ウェブブラウザを用いたシステムの通信の様子

画像であるかコマンドであるか判別する。この処理は Python3 で動作するプログラムによって行われる。

Raspberry Pi 3 は TV と HDMI ケーブルで接続されていて、Raspberry Pi 3 の画面がテレビに出力される。TV システムの GUI は HTML, CSS, JavaScript で構成される。Raspberry Pi 3 で localhost に http サーバを立ち上げ、フルスクリーンモードのウェブブラウザで localhost にアクセスすることで GUI の表示を実現している。小型 PC の内部処理は主に Python3 で行われて、動的な処理は JavaScript で行われている。連携する手法として WebSocket を使用している。Python3 で localhost に WebSocket サーバを立ち上げ、Python3, JavaScript それぞれからクライアントとしてサーバに接続して、クライアント同士で文字列を送信する。それぞれのクライアントでその内容に応じた処理をする。

図 6 に通信の様子を示す。スマートフォンのボタンを選択すると、Bluetooth を用いてコントローラの入力情報が小型 PC の処理系に送信される。処理系から WebSocket を用いてコントローラの入力情報が表示系に送信される。表示系は受け取ったコントローラの入力情報に応じた画面遷移をする。表示系から WebSocket を用いて画面遷移に応じた監視画像の要求が処理系に送信される。処理系から Bluetooth を用いて画面遷移に応じた監視画像の要求がスマートフォンに送信される。スマートフォンは携帯回線を用いてサーバに該地点の画像を要求し、サーバから監視画像を受け取る。スマートフォンから Bluetooth を用いて監視画像が処理系に送信される。処理系から WebSocket を用いて監視画像が表示系に送信され、監視画像が表示される。

3.4 システムの動作例

南原地区を例に、TV システムの実際の表示画面を示す。図 7a, 図 7b, 図 7c, 図 7d に、それぞれ警報未発令時、避難準備情報発令時、避難勧告発令時、避難指示発令時の警報画面をそれぞれ示す。警報未発令時は「監視中」という文字と画面上部が緑色になる。避難準備情報発令時は「避難準備情報」という文字と画面上部が橙色になる。避難勧告発令時は「避難勧告」と

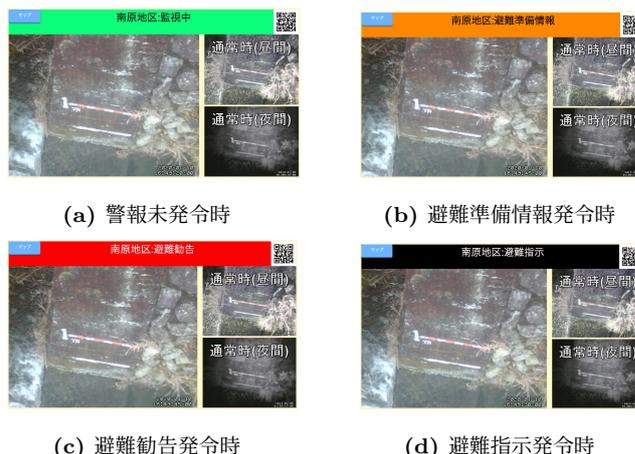


図 7: 従来の TV 表示システムにおける表示画面 (南原地区)



図 8: 従来の TV 表示システムにおける全体図 (マップ)



図 9: GUI の構築に Tkinter を用いたスマートフォン連携型の土砂災害関連情報 TV 通知システムの外観

いう文字と画面上部が赤色になる。避難指示発令時は「避難指示」という文字と画面上部が黒色になる。さらに、図 8 にマップの画像表示画面の例を示す。画面にある青色の要素は選択可能な要素である。濃い青色が選択状態で、薄い青色が未選択の状態である。選択地点を変更するためには Bluetooth で接続されているスマートフォンの上下左右キーのいずれかのボタンを押すことにより、任意の方向へ動かすことができる。

4. ユーザビリティの改善

3 節で示したシステムには、監視画像が更新される度にシステムが重くなるという課題と、ボタンが記号で構成されていて

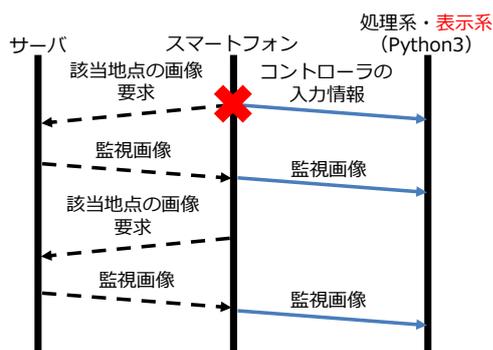


図 10: Tkinter を用いたシステムの通信の様子

情報機器の取り扱いに不慣れな人が利用しにくいという課題がある。以上の課題を解決するために、新たにスマートフォン上での操作系および TV モニタ上に情報を表示するための処理系の改善を行なった。図 9 に改善を行なったスマートフォン連携型の土砂災害関連情報 TV 通知システムの外観を示す。TV モニタには監視画像、表示している監視画像の地点名、警報の有無を表示する。スマートフォンは接続、全体図、監視地点名 (9 個) のボタンで構成されている。

4.1 TV 表示システムの改善

3 節で述べたシステムでは、GUI にウェブブラウザを用いることから、動作を続けるとメモリ使用率が上昇して、動作が重くなるという問題があった。本研究では、GUI の構築にウェブブラウザではなく Python3 の標準ライブラリである Tkinter を用いることで、TV 表示系のユーザビリティを改善した。

Tkinter は Python3 から GUI を構築、操作するための標準ライブラリである。Tcl/Tk の GUI ツールキットである Tk 部分を Python3 で利用できるようにしたもので、スクリプト言語である Python3 から簡単に GUI 画面をもったアプリケーションを作ることが可能になる。Tkinter を使用した理由は以下の 2 つである。

- (1) Python3 の標準ライブラリであること
- (2) 処理速度が速いこと

Python3 に標準で付属しているため、動作環境に Python3 が存在すると GUI の構築が可能である。他の GUI ライブラリをインストールする手間や、うまくインストールができないなどの問題が少ない。Tkinter の文法はシンプルで、起動や処理が早いので、TV システムの動作改善に適している。

図 10 に Tkinter を用いたシステムの通信の様子を示す。スマートフォンからコントローラの入力情報を Bluetooth を用いて小型 PC に送信する。その後入力に応じた画像データをスマートフォンから Bluetooth を用いて小型 PC に送信する。Tkinter を用いることで小型 PC からスマートフォンへの通信が必要なくなった。

さらに、3 節で述べたシステムでは、スマートフォンから送られてきた画像のバイナリデータとコマンドを UTF-8 でデコードしてから、画像であるかコマンドであるか判断していた。サイズの大きい画像のバイナリデータを文字としてデコードしよ

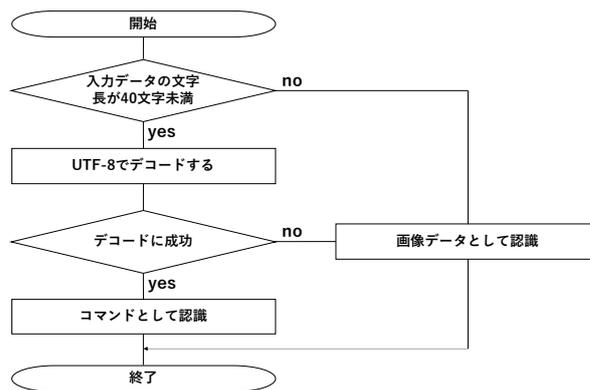


図 11: 画像のバイナリデータかコマンドかを判別する処理のフローチャート



図 12: 改善した TV 表示システムにおける表示画面 (南原地区の避難準備情報発令時)

うとすることで、動作が重くなっていたと考えられる。本研究では、画像データを文字としてデコードしないように変更した。図 11 に変更後の処理のフローチャートを示す。スマートフォンから送られてきたデータが、画像のバイナリデータかコマンドかをデコード前に判別できるようにした。送られてきたデータの文字長が 40 文字未満ならデコードしてコマンドとして認識する。もし 40 文字未満であってもデコードできなければ画像として認識する。送られてきたデータの文字長が 40 文字以上なら画像として認識しデコードしない。文字長の長い画像のバイナリデータをデコードしないため、デコード処理回数の削減を実現した。

実際に構築した TV 表示システムを示す。図 12 に、一例として避難準備情報が発令されている時の、南原地区の画面表示を示す。TV 表示システムの構成は以下のようになっている。

- (1) 監視画像
- (2) 現在表示している監視画像の地点名
- (3) 警報の有無 (警報未発令, 避難準備情報発令, 避難勧告発令, 避難指示発令)

表示している監視画像は一定時間経つと自動で更新されて、利用者が画像の更新のための操作を行う必要がない。地方自治体から送られてきた避難情報メールの内容を処理サーバで解析した結果が、小型 PC にスマートフォン経由で転送されている。小型 PC に転送された処理結果は、警報未発令なら 0、避難準備情報発令なら 1、避難勧告発令なら 2、避難指示発令なら 3 となっている。画像更新時に、処理結果の 0~3 が書き込まれたファイルを確認して、警報の有無の表示も更新する。



図 13: 改善したスマートフォンアプリケーションのキャプチャ

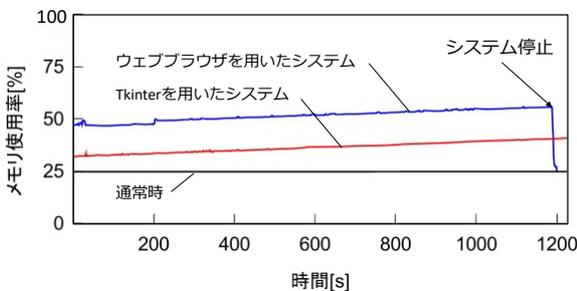


図 14: GUI 構築にウェブブラウザ, Tkinter を用いたシステムのメモリ使用率の比較

4.2 スマートフォン操作システムの改善

スマートフォン操作系のユーザビリティの改善を目的として、スマートフォンのボタンの構成と配置を改善した。図 13 に本研究で構築したスマートフォンアプリケーションのキャプチャ画像を示す。接続、全体図、9 箇所の監視地点名の計 11 個のボタンが配置されている。

それぞれのボタンごとの機能は以下の通りである。

- (1) 接続：カメラが起動して、TV システムに表示される QR コードを読み込むことで TV システムに接続する
- (2) 全体図：監視地点の全体図を TV システムに表示する
- (3) 監視地点名：ボタンに書かれた監視地点の監視画像を TV システムに表示する

ボタンを押すことでどのように動作するか分かりやすいようにボタンに表記する文字は全て日本語にした。各監視地点名の書かれたボタンに対応した監視画像を表示する機能を持たせることによって、1つのボタン操作で違う地点の監視画像への切り替えができるようになった。

5. 改善したシステムの評価

GUI の構築をウェブブラウザから Tkinter に変更したシステムを評価した。図 14 に GUI の構築にウェブブラウザを用いたシステムのメモリ使用率と、GUI の構築に Tkinter を用いたシ

ステムのメモリ使用率の比較を示す。システムを動作させて、1 秒ごとに 20 分間メモリ使用率を計測した。本稿で提案した GUI の構築にウェブブラウザを用いたシステムのメモリ使用率は平均約 27 %、GUI の構築に Tkinter を用いたシステムのメモリ使用率は平均約 12 %であった。GUI の構築に Tkinter を用いることで、メモリ使用率を平均約 15 %削減することができた。

スマートフォン操作系システムの改善によりボタンを押すことで、選択したボタンの地点名に対応した監視画像を表示できるようになったため、小型 PC からスマートフォンへの通信が必要なくなった。Python3 に標準で付属している Tkinter を用いることで画像表示のためにウェブブラウザと websocket を用いる必要がなくなり、小型 PC のリソースを節約することができた。

6. おわりに

本研究ではスマートフォン連携型の土砂災害関連情報 TV 通知システムにおける TV の情報表示系およびスマートフォン操作系のユーザビリティの改善を行なった。以下に改善した項目を示す。

- TV 表示系の GUI の構築をウェブブラウザを用いず、Python3 の標準ライブラリである Tkinter を用いた。
- スマートフォンから送られてくるデータを文字数によって処理方法を切りかえた。
- スマートフォン上のボタンの配置や機能、直接地域を選択できるようにした。

以上の変更によりシステムに用いている小型 PC のメモリ使用率を 15 %削減して、1つのボタン操作で閲覧したい監視画像の表示を可能にした。

今後、スマートフォン上のボタンを変更したことによる操作難易度の変化や、閲覧したい監視画像を表示するまでの時間を以前のシステムと比べることで、ユーザビリティが改善されたことを定量的に示すことを考えている。さらに、実際に情報機器の取り扱いが苦手な人にシステムを利用してもらい、システムの改善する予定である。

謝辞 本研究の一部は総務省 SCOPE 社会展開指向型 191608001, JSPS 科研費 17K06437 のもとに行なった。

参考文献

- [1] 広島市危機管理室危機管理課：平成 30 年 7 月豪雨災害における避難対策等の検証とその充実に向けた提言，平成 30 年 7 月豪雨災害における避難対策等検証会議，pp. 1-54 (2018).
- [2] Utsunomiya, E.: Information Propagation that Induces Evacuation Behavior at the Time of Disaster, *Proceedings of 2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, pp. 681-686 (2019).
- [3] 西 正博, 新 浩一, 河野英太郎, 井上伸二, 大田知行,

- 石田賢治, 宇都宮栄二, 角田良明: [招待講演] 土砂災害から身を守るための草の根情報伝搬システムの提案, 電子情報通信学会安全・安心な生活と ICT(ICTSSL) 研究会, Vol. 116, No. 251, pp. 31–36 (2016).
- [4] 西 正博, 古川達也, 新 浩一: 電源自立型の土砂災害監視カメラシステムの構築と評価, 情報処理学会論文誌, Vol. 58, No. 3, pp. 736–744 (2017).
- [5] 徳本敬祐, 小林 真, 新 浩一, 西 正博: スマートフォン連携型の土砂災害関連情報 TV 通知システムの開発, 電子情報通信学会総合大会, No. A-19-18, pp. 1–1 (2020).
- [6] 出原 聡, 小林 真, 新 浩一, 西 正博: 土砂災害前兆現象検知のための LPWA を用いたセンサシステムにおける送信スケジュール手法の一検討, 電子情報通信学会センサネットワークとモバイルインテリジェンス (SeMI) 研究会, Vol. 2021-MBL-98, No. 18, pp. 1–6 (2021).
- [7] 一般社団法人日本コミュニティ放送協会: コミュニティ FM の災害時における放送の確保について, 一般社団法人日本コミュニティ放送協会(オンライン), 入手先 (https://www.soumu.go.jp/main_content/000680839.pdf) (参照 2021/05/06).
- [8] 株式会社 NTT ドコモ, KDDI 株式会社, ソフトバンク株式会社: 緊急速報メールへの取り組みについて, 国土交通省(オンライン), 入手先 (https://www.mlit.go.jp/river/shinngikai_blog/hazard_risk/wg/dai01kai/dai01kai_siryoku2-4.pdf) (参照 2021/05/06).
- [9] Raspberry Pi Foundation: Raspberry Pi 3 Model B, RASPBERRY PI FOUNDATION (online), available from (<https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>) (accessed 2021/04/30).
- [10] Google: Google Pixel 3, Google (online), available from (<https://store.google.com/jp/product/pixel-3>) (accessed 2021/02/03).
- [11] Apache Software Foundation: Apache Cordova, Apache Cordova (online), available from (<https://cordova.apache.org/>) (accessed 2021/02/03).
- [12] Don: cordova-plugin-bluetooth-serial, Apache Cordova (online), available from (<https://github.com/don/BluetoothSerial>) (accessed 2021/02/03).
- [13] Erisu: phonegap-plugin-barcodescanner, Apache Cordova (online), available from (<https://github.com/phonegap/phonegap-plugin-barcodescanner>) (accessed 2021/02/03).