



全国高等専門学校 第25回

プログラミングコンテスト

プロコン公式サイト

<http://www.procon.gr.jp/>

高専プロコン 検索

※本選当日はライブ配信等を予定しています。

Twitter | @KosenProcon (公式アカウント)

#procon25 (ハッシュタグ)

とどけよう、イーハトーヴの風 ~僕らが創る希望郷~

本選
平成26年

10月18日(土)
~19日(日)

会場 一関文化センター

【入場無料】(岩手県一関市大手町2-16)

お問い合わせ

全国高等専門学校 第25回プログラミングコンテスト事務局

一関工業高等専門学校 学生課学生支援係

〒021-8511 岩手県一関市萩荘字高梨

TEL:0191-24-4718 FAX:0191-24-4530

E-Mail: jimu25@procon.gr.jp

審査部門

課題部門「防災・減災対策と復興支援」

自由部門

競技部門「キオクのかげらⅡ」

同時開催

NAPROCK 第6回国際プログラミングコンテスト

(NAPROCK 6th International Programming Contest)

<http://www.naprock.jp/IntProcon/>



わんこきょうたいおもち

主催 一般社団法人 全国高等専門学校連合会

共催 特定非営利活動法人 高専プロコン交流育成協会 (NAPROCK)、一関市

後援 文部科学省、総務省、経済産業省、岩手県、岩手県教育委員会、公益財団法人日本教育公務員弘済会岩手支部、一関市教育委員会、一関商工会議所、公益財団法人岩手県南技術研究センター、一般社団法人コンピュータソフトウェア協会、一般社団法人情報処理学会、一般社団法人電子情報通信学会、一般社団法人教育システム情報学会、独立行政法人情報通信研究機構、オープンデータ流通推進コンソーシアム、株式会社BCN、日刊工業新聞社、岩手日報社、岩手日日新聞社、NHK、IBC岩手放送、ICN一関ケーブルネットワーク、一関コミュニティFM、一関工業高等専門学校教育研究振興会、一関工業高等専門学校後援会、一関工業高等専門学校同窓会

特別協賛 東芝ソリューション(株)、アイピーシー(株)、(株)インテリジェントウェイブ、さくらインターネット(株)、(株)シーエーシー、(株)スペースタイムエンジニアリング、(株)トヨタコミュニケーションシステム、(株)ドリーム・アーツ、ネクストウェア(株)、富士通(株)、(株)ブロードリーフ

一般協賛 アイフォーコム(株)、インテル(株)、(株)NTTPCコミュニケーションズ、(株)NTTぷらら、(株)オプティム、(株)ぐるなび、セイコーエプソン(株)、ソニーデジタルネットワークアプリケーションズ(株)、(株)BCN、(株)フォーラムエイト、富士ソフト(株)、(株)豆蔵ホールディングス、(株)メンバーズ、ヤフー(株)、(株)ワコム

メディアスポンサー 週刊BCN、岩手日日新聞社

募集部門 パソコン等で実行可能なソフトウェア環境のもとで以下の3部門で競う。

1. 課題部門 「防災・減災対策と復興支援」をテーマにした作品
2. 自由部門 自由なテーマで独創的な作品
3. 競技部門 「キオクのかげらII」与えられたルールによる対抗戦

応募資格 国公立高専の学生（専攻科生を含む）

応募期間 平成26年5月23日(金)～5月30日(金)

審査

1. 予選（書類による審査）
日時 平成26年6月28日(土)
会場 関東ITソフトウェア健保会館・市ヶ谷会議室（東京都新宿区市谷仲之町4-39）
2. 本選（プレゼン・デモ等による審査、競技は対抗戦）
日時 平成26年10月18日(土)～10月19日(日)
会場 一関文化センター（岩手県一関市大手町2-16）

表彰 次の各賞を授与します。

課題・自由部門

最優秀賞 …… 各1点（賞状及び副賞）
優秀賞 …… 各1点（賞状及び副賞）
特別賞 …… 各数点（賞状及び副賞）

※最優秀賞には、文部科学大臣賞、情報処理学会若手奨励賞が授与される。

競技部門

優勝 …… 1点（賞状及び副賞）
準優勝 …… 1点（賞状及び副賞）
第3位 …… 1点（賞状及び副賞）
特別賞 …… 数点（賞状及び副賞）

※優勝には、文部科学大臣賞、情報処理学会若手奨励賞が授与される。

ポスターデザイン 一関工業高等専門学校 平成22年度本科（電気情報工学科）卒業生 吉田 竜一

NAPROCK

6th INTERNATIONAL PROGRAMMING CONTEST

INTRODUCTION

NAPROCK has co-sponsored the College of Technology Programming Contest (Kosen Procon) since 2008, and started to sponsor the NAPROCK international programming contest since 2009. In this year, the NAPROCK 6th international programming contest is held with the 25th Kosen Procon. This contest aims to promote flexibility of thinking through programming, at a very high level. At the contest, students from kosens or universities compete with each other by utilizing knowledge and ideas in information processing technology they have learned everyday of their lives. It is required for them to make full use of the latest and evolving information processing technology.

CONTEST INFORMATION

- **DATE:** October 18th (Sat.) – 19th (Sun.), 2014
- **VENUE:** Ichinoseki Cultural Center (2-16, Otemachi, Ichinoseki-shi, Iwate, Japan)
- **PARTICIPANTS:** KOSEN students (who participant in KOSEN Programming Contest), and students in domestic/foreign universities/institutes
- **EXAMINATION METHODS:**

Themed Section and Original Section:

Both of the presentation and demonstration will be examined by the judges. The examination standard includes originality, technical capabilities of systems development, usefulness, ease of use, manual/documentation preparation, speech and presentation skills, and so on. The operation manual and program source list will also be reviewed.

Competition Section:

Each team competes in a progressive tournament for victory.

- **AWARDS:**

In the Themed Section and the Original Section, the following prizes will be awarded.

Grand Prize: 1 team, Second Prize: 1 team, Special Prize: several teams

The following prizes are awarded in the Competition Section.

Champion: 1 team, First Runner-up Prize: 1 team, Special Prize: several teams

PROCON WEB SITE:

NAPROCK Procon official site: <http://naprock.jp/IntProcon/>
NAPROCK facebook page: <https://www.facebook.com/naprock2008>
Kosen-Procon official site: <http://www.procon.gr.jp/>

SPONSORS and SECRETARIAT:

Main sponsor: Nourishment Association for Programming Contest KOSEN
Co-sponsor: Technical College Association
Supporters: Nextware Ltd.
Toshiba Solutions Corporation
Other companies contributed to Kosen Procon support to this contest.

Supervising college: National Institute of Technology, Ichinoseki College
Judging Committee: Kosen Procon judging committee
Foreign Participants: Vietnam National University, Hanoi, Vietnam
Chengdu Neusoft University, China
Mongolian University of Science and Technology, Mongolia
Universiti Teknologi PETRONAS, Malaysia
Pathumwan Institute of Technology, Thailand

Domestic University Participants: Kyoto University / University of Tokyo
Toyohashi University of Technology
University of Tokyo

Secretariat: NAPROCK

ご挨拶

大会会長挨拶

全国高等専門学校連合会会長
東京工業高等専門学校校長

古屋 一仁



全国高等専門学校第25回プログラミングコンテストを、全国高等専門学校連合会、高専プロコン交流育成協会、そして一関市の主催・共催で、一関文化センターにて開催できますことを大変うれしく存じます。

我が国は、より高度なものづくりを創り出す必要に迫られております。創り出す訳ですから、未だ見えない一方で、際限ありません。この創造に意欲をもつ若者の活躍が大いに期待されます。創造の意欲と力を若者に着けさせることが肝要です。今こそ、そのための技術者教育に力を入れるときです。しっかりした学問を基礎に新たな社会を創り出すイノベティブな技術者を育てる必要があります。この教育に適したいくつかの仕掛けを高専は既に活用しています。学生がユーザーと価値を共創し、主体的に学び、その面白さを体験し、生涯学び続ける力を獲得するような仕掛けをこれまでに築き上げてきております。

国公立高専のIT力を競うコンテスト、高専プロコンは、学生が夢中になる、まさに、主体的に学ぶ格好の仕掛けの一つです。高専連合会は1990年（平成2年）から高専プロコンを主催して参りました。今では関連業界や学会から高く評価されています。今年も多数応募があり、厳正な予備審査で本選参加チームを決定しましたので充実した競技と発表をお楽しみいただければ幸いです。高度化を続ける高専教育を、IT教育という切り口でご覧いただくことになります。高専生の若さ溢れる感性・創造性・技術力、そして高専の技術者教育のレベルの高さをお感じいただけるものと自負いたしております。

最後に、ご後援いただきました文部科学省、総務省、経済産業省、岩手県、岩手県教育委員会、日本教育公務員弘済会岩手支部、一関市教育委員会、一関商工会議所、岩手県南技術研究センター、コンピュータソフトウェア・情報処理・情報処理通信関連の学会・協会・団体、報道機関、一関高専の諸団体、ご協賛頂きました企業、そして、プロコンにとって大変重要な審査をお引き受けいただいた先生方、高専プロコンを企画・運営された実行委員会の教員、主管校である一関高専の柴田尚志校長はじめ教職員の皆様、ご支援ご協力いただきましたすべての皆様に心より感謝申し上げます。

特定非営利活動法人 高専プロコン交流育成協会 (NAPROCK) 理事長挨拶

一般社団法人
情報処理学会 フェロー

神沼 靖子



全国高等専門学校第25回プログラミングコンテストが、2011年の大震災を乗り越えて、今日この一関市で開催されますことに心よりお喜び申し上げます。1年余りにわたってこの大会の企画・準備にご苦労されてこられた一関工業高等専門学校の皆様はじめ、高専プロコン委員の先生方に心から敬意を表します。1990年に始まった高専プロコンは毎年継続され、本大会で四半世紀を迎えることができました。この記念すべき大会までの足跡をたどりながら感慨に浸っているところでございます。本大会に参加できた学生諸君には、これからのIT社会で役立つアイデアと技術を思う存分披露されますことを期待しております。

NAPROCKは、全国の高専が「優れたエンジニアを育成すること」、そして「国際人としてのスキルを有する卒業生を輩出すること」を支援するために設立されました。その活動も7期目に入り、同時開催するNAPROCK国際プログラミングコンテストも第6回を数えることになりました。今回は、海外からベトナム、中国、モンゴル、マレーシア、タイの5ヶ国7チーム、及び日本の3大学3チームが参戦されるなど参加者も多様化し、素晴らしい大会になることが期待されます。

さらなる人材育成のために、NAPROCKはこれからもプロコン委員の皆さまと協働して多面的な支援活動を続けて参りますのでご支援のほど、よろしくお願い致します。

最後に、活動を支えてくださる協賛企業の皆さま、そして後援いただいている関係諸機関の皆さまに心から感謝を申し上げて、ご挨拶とさせていただきます。

プロコン委員長挨拶

一関工業高等専門学校 校長

柴田 尚志



記念すべき全国高専第25回プログラミングコンテストを一関で開催できますことを光栄に存じます。中東北の中心である一関市は仙台市と盛岡市のちょうど中間に位置し、昔から交通の要所として栄えてきた街です。巖美溪、狛鼻溪、栗駒山、室根山など風光明媚な名所が多く、隣の平泉町には毛越寺、中尊寺を中心とした世界文化遺産もあり、自然と文化が調和した街でもあります。一関市を含む北上山地は国際リニアコライダー(ILC)の国内建設候補地に決定しており、今後国際研究都市としての発展の可能性も持っています。

さて、今年のプロコンのキャッチフレーズは「とどげよう、イーハトーヴの風～僕らが創る希望郷～」です。イーハトーヴとは岩手県が生んだ有名な詩人であり童話作家でもある宮沢賢治による造語で、賢治の心象にある理想郷を表した言葉と言われていますが、賢治自身は語源について説明を残していないため諸説あるようです。皆さんにはICTを活用したそれぞれの希望郷を創ってほしいと思います。

今年のプロコンも、課題部門、自由部門、競技部門で争われますが、6月の予選を通過したのは課題部門21チーム、自由部門20チームであり、競技部門の参加59チームとあわせて本選で戦います。今回の課題部門のテーマは「防災・減災対策と復興支援」ですが、ご承知のように、ここ岩手県は東日本大震災で未曾有の被害を受けました。そのため、防災対策に関する意識も高く、講演会、セミナー等も頻繁に開催されています。今回学生の皆さんが発表する内容には多くの関係者が関心を持っています。直ぐにでも防災、減災、復興に役立つものができて欲しいと願っています。3部門に参加される学生の皆さんの活躍を期待しています。

最後に、本コンテストに協賛いただいている団体・企業の皆様、多方面でご支援をいただいている神沼理事長をはじめとする高専プロコン育成協会(NAPROCK)の皆様並びに高専連合会の関係の皆様へ感謝申し上げます、挨拶といたします。

大会日程

月日	会場	時間												
		8	9	10	11	12	13	14	15	16	17	18		
10月18日(土)	大ホール		参加者連絡会議(競技) 8:30-8:50		開会式 9:20-10:00		予行演習(競技) 10:10-13:00					1回戦(競技) 14:00-17:00		
	小ホール・展示室	(開場予定時刻8:00)	参加者連絡会議(課題/自由) 8:20-8:30	システム搬入チェック(課題/自由) 8:30-9:05		システムセッティング(課題/自由) 10:10-10:40	デモンストレーション一般公開(課題・自由) 10:40-17:00							
	中ホール					プレゼンテーション審査(課題) 10:10-17:00					学生交流会 17:20-18:20			
	研修室1					プレゼンテーション審査(自由) 10:10-17:00								
10月19日(日)	大ホール		参加者連絡会議(競技) 8:05-8:20	敗者復活戦・準決勝・決勝(競技) 8:30-14:00						特別講演会 14:30-15:00	閉会式 15:10-16:30			
	小ホール・展示室	(開場予定時刻8:00)	参加者連絡会議(課題/自由) 8:05-8:15	システムセッティング(課題/自由) 8:15-8:45	デモンストレーション審査・マニュアル審査(課題/自由) 8:45-12:00				システム稼働者引き渡し 14:00-14:30					
					デモンストレーション一般公開(課題/自由) 9:00-14:00									

特別講演会

講師：岩手県復興局復興推進課 推進協働担当課長 菊池 学 氏



講演題目：東日本大震災津波からの復興の取組状況について

～いのちを守り 海と大地と共に生きる ふるさと岩手・三陸の創造～

講演概要：

平成23年3月11日に発生した東日本大震災津波により、岩手県内では、多くの尊い命と財産が奪われました。震災発生直後から現在に至るまで国内外の多くの方々からご支援やご協力をいただいたことに対し、あらためて感謝申し上げます。

岩手県では、震災を乗り越えて力強く復興するための地域の未来の設計図として、「岩手県東日本大震災津波復興計画」を策定し、迅速かつ着実な復興の推進に取り組んでいます。

本講演では、震災から4年目を迎える被災地域の状況や課題、岩手県の復興への取組等をご紹介します。

審査委員

審査委員長	神沼 靖子	一般社団法人 情報処理学会 フェロー 特定非営利活動法人高専プロコン交流育成協会理事長
審査委員	井場 辰彦	株式会社シーエーシー 技術企画本部 アーキテクト
	岩井 靖	アイビーシー株式会社 常務取締役
	白井 支朗	豊橋技術科学大学 エレクトロニクス先端融合研究所 特任教授
	梅村 恭司	豊橋技術科学大学 情報・知能工学系 教授
	大岩 元	慶應義塾大学 名誉教授
	金田 茂	株式会社スペースタイムエンジニアリング プロジェクトマネージャ
	柴田 義孝	岩手県立大学 副学長／地域連携本部長／ソフトウェア情報学部 教授
	杉田 泰則	長岡技術科学大学 電気系 准教授
	千賀 大司	株式会社ブロードリーフ 技術企画部 部長
	竹俣 信榮	株式会社トヨタコミュニケーションシステム ビジネスシステム本部 BS2部 部長
	玉乃井慎児	NHK放送センター 放送技術局 メディア技術センター クロスメディア部 部長
	林 直樹	富士通株式会社 ヘルスケア・文教システム事業本部 文教第三ソリューション統括部長
	堀内 文	ネクストウェア株式会社 社長室
	前川 賢治	株式会社ドリーム・アーツ 取締役執行役員 VC企画開発本部長
	前川 徹	一般社団法人コンピュータソフトウェア協会 専務理事
	松澤 照男	北陸先端科学技術大学院大学 理事・副学長
	宮地 力	国立スポーツ科学センター スポーツ科学研究部 副主任研究員
	森 良哉	東芝ソリューション株式会社 IT研究開発センター 技監
	吉田 育代	フリーランスライター
	山本 祥之	株式会社インテリジェントウェイブ 代表取締役社長
	鷺北 賢	さくらインターネット株式会社 さくらインターネット研究所 所長

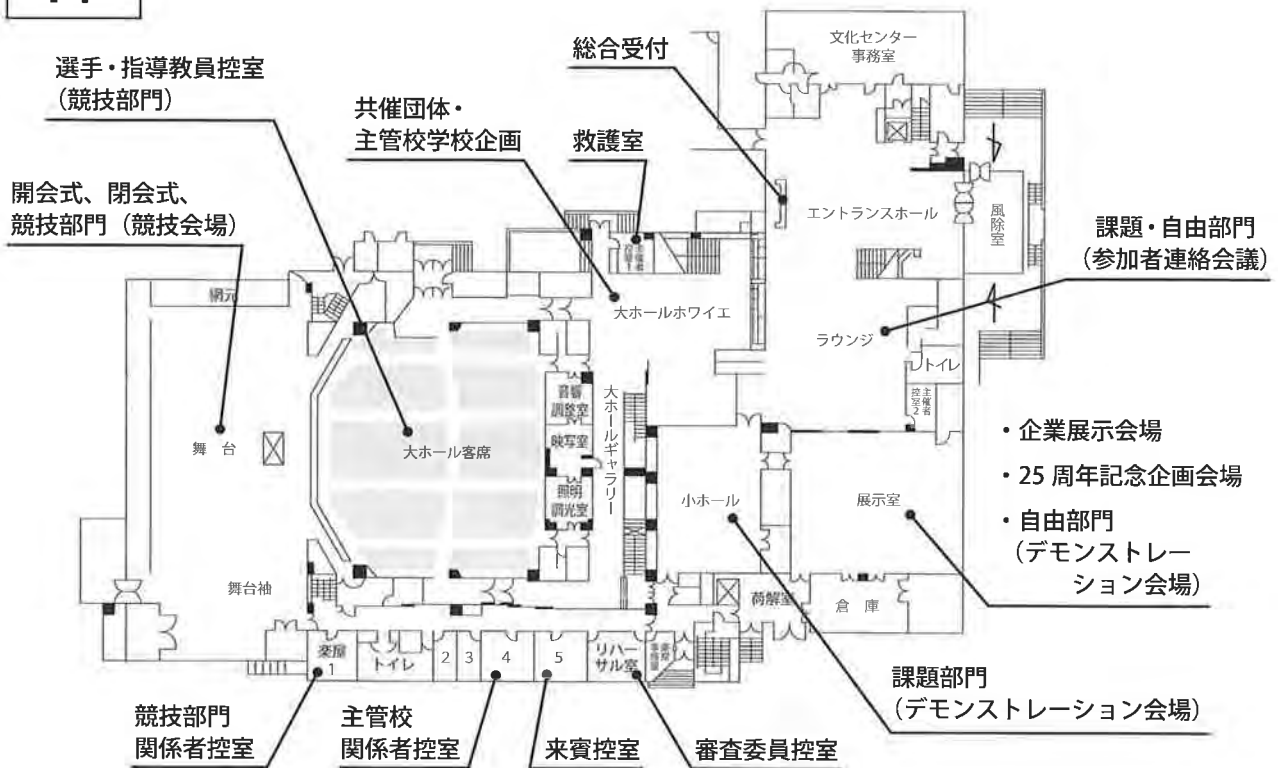
[マニュアル審査]

久保 慎一	ネクストウェア株式会社
津曲 潮	株式会社デザイン・クリエイション 顧問

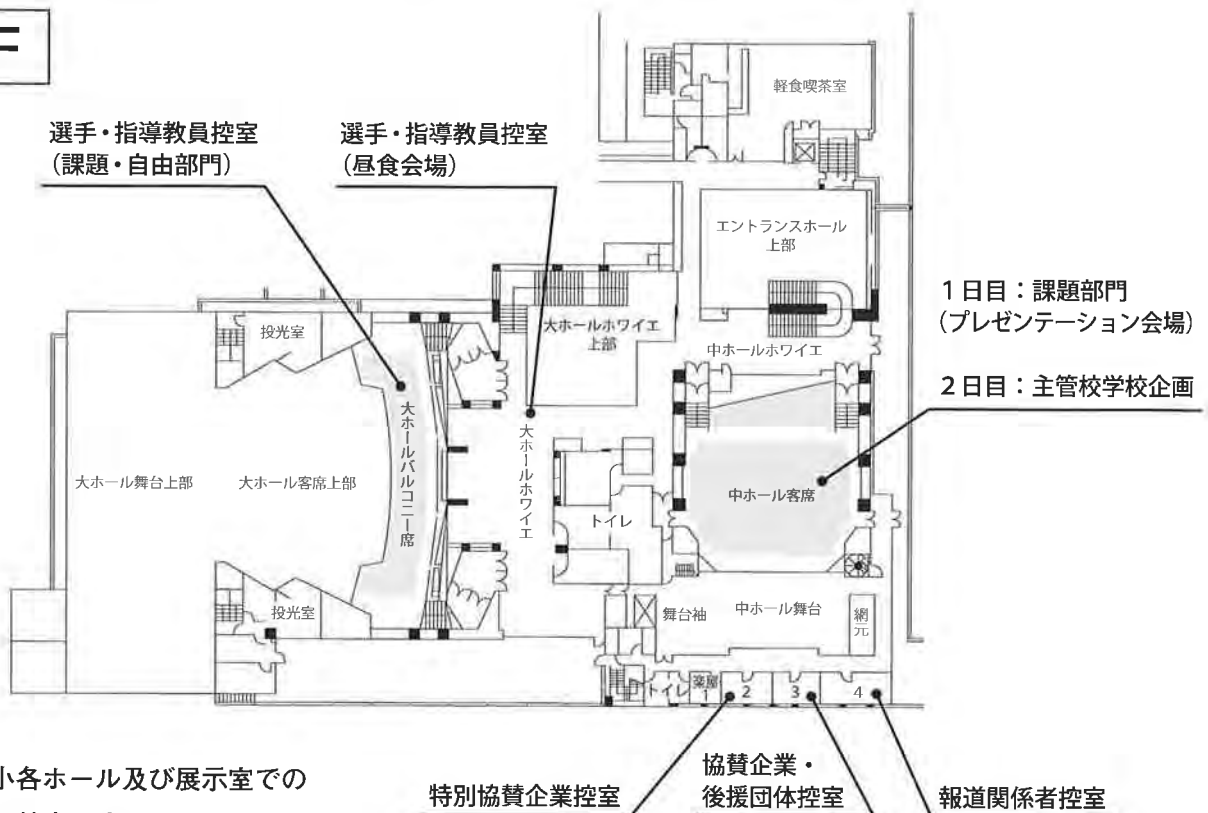
(五十音順 敬称略)

会場案内図

1F



2F



3F

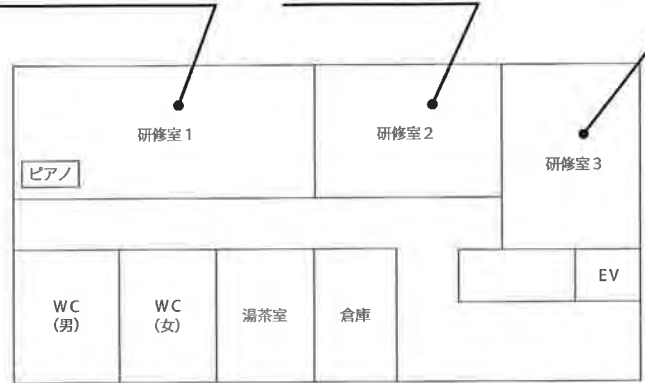
前 日：プロコン委員会（会議室）

1日目：自由部門
（プレゼンテーション会場）

2日目：主管校控室（荷物置き場）

プロコン委員会室

審査室・
審査委員会室



4F

一関高専
教職員・学生控室

海外チーム控室

主管校事務局



プログラミングコンテスト発展の経緯

高専プロコンは今回で25回の節目を迎えることになりました。これも、関係各位のお陰と感謝申し上げます。

今年度は一関高専が主管校となり、一関文化センターを会場に本選が開催されます。今回は171チームの応募があり、予選を通過した課題部門21チーム、自由部門20チーム、競技部門59チーム、国内大学3チーム、海外7チームの参加で本選が行われます。

プロコン発展の経緯について説明させていただきます。

本コンテストの主催団体は一般社団法人全国高等専門学校連合会（旧高等専門学校連合会）です。旧連合会の組織に高等専門学校情報処理教育研究委員会があり、情報処理教育に関わる全国高専の教員の代表が、種々の調査研究や催し物の立案を行っていました。平成元年8月、この委員会の常任委員会で全国の高専の学生を対象としたプログラミングコンテストの開催が採択され、この会を母体として本コンテストの実行委員会が編成されました。本コンテストは、情報処理技術の高揚や教員・学生の交流の機会拡大などの狙いもありましたが、高専が持つ若くて力強いエネルギーや発想の柔軟性を世の中に紹介したいという願いもありました。

第1回コンテストは、1年の準備期間を経て京都国際会館で開催されました。全国41高専の応募から、予選審査を経て、自由部門10テーマ、課題部門6テーマが本選に臨みました。初回の本選は、盛大な中にもアカデミックな香りが満ちあふれ、反響も極めて良いものでした。応募作品の一部はソフトハウスからアプローチを受けるなどの実績も得られました。第1回の成功以後、回を重ねるごとに規模も大きくなり、内容も充実してきています。

当初、課題・自由の2部門でスタートしたコンテストですが、第5回から競技部門を設け、3部門制で実施しています。高専全体のイベントとして定着するとともに、運営面でも変化が現れてきました。第4回大会から開催校（主管校）が本選の運営担当として設けられ、最近では、募集から本選開催にわたりコンテスト運営の中心となっています。

技術教育に主眼をおく高専においては、創造性・独創性を涵養する教育への取り組みが強く求められています。創造性教育のプロジェクトの一つとして、プロコンにも大きな期待が寄せられています。その目的を果たすため、作品の独創性を審査で重視するとともに、初回からプレゼンテーション審査とデモンストレーション審査の両方を課し、学生に対し表現力の涵養を計って参りました。主催団体である連合会も、教育プロジェクトとしてのプロコンの役割を重視し、下部機関としてプログラミングコンテスト委員会を独立して発足させ、プロコンのさらなる充実を図っています。

第2回からは文部省からもご後援を賜り、第4回からは念願の文部科学大臣賞を、第6回からは競技部門を含む全部門で文部科学大臣賞をいただけるようになりました。25回大会では、総務省、経済産業省の後援を賜りました。また18回課題・21回自由・22回課題部門の最優秀作品が第3～5回ものづくり日本大賞（内閣総理大臣賞）を連続受賞しました。プロコンの教育効果に対する評価も各界からいただいています。プロコン委員が、日本工学教育協会から工学教育賞を、情報処理学会からは教育賞を頂戴しております。

一方、高専が社会に対して貢献していくためには、産業界との連携も重要な課題の一つです。プロコンは第1回より（社）日本パーソナルコンピュータソフトウェア協会（現（一社）コンピュータソフトウェア協会）から後援をしていただ

き、絶大なご援助をいただいております。第1回は6社からスタートした協賛ですが、最近では20数社に及ぶ大きなご支援をいただけるようになりました。また、マスコミからもご後援を頂戴しております。このように、本コンテストの趣旨や意義がますます社会的に評価されてきたものと、喜ばしくまた有り難く思っています。更にプロコンを支援するNPO法人高専プロコン交流育成協会が平成20年7月に東京都の認可を受け、第19回本選から共催団体として加わりました。これも、後援団体および協賛企業はじめ各方面からのご支援があって実現したものと深く感謝しております。

プロコンの国際化も進んでいます。第8回ではオーストラリア、第10回では韓国への課題部門最優秀賞受賞チームの派遣が行われました。そして、第15回ではベトナムのハノイ工科大学をはじめオープン参加で受け入れ、これまで、ベトナム、モンゴル、中国、台湾、タイの6カ国から43チームを本選に迎えています。第20回大会より、NAPROCK国際プログラミングコンテストを同時開催しています。海外チームの招聘には、国際化にご協力いただいている企業およびNPO法人高専プロコン交流育成協会にご支援いただいております。25回大会では国内大学にも門戸が開かれました。

コンテスト本選では、いずれの部門も高専生のエネルギーと創造力を肌で感じることができると思います。次世代の日本を支える高専生のエネルギーと皆様のご支援を糧として、プロコンを核とした更なる展開を目指して努力したいと考えております。

回数	開催年	開催地	主管校	予選会場
第1回	平成2年	京都市		フォーラム8
第2回	平成3年	大分市		サンプラザ
第3回	平成4年	仙台市		東京文化会館
第4回	平成5年	名古屋市	豊田高専	都立高専
第5回	平成6年	富山市	富山商船高専	東京高専
第6回	平成7年	函館市	函館高専	東京高専
第7回	平成8年	北九州市	北九州高専	東京高専
第8回	平成9年	長岡市	長岡高専	東京高専
第9回	平成10年	明石市	明石高専	東京高専
第10回	平成11年	呉市	呉高専	都立高専
第11回	平成12年	津市	鈴鹿高専	都立高専
第12回	平成13年	鶴岡市	鶴岡高専	都立航空高専
第13回	平成14年	金沢市	石川高専	都立航空高専
第14回	平成15年	八王子市	東京高専	育英高専
第15回	平成16年	新居浜市	新居浜高専	都立高専
第16回	平成17年	米子市	米子高専	都立高専
第17回	平成18年	ひたちなか市	茨城高専	都立産技高専(品川)
第18回	平成19年	津山市	津山高専	都立産技高専(品川)
第19回	平成20年	いわき市	福島高専	サレジオ高専
第20回	平成21年	木更津市	木更津高専	田町CIC
第21回	平成22年	高知市	高知高専	サレジオ高専
第22回	平成23年	舞鶴市	一関・舞鶴高専	舞鶴市総合文化会館
第23回	平成24年	大牟田市	有明高専	都立産技高専(品川)
第24回	平成25年	旭川市	旭川高専	都立産技高専(品川)
第25回	平成26年	一関市	一関高専	関東ITソフトウェア 健保会館(市ヶ谷)

全国高等専門学校 第24回プログラミングコンテスト 本選結果

■ 課題部門

賞の名称	タイトル	高専名	指導教員	作成学生
文部科学大臣賞	緑 —ゆかりのある子育て—	東京	松林 勝志	五味 京祐, 松林 圭, 松尾 祐佳, 木暮 健悟, 大林 直樹
最優秀賞	緑 —ゆかりのある子育て—	東京	松林 勝志	五味 京祐, 松林 圭, 松尾 祐佳, 木暮 健悟, 大林 直樹
優秀賞	かぞくぐるみ —ぬいぐるみ型コミュニケーションロボット—	鳥羽商船	江崎 修央	鳥影 瑞希, NOROV ERDENE BATZAYA, 宮村 駿久也, 濱口 堅太, 小山 紗希
特別賞	SMASH —明るい未来にスマートシューズ—	豊田	平野 学	大林 真菜, 伊藤 啓二, 佐橋 広也, 水野 由基, 鳥飼 峻介
特別賞	えみシステム —ICTを活用した明るい子育て支援—	高知	今井 一雅	石野 達也, 宮本 悠史, 門田 皓成, 山形 亮水, 森本龍之助
特別賞	あいらんどっ —あんしん・あんぜんを いつもいっしょに—	沖縄	鈴木 大作	照屋のぞみ, 松井くるみ, 宮里 和裕, 仲嶺 真豪
特別賞	きゅ〜ぶめんと	香川 (詫間)	宮武 明義	大野健太郎, 入江 桃子, 大谷 優果, 木下 魁, 樋口 祐太

■ 自由部門

賞の名称	タイトル	高専名	指導教員	作成学生
文部科学大臣賞	すなケッチ!	鳥羽商船	江崎 修央	栗原 亨穂, 伊藤由祐紀, 稲田 樹, 萩原 海仁, 田川 瑞希
最優秀賞	すなケッチ!	鳥羽商船	江崎 修央	栗原 亨穂, 伊藤由祐紀, 稲田 樹, 萩原 海仁, 田川 瑞希
優秀賞	Z!BA	香川 (詫間)	金澤 啓三	白川 翔太, 西川 翼, 長谷川誠志郎, PARINDA WONGBENJARAT, 山崎 啓太
特別賞	SNOW-FIGHT —Anytime, Anywhere—	八戸	釜谷 博行	武田 瑞生, 日山 拓海, 坂本 和哉, 永田 大也, 木村 知夏
特別賞	USAKAME —娯楽型リハビリテーション支援システム—	広島商船	岩切 裕哉	倉田 茉季, 大野 信康, 山根 奈々
特別賞	Any Ware —Hard, Softに続く第3のware—	沖縄	正木 忠勝	日熊 悠太, 照屋 大地, 大田 有夏, 山城 響, 當間 環
特別賞	FaceList —顔リスト自動作成システム—	弓削商船	田房 友典	花岡久華莉, 檜垣 俊希, GURMU YIDIDYA GIRMA, 山岡 智美, 蔵田 千穂

■ 競技部門

賞の名称	タイトル	高専名	指導教員	作成学生
文部科学大臣賞	君に届けこの想い —想いを伝える魔法のサイコロ—	鈴鹿	浦尾 彰	石河 純輝, 武田 源生, 松田 薫
優勝	君に届けこの想い —想いを伝える魔法のサイコロ—	鈴鹿	浦尾 彰	石河 純輝, 武田 源生, 松田 薫
準優勝	TRIDE HC++	茨城	安細 勉	中村 泰大, 木下 郁章, 清水 琢見
第三位	今年度のサイコロの使い道が決定した会 —来年どうしよう—	鹿児島	豊平 隆之	福永 彬, 前田 哲志, 小永吉健太
特別賞	No Dice No Life	近畿大学 高専	政清 史晃	山下 陽大, 岡森 悠真, 矢羽田浩志
特別賞	パケットモンスター 赤・黒	徳山	力 規晃	有馬 諒, 近藤 佑樹, 杉原 航平
特別賞	サイコロの逆襲	都立 (品川)	福永 修一	池田 基樹, 小林 瑞樹, 前崎 公佑

NAPROCK PROCON2013

■ 競技部門

賞の名称	タイトル	高専・大学名	指導教員	作成学生
Champion	君に届けこの想い —想いを伝える魔法のサイコロ—	鈴鹿	浦尾 彰	石河 純輝, 武田 源生, 松田 薫
First Runner-up Prize	TRIDE HC++	茨城	安細 勉	中村 泰大, 木下 郁章, 清水 琢見
Special Prize	VNU-HANOI	ハノイ 国家大学	Xuan-Hieu Phan	Nguyen Tien Hoang, Viet-Ngoc Hua
Special Prize	ROOT POWER	モンゴル科 学技術大学	Khuder Altangerel	Ganbaatar Arslanbaatar, Khureltsooj Davaatseren
Special Prize	H3O	成都東軟 学院	Chen Wenfei	Huang Qichuan, Wang Yanheng

課題・自由部門について

●課題部門の概要

課題部門では、与えられた課題テーマに沿った独創的なコンピュータソフトウェア作品を募集しています。

今大会のテーマは「防災・減災対策と復興支援」となっています。近年、全国各地で多発している自然災害によって、毎年、たくさんの命や財産が失われています。課題部門では、このような自然災害における「防災・減災対策」と「復興支援」においてどのような課題や問題点があるのかを分析し、ICTや大規模に蓄積されたオープンデータを活用してその課題や問題点をどのように技術的に解決するのか。高専生ならではの切り口、独創的なアイデアで、被災地をそして日本を元気にするような魅力溢れる作品を期待しています。

今大会では、課題部門に56作品の応募をいただき、6月に東京で行われた予選審査において、書類選考によって21作品が選抜されました。これに海外からマレーシアのペトロナス工科大学チームを加えた22作品が本選に参加します。予選審査では、作品の独創性と課題との適合性が重点的に審査されるため、システムが完成していない設計コンセプトの段階での応募が可能となっています。本選ではこれらのアイデア段階の設計コンセプトがいかにして具体的な作品として実現されたかも審査の重要な項目の一つとなります。

本選では、次の4つのステージで審査されます。

- 1) 学会形式のプレゼンテーションによる審査
- 2) 実際に完成したシステムを動作させて説明する
デモンストレーションによる審査
- 3) 操作マニュアルの適正度のチェック
- 4) ソースリストのチェック

いずれのステージでも、独創性をはじめとして有用性・技術力・操作性などが総合的に審査されます。また、プログラミング能力のみならず、プレゼンテーション能力やマニュアル記述力など、総合的に評価する点が、高専プロコンの大きな特色となっています。

●自由部門の概要

自由部門では、参加者の自由な発想で開発された独創的なコンピュータソフトウェア作品を募集しています（第10回～第12回大会は、コンテンツを主体としたコンテンツ部門として実施されました）。

スマートフォンやタブレット端末が普及し、最近ではウェアラブルコンピュータのような新しいデバイスも登場してきました。また、クラウドコンピューティングやオープンデータの活用などインターネットを取り巻く環境も大きく変化してきています。本部門では、このような社会的背景において、既成の枠にとられない自由な発想で考案された独創的な作品を期待しています。過去の自由部門の優秀作品がIPA未踏ソフトウェアに採択されたり、マイクロソフト社のImagineCupで優秀な成績を残す等、コンテストの枠を超えて高い評価を得ており、アイデア創生と熟成の場として大きな役割を果たしていることがうかがえます。

今大会では、自由部門に56作品の応募をいただき、6月に東京で行われた予選審査において、書類選考によって20作品が選抜されました。これに海外からマレーシアのペトロナス工科大学チームを加えた21作品が本選に参加します。予選審査では、課題部門と同様の方法で作品の独創性が重点的に審査されました。

本選審査は、課題部門と同様にプレゼンテーション、デモンストレーション等により、学生のプレゼンテーション能力や作品の完成度等を含めて総合的に優秀な作品が選抜されます。

競技部門について

●競技部門の概要

競技部門は、第5回大会から導入されました。課題部門や自由部門と異なり、各チームの直接対決により勝敗を決します。競技内容は、コンピュータを用いた時間競争、精度競争、最良解探索競争等で、毎年異なるテーマで実施されています。

過去の大会では解を求めるだけでなく、実際に巨大迷路を使用したり巨大パズルを動かしたりして実演する競技や、ネットワークに接続されたコンピュータを利用した競技などが実施されました。そのため、解を求めるアルゴリズムが優れているだけでなく、問題の入力から解答の表示、更にミスへの対処、あるいは自動的にネットワーク通信等、あらゆる面で優れた、完成度の高いシステムが要求されてきました。

今年の競技部門では、1枚の原画像から同サイズに切り分けられた断片画像をバラバラに並べた問題画像から原画像に並べ替えるパズルゲームを行います(図1)。原画像は提供されないので、問題画像から原画像を素早く推測する必要があります。

●今大会の競技内容

「キオクのかげらⅡ」

第25回大会の競技部門は、1試合最大18チームによる勝ち抜け戦を行います。競技時間は1～15分とし、今回の競技は勝ち抜け方式で行います。1試合を複数の問題で行い、1問ごとに上位チームが勝ち抜け、次の試合に進みます。各チームはネットワークを経由し、問題画像を入手します。その後、独自の方法で、少ない回数で、隣り合う断片画像同士を入れ替えし、原画像に近づけて行きます。勝敗は以下のルールで決定します。

- ① 一致断片画像数(原画像との位置が一致した断片画像が多いチームが上位)
- ② 総コスト(総コストの少ないチームが上位)
- ③ 選択コスト(選択回数の少ないチームが上位)
- ④ 交換コスト(交換回数の少ないチームが上位)
- ⑤ サイコロの目で勝負(サイコロを振って、サイコロの目の合計が多いチームが上位)

18チームによる勝ち抜け戦ですが、チーム間の干渉はないので各チームのアルゴリズムの優劣がそのまま勝敗に繋がる競技になっています。

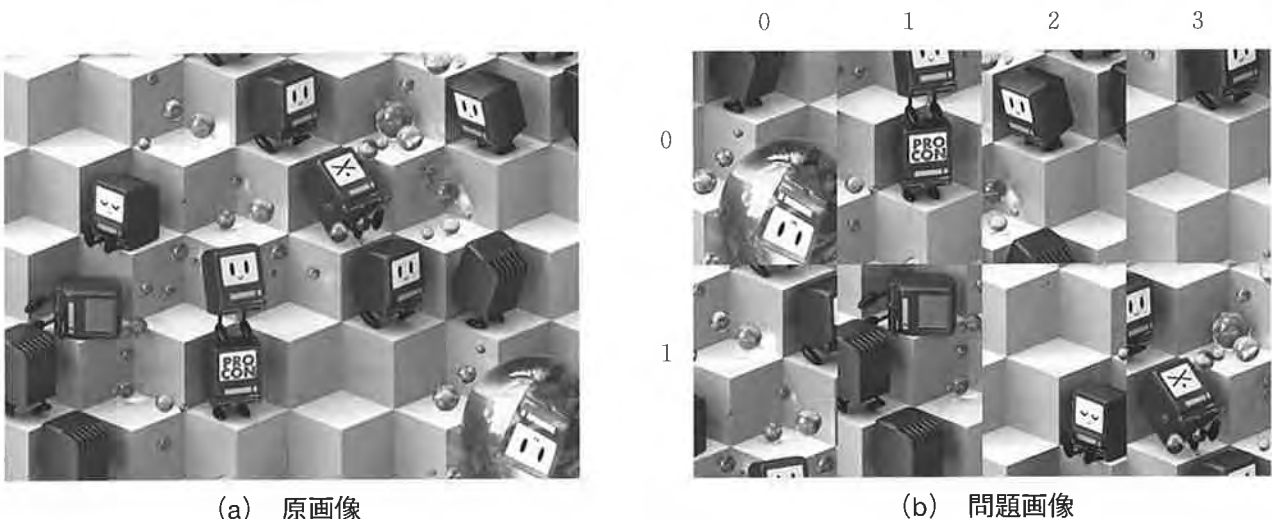


図1. 幅640ピクセル、高さ480ピクセルの(a)原画像を、4×2に分割して作成した(b)問題画像

課題部門本選参加作品

■「防災・減災対策と復興支援」

発表順	タイトル	高専・大学名	指導教員	参加学生
1	光陰如箭 —A ray of hope—	弓削商船	長尾 和彦	宇崎 裕太, 瀬尾 敦生, 肥田 琢弥, 山本 愛奈
2	潮エネ!“安” —スゴいぞ潮流発電・3つの安で世界を変える—	大島商船	北風 裕教	重本 昌也, 村上 秀隆, 柿本 福実, 中野 裕次, BATNASAN BAASANCHULUUN
3	Change —希望が育つ願いの樹—	松 江	和田守美穂	今若 悠樹, 吉田 悠花, 岩崎 未来, 古志 優菜, 土井 一磨
4	Go! UP —助かる命を守りたい—	沖 縄	正木 忠勝	大西 諒, 山城 響, 野崎清太郎, 賀数 志乃, 又吉 純次
5	Man-Hold —町を守る近未来マンホール—	明 石	新井 イスマイル	森 篤史, 森 恵, 西原 大貴, 萩野 秀祐
6	DTN通信を用いた災害時の安否及び避難所情報収集システム	富山(射水)	山口 晃史	館山 北斗, 堺 瑞起, 前田 広夢, 柴田 大希, 藤野 裕時
7	BlockSnow —ホワイトアウト回避アプリ—	旭 川	森川 一	板坂 優人, 山口 凌, 佐藤 広基
8	マップコXD —マップでつなく地域コミュニケーション—	一 関	小保方幸次	鬼柳 元樹, 佐藤 健太, 下田 将之, 千葉 大輝, 及川 遥
9	PoP —災害時被災者情報管理システム—	米 子	河野 清尊	大野 貴昭, 嶋本 智矩, 中村紗也佳, 西尾 泰希, 原田 尚哉
10	Groupier —集まりを つなぐ—to—	津 山	宮下 卓也	中村 悠生, 須和田与春, 萩原 涼介, 綾部 敬祐
11	BBB —非常時の非インターネット環境での情報交換システム—	仙台(広瀬)	穂坂 紀子	高橋 滉一, 及川 達希, 東海 佳祐, 後藤 隆太, 齋藤 翼
12	Relief Supplies Relay	鈴 鹿	浦尾 彰	伊藤 貴哉, 紀平 将史, 早川 栄作, 道上 将志, 櫻井 真子
13	津波避難エキスパート	金 沢	伊藤 周	澤田 友樹, 北 直樹, 中嶋 悠也, 中山 義崇, 長井 健太
14	i-BadgeOVERしまNET —子供見守り防災システム—	弓削商船	田房 友典	檜垣 俊希, 岡野さくら, 亀島加奈恵, 福羅亜利沙, 村上麻矢加
15	Sneaker@mail —心配なあの人に届ける便り—	沖 縄	鈴木 大作	近藤 史麻, 久保田明成, 下地 春希, 川満 大輝, 貝盛 陽平
16	キズナ時計 —万が一に備えた腕時計—	新 居 浜	先山 卓朗	山之口智也, 末光史有土, 宇佐美健大, 田中 宏政
17	MAP FIXER —災害時に役立つ町内道路地図アプリ—	熊本(八代)	小島 俊輔	木村 匠, 関原 至音, 湯舟 武龍, 岩下 将大, 本山 和輝
18	つながっタワー —津波避難タワー間を結ぶ安心防災システム—	高 知	今井 一雅	佐々木 渉, 島内 良章, 南 光成, 森國 健吾
19	人(ヒート)マップ —みんなで作る情報マップ—	鳥羽商船	江崎 修央	伊藤由祐紀, 竹口 優里, 谷口 昂汰, 萩原 海仁, 濱口 堅太
20	WT —つなく・つながる パケットを速くまで届け隊!!!—	福 井	斉藤 徹	野村 弘樹, 中後 和希, 野村 信吾
21	SKcAP —空から支える復興支援—	宇 部	田辺 誠	金子 昂稔, 西村 優佑, 伊藤 賢也, 田代 翔也, 久継 宏樹
22	(未定)	ペトロナス 工科大学	Faizal Ahmad Fadzil	Muhammad Izzuddin Eshak, Mohd Syazwan Zakaria

・提出された原稿をそのまま印刷しています。

1

光陰如箭

—A ray of hope—

弓削商船

宇崎 裕太(4年) 瀬尾 敦生(4年)
 肥田 琢弥(4年) 山本 愛奈(4年)
 長尾 和彦(教員)

1. はじめに

災害発生から 72 時間(超急性期)が経過すると生存率が急激に低下するため、すみやかに救助を求めることが必要です。しかし様々なインフラが壊滅し、利用できない期間であるため、インターネットはもちろん電話やメールで助けを呼ぶことはできません。我々は情報インフラ途絶時の通信手段として光モールス通信システムを提案します。

2. 概要

我々は「身近なもので遠くに情報を伝える」ツールとしてスマートフォンによる光モールス通信システムを提案します。スマートフォンは、多くのユーザが常時携帯するコンピュータ端末であり、緊急時でも活用できます。光モールス信号は遠距離通信(~数 10Km)ができ、機器を持たない自衛隊等でも解読可能です。

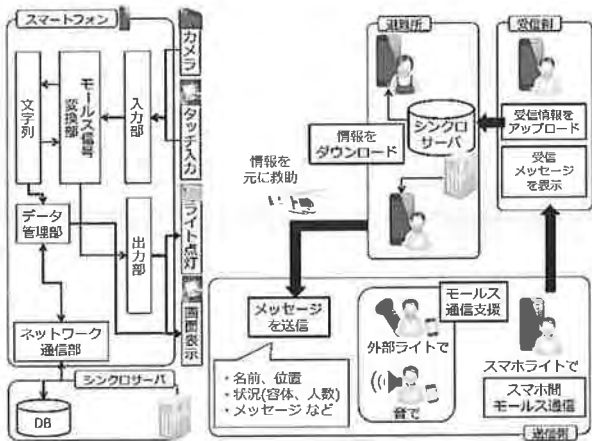


図 1. システム構成図

3. 提供する機能

3.1 モールス送信機能

定型文および任意文字列をモールス信号に変換し、内蔵ライトの点滅として発信します。点滅のタイミングは画面に表示されるので、そのタイミングに合わせて車のヘッドライトなど、より強力なライトを使用することも可能です。

3.2 モールス受信機能

スマートフォンのカメラで光の点滅をとらえると、

画像からモールス信号と思われるポイントを検出し、モールス信号の解読を開始します。解読されたモールス信号は文字列へ変換しモニタ表示されます。送信者によってテンポが異なる場合でも正しく変換できます。

3.3 定点観測モード

固定設置した端末では、複数のモールス信号を同時に受信することができ、高所に設置された定点カメラやヘリコプターなどからの検索でも効果を発揮します。



図 2. 定点観測モード(PC)

3.4 モールス学習機能

モールス信号に慣れてもらい、防災意識を高めるため、学習機能を作成しました。モールス信号や仕組みなどを学習し、小テストを行うことができます。

3.5 シンクロサーバ機能

本システムと連携できる情報共有サーバを開発しました。本システムで通信した内容を複数の端末でシンクロでき、避難所単位での円滑な情報共有をサポートします。

4. まとめ

自然災害の多い日本では、今後も多くの災害が予測されています。本システムは通信インフラ復旧までの期間に利用できる伝達システムです。災害時の被害減少に本システムが活用できれば幸いです。

5. 参考文献

- ・澤田努(2014)「南海トラフ大規模災害に備えて仮想化技術による地域間連携医療情報ネットワーク」
- ・柴田義孝(2014)「災害時に有効な情報ネットワーク」

2

潮エネ！ “安”

—スゴいぞ潮流発電・3つの安で世界を変える—

大島商船

重本 昌也(専1年) 村上 秀隆(専1年)
 柿本 福実(4年) 中野 裕次(4年)
 BATNASAN BAASANCHULUUN(4年)
 北風 裕教(教員)

1. はじめに

福島第一原発事故が発生して以来、風力発電や太陽光発電など自然エネルギーを用いた発電方法の需要が高まっています。しかし、これらの方法は天候に左右されやすい、環境に悪いなどのデメリットがあり、未だに日本の発電量の約1割にとどまっています。

そこで私たちは、自然エネルギーを用いて、有害物質を排出せず、安定して発電量がまかなうことができる潮流発電に着目しました。「潮エネ！ “安”」では、3つの“安”（安心、安全、安定）を達成することにより、効率的に潮流発電を行えるシステムを提案します。

2. 潮流発電装置

2.1 潮流発電の仕組み

潮流発電は、水車（翼）を海中に沈め、潮の流れを用いて回転させることによってエネルギーを得る発電方式です。それぞれの地点における潮が満ち引きする時間帯は一定であるので、潮の流れる速さを予測することが可能になり、発電量が予測しやすいというメリットがあります。

2.2 多段式翼を用いた潮流発電装置

本システムで用いる発電装置には、多段式翼を採用しました。多段式翼は火力発電などで使われる蒸気タービンを参考にしました。流速が遅い段階からでも回転し始めることができます。これにより、従来の潮流発電よりも多く発電が可能です。

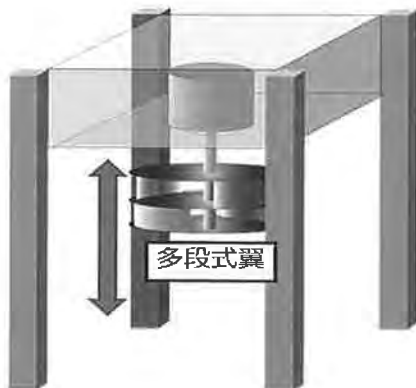


図1 潮流発電装置のイメージ図

2.3 自動引き上げ機能

本装置では、翼を自動的に海中から上昇させる機能を備えています（図1）。これをプログラムで制御することで、流速が小さい時間帯に翼を上昇させることによって貝などの海洋生物の付着を防ぐことができます。また、過負荷による事故など緊急時にも対応できます。

3. 管理システム

3.1 発電装置の管理

管理システムでは、流速情報と発電情報を照らし合わせ、正しく動作しているか確認が行えます。

3.2 情報の管理

発電装置から USB-I2C.0 を利用して、Web サーバ（PHP）や管理システムとのデータのやりとりが可能になっています。管理者はサーバに蓄積された情報をもとに、電力会社へ売電を行うことができます。

また、発電情報はグラフやアニメーションなどでわかりやすく表示されるので、電気に詳しくない方も簡単に発電状況が理解できます。

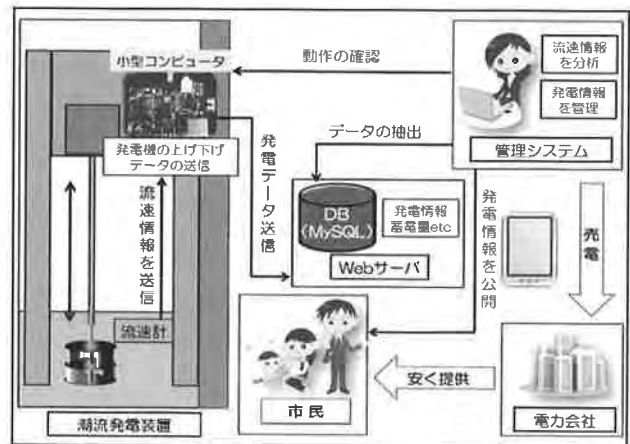


図2 システムの構成図

4. おわりに

「潮エネ！ “安”」によって、日本の電力を少しでも多く潮流発電でまかなえるようにしていきたいです。そしてみなさんの暮らしに3つの“安”が訪れることを願います。

3

Change

—希望が育つ願いの樹—

松江

今若 悠樹(4年) 吉田 悠花(5年)
 岩崎 未来(3年) 古志 優菜(3年)
 土井 一磨(1年) 和田守美穂(教員)

1. はじめに

2011年、東日本大震災が起きた際に多くの人が募金を行いました。しかし、募金の使われ方は不透明で、実際には震災の復興とは無関係なことに使われているという問題がありました。いわば、集まった募金が募金団体を通じて被災者に届けられるという「一方通行の関係」でした。

そこで私たちは、被災者の声を募金者に届け、募金の流れを可視化することで「双方向の関係」を実現するシステム「Change」を考案しました。

2. システムの概要

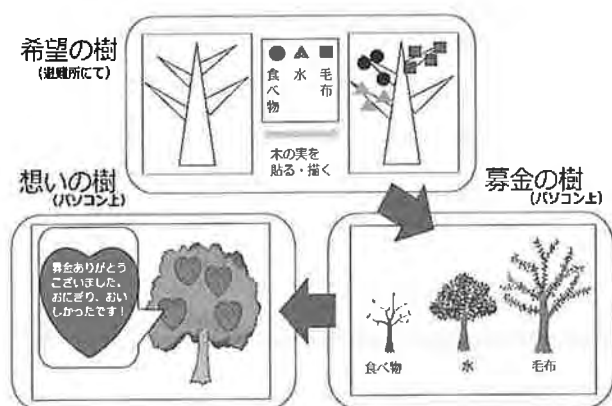


図1. 主な機能

本システムは、災害発生直後の混乱した時期ではなく、ある程度時間が経過してから使用することを想定しています。募金管理団体は、本システムを使用することで、募金活動の活性化と、被災者・募金者間の双方向の関係を築く手助けをします(図1)。

本システムはWebサービスとして提供され、「希望の樹」、「募金の樹」、「想いの樹」の3つの機能があります。「希望の樹」では、避難場所で必要とされている物資の集計を行い、「募金の樹」ではWebページ上で募金を行うことができ、募金の使われ方を知ることができます。また、「想いの樹」では募金者や被災者がメッセージを送ることができます。

3. システムの機能

3.1 希望の樹

避難所に貼られた「希望の樹」(模造紙等の大きな紙)に、被災者が必要な物資を表す「木の实」を描きます。避難所を管理している人などがそれをカメラで撮影し、本システムにアップロードします。本システムでは、「希望の樹」の画像が解析され、木の实の種類や個数を認識することで、各避難所で必要とされている物資を自動集計します。

3.2 募金の樹

「希望の樹」や被災地に実際に届けられた支援物資の情報(種類、量など)を元に、物資ごとに「募金の樹」がWeb上に作成されます。樹の大きさは必要に応じて異なり、物資の供給達成度が葉の量で表示されます。募金者は、「募金の樹」の状態から募金の使われ方を選択し、募金を行います。

3.3 想いの樹

募金者は、自身の募金が資金の一部となり購入された支援物資を受け取った被災者からのメッセージを、被災者は募金者からの応援メッセージを受け取ることができます。これにより、募金者と被災者は互いの想いを知ることができます。

4. おわりに

本システムで、募金者と被災者との「双方向の関係」を築くことにより、募金活動の活性化や被災地の復興加速の一助となれば幸いです。

4

Go! UP

—助かる命を守りたい—

沖 縄

大西 諒(3年) 山城 響(2年)
野崎清太郎(2年) 賀敷 志乃(2年)
又吉 純次(1年) 正木 忠勝(教員)

1. はじめに

2011年3月11日、甚大な被害をもたらした東日本大震災で多くの人が津波で命を落としました。震災後の調査から、逃げ遅れによる溺死が死因の大部分を占めたことがわかりました。さらに、避難に関してのシミュレーション結果より、津波から避難しようとしても渋滞で動けなくなる、避難所が遠くて間に合わないなどの問題が浮かび上がりました。そこで、私たちは避難所までたどりつくことができない人が最後に自分の命を守るために特化したアプリ「GO!UP」を提案します。

2. システム概要

「GO!UP」は津波から逃げ遅れた人が自分の命を守るための支援をするスマートフォンアプリです。このアプリは気象庁から発表される緊急地震速報・津波予報、自治体のハザードマップや避難所情報などのオープンデータ、現在位置と標高を知るためのGPS、準天頂衛星「みちびき」を活用します。

3. 機能

3.1 緊急避難機能

避難所まで逃げるができない状況で、気象庁からの津波予測、ハザードマップや標高などのオープンデータ、GPSを用いて測定した現在地の標高を用いて、今いる場所からあと何メートル高いところに避難すればよいかを指示する機能です。このとき、画面表示だけではなく避難のため両手がふさがっていることを想定して通知音や音声によっても誘導します。



← どの高さまで上がればよいかを表示

← 津波がくるまでの時間

3.2 避難所誘導機能

津波が引いたあと、安全な避難所まで誘導する機能です。しかし、津波によって地図が役に立たない状況

になっていると想定されるため、地図を使わずに避難所までの「方向と距離」だけを表示して誘導をします。



← 距離と方向だけを表示して避難所までを誘導

4. 実現方法

4.1 みちびきと DGPS による高精度高度測定

GPSのみでの高度測定では数十mの測定誤差があるため、準天頂衛星「みちびき」を利用してGPS補完信号の情報による補正と、A-GPS方式の利用によって国土地理院の電子基準点を利用した携帯電話での位置補正を行うことによって高精度高度測定を実現します。

4.2 オープンデータの利用

気象庁から集めた緊急地震速報や津波警報・高さ予想と、各自治体が公開している避難所の位置情報などのオープンデータを各機能へ利用します。また、オープンデータは現段階では全てがそろっていないので本作品の機能実現に不足しているデータは従来のアナログデータから作成して対応します。

4.3 災害による通信回線障害対策

災害時には通信障害が発生する事が想定されます。そこで、通信障害が発生する前の緊急地震速報を受信した時点で、避難に必要な各オープンデータを集中的に収集することで対応します。この機能により、通信障害が実際に起きたときはすでにデータの収集が終わり、アプリは正常に動作することを保証します。

5. おわりに

「助かる命を守りたい」私たちは、避難所に行くまで間に合わない、だけど助かる可能性は十分にあるという人たちの命を守るお手伝いをするためにこのアプリを開発しました。しかし、実用化のためにはオープンデータが重要です。「GO!UP」がオープンデータの推進に役立つことを期待します。

5

Man-Hold

—町を守る近未来マンホール—

明石

森 篤史(3年) 森 恵(3年)
西原 大貴(3年) 萩野 秀祐(3年)
新井イスマイル(教員)

1. はじめに

近年、地震・土砂災害などの災害が増えている。それらによる被害の多くは、早期避難していれば軽減できたものも多い。そこで、夜間や不慣れた土地でも、子どもから高齢者まで誰もが安全かつ速やかに避難できるシステム「Man-Hold」を提案する。

Man-Holdは既存のマンホールを活かして、避難所までの安全ルートを表示するだけでなく、スマートフォン等に情報を送る機能も搭載し、安全で素早い避難を支援する。

2. システム概要

Man-Hold の概要を図 1 に示す。避難所までの道をマンホールのふたに設置した LED の矢印で示す。通れない道等については、NFC 通信を用いてスマートフォン等からマンホールに通知しマンホール間通信による伝達、または口頭での伝達により、避難所の PC に情報を集め、それらをもとに安全なルートを再計算して、各マンホールに情報を伝達し、それぞれの矢印の方向を変更する。さらに、簡易な災害情報等も避難所の PC から各マンホールに送信し、NFC 通信を用いてスマートフォン等で情報を入手可能にする。輻輳時にも使用できるようにマンホール間通信には地域ごとに独自のマルチホップ・アドホックネットワークを構築する。マンホールの通信・LED の制御は Raspberry Pi を用いる。

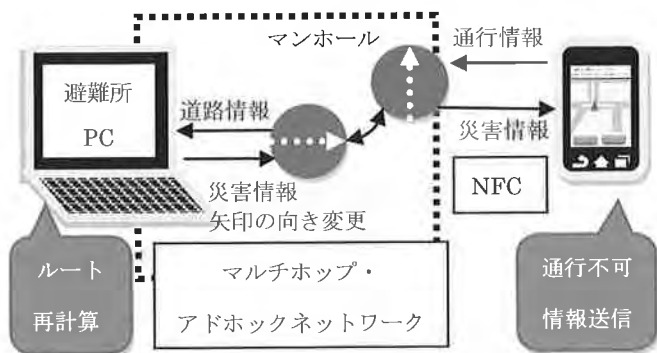


図 1 Man-Hold システム概要図

3. 機能

3.1 避難誘導

避難指示が出ると、避難所までの安全なルートをマンホールの矢印が光って指し示すので、図 2 のように矢印の向きに従って避難するだけでよい。雨天時や夜間でも、身長に関係なく見やすく、直感的でわかりやすいので、子供から高齢者まで、旅行者やビジネスマンの人でも迷わず安全に避難することができる。

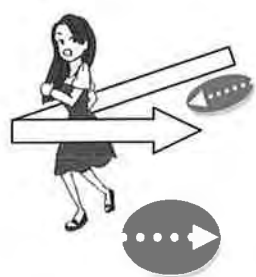


図 2 避難誘導



図 3 災害情報発信

3.2 災害情報発信

スマートフォン等でマンホールにタッチするだけで、津波警戒情報や避難情報、地図などの情報をインターネットに接続せずに入手することができる。(図 3)

3.3 日常的使用

照度・紫外線・超音波距離センサ・加速度センサなどさまざまなセンサを Man-Hold に搭載することにより、気象情報・交通量・地震・夜道の明るさなどの詳細なデータを得て、そのデータを日常生活に役立てることで、Man-Hold の平時の有用性・認知度を高める。

4. まとめ

Man-Hold の避難指示・情報伝達などの機能により避難のしやすさ・安全性を向上させ、逃げ遅れによる被害をなくす。子ども、高齢者、旅行者等、誰もがどこに行っても安心して過ごせる環境を Man-Hold で実現していきたい。

6

DTN通信を用いた災害時の安否及び避難所情報収集システム

富山(射水)

館山 北斗(5年) 堺 瑞起(4年)
前田 広夢(3年) 柴田 大希(2年)
藤野 裕時(1年) 山口 晃史(教員)

1. はじめに

先の東日本大震災では、基地局の破壊により7日間携帯端末の使用ができませんでした[1]。災害時における携帯端末は他の安否確認だけでなく自身の状態を伝える唯一の手段です。そこで我々は基地局が復旧までの間、端末が有するWi-Fi機能を用い、Ad hoc通信を連鎖的に起こすことで、小規模のネットワークを構築できるようにしました。

2. システム概要

2.1 システム構成

本システムの構成図を図1に示します。

①携帯端末

携帯端末間ではMANET通信、DTN通信により、一連の情報の表示と送受信を行います。

②データベースサーバー

避難所で電源確保が可能な時は、運営者が入力した情報(支援、安否、配給、伝言)を直接取得できます。

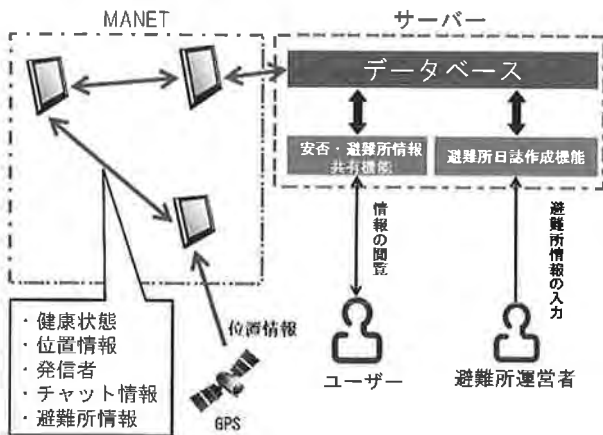


図1 システムの構成図

2.2 携帯端末間通信について

①MANET(Mobile Ad hoc NETWORKing)通信

本システムでは、携帯端末同士をアクセスポイントを必要としないAd hoc通信で接続します。

②DTN(Delay Tolerant Networking)通信

災害時、端末内部に蓄積された各種データは、付近にいる端末へ自動的に送信されます。受信側は情報の

ハッシュ値により既読判断を行い、新規の情報の取得と別の端末への送信を次々と行います。(図2)[2]

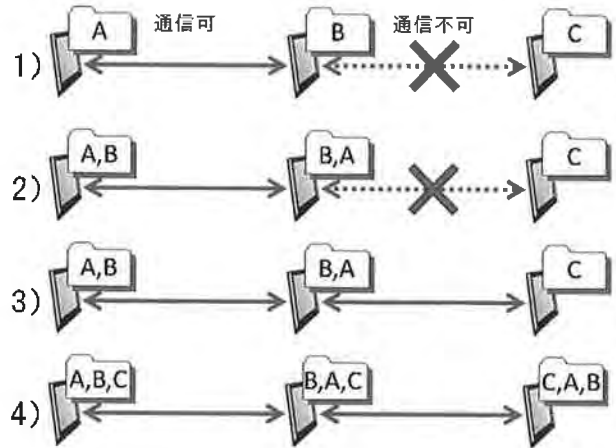


図2 DTN通信の概要図

2.3 システムの機能

①安否情報、避難所情報の共有

これまで紙ベースで行われてきた安否情報開示や避難所での伝言を電子データ化します。

②避難所への誘導

GPSデータを用い、現在地から一番近い避難所まで誘導します。

③避難所日誌作成

避難所運営者が記録した日誌は今後の避難所運営に必要な大切な情報です。

3. 現状の課題

実地試験の結果、携帯端末のバッテリーが24時間しか使用できませんでした。そのため今後は、ソーラーパネルを装着した携帯端末で実験を行います。

4. 参考文献

[1] 東日本大震災における情報通信の状況

<http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h23/pdf/n0010000.pdf>

[2] Disruption Tolerant Networking for Space Operations (DTN) - 07.29.14

http://www.nasa.gov/mission_pages/station/research/experiments/730.html

7

BlockSnow

—ホワイトアウト回避アプリ—

旭川

板坂 優人(3年) 山口 凌(2年)
佐藤 広基(3年) 森川 一(教員)

1. はじめに

震災や台風などの災害が注視される中、あまり知られていない、ホワイトアウト。降雪地方、特に北海道では多く見られ、雪や雲などで視界が悪くなり、それによった事故などで、毎年遭難者や死者を出している恐ろしい災害です。

私たちが開発した BlockSnow は、そんな状況に陥った時に、被害を抑えるようにできるアプリケーションです。



図 1. 実際のホワイトアウト

2. システム概要

まず、何事も予防が大切です。BlockSnow はリアルタイムで天気情報を取得し、ホワイトアウトの警報が出ている地域に近づくと、利用者にわかりやすい形で警告をし、安全地帯に移動することを勧めます。その際には、ナビゲーションシステムを使うことで、できるだけ安全に利用者を避難させることができます。

あらかじめ利用者が住んでいる地域や、緊急時の連絡先を登録しておくことにより、すばやく天気情報を取得することや、利用者に危険があったときに、その緊急連絡先に連絡を送ることができます。また、居住地の情報は後で変更可能なので、引っ越しや旅行のときにも継続してサービスを受けることができます。

ホワイトアウトに関してだけでなく、純粋なマップ

機能や天気情報取得機能が付随しています。また、Google Playへの登録をすることで、多くの方が利用できるようにしようと思っています。その他にも、見やすく使いやすいユーザーインターフェースを提供できるようにしました。

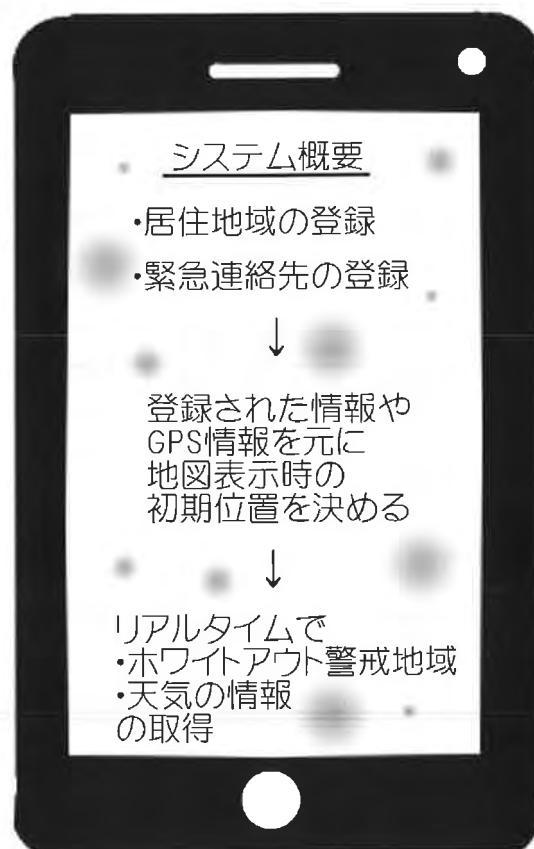


図 2. システム概要

3. おわりに

私たちの BlockSnow は、今までにあった信頼できる機能を持ちながら、より高い水準で、使いやすさ、手軽さ、見やすさの3つのユーザーインターフェースとしての理想を実現しています。

また、ホワイトアウトによる危険から利用者を守ることに他に、日常生活においても便利に活用できる機能を備えており、幅広い範囲でユーザーをサポートする優れたアプリケーションとなっています。

8

マップコXD

—マップでつなぐ地域コミュニケーション—

— 関

 鬼柳 元樹(5年) 佐藤 健太(5年)
 下田 将之(5年) 千葉 大輝(5年)
 及川 遥(5年) 小保方幸次(教員)

1. はじめに

東日本大震災から三年半が経ち、復興が進められています。しかし、被災地での集団移転について、住民間での話し合いの不足などにより計画が進んでいない地域が多く残っています。

そこで、復興に関わる人々のコミュニケーションと街づくりに関する意見交換をサポートする「マップコXD」を提案します。

「マップコXD」は、PC・スマートフォン・タブレットで利用できるwebアプリで、地図上で行われるコミュニケーションツールです。地図上にコメントを埋め込むことや、用意されている掲示板に意見を書き込むことで、他のユーザーとコミュニケーションを取ることができます。また、コメントによって街の建物に関するアンケートが実施され、街づくりへの意見交換の場が設けられます。

2. 機能説明

2.1 地図について

地図データはOpenStreetMapを利用しています。地図では、街にある建物を知ることが出来ます。自分の家や店を登録することができ、それらに自己紹介やコメントなどを載せられます。



図1 地図上の建物

2.2 コメント

地図上の建物のない場所をクリック・タッチすることでコメントを埋め込みます。コメントを埋め込むことで、街の良いところや改善意見を発信することが出来ます。更に、改善意見が多くなることで、街の改善

に関するアンケートを行います。アンケートによってユーザーの街づくりへの積極的な参加をサポートし、意見を出しやすくします。

2.3 掲示板

掲示板では、メッセージの書き込みやアンケートへの回答を行えます。また、地域の店の情報を確認したり、自分の店の情報を発信したりすることが出来ます。これにより、住民間のコミュニケーションを手助けし、地域の活性化にもつながります。

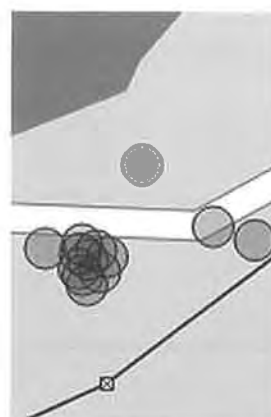


図2 コメント



図3 掲示板

3. システム構成

「マップコXD」はデータベースサーバーと、各利用者の端末で構成されています。



図4 システム構成

4. まとめ

「マップコXD」を用いることで、既存のコミュニケーションの手段にはない、地域内での繋がりを意識することが出来ます。また、街の情報を発信することで、商業面での復興にも発展すると考えています。

1. はじめに

災害後の避難場所（体育館、公民館、仮設住宅等）における問題点として以下のことが挙げられます。

(ア) 被災者にとって

知らない支援者、団体の人が情報収集のために何度も訪問してくることによる精神的負担が大きい。

(イ) 支援者にとって

- 被災者のニーズ（要望）を知るために、被災者のところまで情報収集に行かなければならない。
- 被災者が生活の場を変えることにより、避難している場所・人数・人物像などが把握しづらい。
- 支援者・団体の無秩序な活動展開により、支援活動が重複したり、必要な人に必要な支援が行き渡らない。

そこで、私たちは、タブレット端末を用いて被災者が要望を簡単に入力でき、その要望を自動的に集計して支援者が確認できるようにするとともに、AR（拡張現実）機能を用いて要望を可視化することで、被災者のニーズを正確かつ迅速に把握することのできる災害時被災者情報管理システム『PoP(Personal orders Popping)』を開発しました。

2. 本システムの概要

2.1 システム構成

本システムは、図1に示すように、被災者用と支援者用のタブレット端末およびサーバマシンから構成されます。

①被災者は、タブレット端末から要望を入力するとともに、要望に対する支援者の対応状況を確認することができます。 【被災者サポート機能】

②被災者の要望はサーバで集計され、支援者用タブレット端末に転送されるとともに、支援者の対応状況を被災者用タブレット端末に転送します。 【集計・転送機能】

③支援者は集計された要望を確認するとともに、要望への対応を返信します。また、避難場所でタブレット端末をかざすことにより、AR機能を使って要望を可視化して確認することができます。 【支援者サポート機能】



図1 システム構成

2.2 システムの機能

(1) 被災者サポート機能

「要望の入力と対応状況の確認」を実現します。

被災者は、アイコンを選び送信するという簡単な操作で要望を支援者に届けます。(図2)

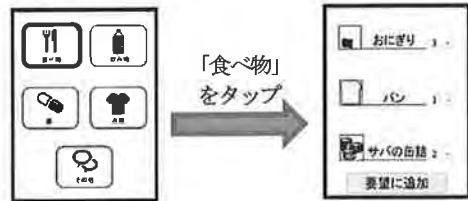


図2 要望の入力画面

(2) 集計・転送機能

「要望の集計と要望・対応状況の転送」を実現します。

被災者から送られた要望を、避難場所別・被災者別・要望の種類別等で集計します。将来的には、XMLやRDFを使ってオープンデータ化することにより、連携機関の適切な物資配給や人員配置につなげたいと考えています。

(3) 支援者サポート機能

「要望の確認・可視化と対応の返信」を実現します。

AR機能を用いて要望を可視化することにより、被災者のニーズを的確に把握することができます。(図3)



図3 AR機能を使った要望の可視化

(4) 登録機能

被災者の世帯ごとにタブレット端末を配布し、配布時に必要な情報を入力してもらい、サーバに登録します。

3. 実現にあたっての課題と解決策

【課題1】被災者がPoPを使いこなすことができるか？

タブレット端末を事前に各家庭に配布し、普段から生活の中で、自治体への要望提出等に使用してもらいます。

【課題2】要望の可視化：タブレット端末の位置の特定？

無線LANアクセスポイントや端末のGPS機能を用いて、被災者用タブレット端末の位置情報を取得します。

【課題3】タブレット端末とサーバマシン間の通信手段？

http通信を用います。処理結果はJSON(XML)で返し、サーバプログラムはLinux+PHP+MYSQLで開発します。

4. まとめ

PoPは、授業での質問、会議での意見、自治体への要望提出等、学校・会社・地域での応用発展が考えられます。

10

Grouper

—集まりをつなぐに—

津山

中村 悠生(4年) 須和田与春(3年)
萩原 涼介(2年) 綾部 敬祐(1年)
宮下 卓也(教員)

1. はじめに

東日本大震災後、人々はより一層「つながり」を重要視しています。そこで私たちは、非常時でも災害時でも「人とのつながり」をサポートすることができるツールとして「Grouper」を作成しました。

「Grouper」は、平常時はコミュニケーションツールとして、災害時は情報共有・伝達ツールとして「人とのつながり」をサポートします。

2. システム概要

「Grouper」は短期間・少人数でグループを作成し、その中でさまざまなツールを利用することができるシステムです。グループ利用に特化した機能を搭載しています。

Android スマートフォン上で動作し、インターネット環境に接続している環境で動作します。移動式サーバを利用している場合は、そこに接続することで Grouper を利用することができます。

3. 基本機能 (図1 参照)

3.1 トーク機能

グループ内で自由に文字を使ってトークをすることができます。位置情報の共有や画像送信も可能です。

3.2 アラーム共有機能

グループのメンバーの端末のアラームを一斉に鳴らすことができます。また、アラームに対し選択肢を設定し、メンバーの選択状況を確認することができます。

3.3 災害時専用機能

グループ作成時に「災害用」を選択すると、「掲示板機能」「スケジュール機能」を利用することができます。

掲示板機能は、行政手続きに関する情報や支援の情報をグループ内で共有することができる機能です。

スケジュール機能は、複数人で行動する際、予定時刻になったら通知を送信することができる機能です。

3.4 移動式サーバ

インターネット回線がない環境でも Raspberry Pi を利用した移動式サーバがあれば、Grouper をどんな時でも利用することができます。



トーク機能
手軽にいつでも話せる



アラーム機能
全員に一斉通知
メンバーの状況も確認



災害時専用機能
情報伝達を素早く正確に
掲示板機能・スケジュール機能

図1 Grouper の主な機能

4. 想定する利用例

4.1. 災害時

- ・ボランティアメンバー間の情報交換
- ・避難所での情報交換
- ・家族や友だちの安全確認(アラーム機能の応用利用)

4.2. 平常時

- ・パケットツアーなどでの情報伝達やアラーム共有
- ・迷子になっている人へ集合場所の送信
- ・寂しい時の誰かとの会話

5. おわりに

Grouper は「災害時」のみならず、「平常時」でも便利に使うことができるように設計されています。

Grouper を使うことで、ふだんから「つながり」を意識するきっかけになればと思っています。

11

BBB

—非常時の非インターネット環境での情報交換システム—

仙台(広瀬)

高橋 滉一(2年) 及川 達希(2年)
 東海 佳祐(2年) 後藤 隆太(2年)
 齋藤 翼(2年) 穂坂 紀子(教員)

1. はじめに

インターネット環境というのは、短時間で広範囲の情報を収集できることから、災害時、被災地の人々が家族や友人の安否情報を素早く確認するうえで重要なものである。しかし、東日本大震災が起きた時、インターネット環境を利用しようと試みた人は多かったものの実際には使用できなかった。大規模な災害時には、インターネット環境を使えない非インターネット環境になる可能性が高い。

私たちは、そのような非インターネット環境に置かれた際でも広範囲の情報交換を可能にしたいと考え、本システム「BBB」を開発した。

2. システム概要

2.1 システム構成

「BBB」は Bluetooth 通信機能を利用した非インターネット環境で情報交換ができるようになるシステムである。spot と呼ぶサーバと携帯端末で本システムが成り立つ。使用可能なデバイスは android 搭載端末である。一般ユーザー用と管理者用の 2 種類の端末があり、それぞれ専用のアプリケーションを用意する。ユーザーは web サイトからアプリケーションを端末にダウンロードする。管理者には spot を配布すると同時に専用アプリケーションを直接配布という形にする。次の節で管理者端末について説明する。

2.2 管理者端末

発信元の保証された公共情報を流すために管理者端末を用意する。管理者端末のユーザー端末との違いは「公共情報」の категорияに情報を投稿できること、ユーザーが投稿した情報が不適切と判断される場合に情報を削除できることである。この管理者端末は市役所等の公共機関からの情報提供に利用することを想定している。管理者端末から投稿された情報とユーザー端末から投稿された情報はカテゴリーを分け、さらに端末の表示画面で色分けしてどちらが投稿した情報か区別する。

3. システムの利用

3.1 情報の閲覧と投稿

図 1(a)のような状況で、ユーザーは spot に対して投稿という形で情報を提供し、spot に保存された情報を掲示板形式で閲覧、取得することができる。また、管理者は公共情報を提供する。図 2 に示すように端末の表示画面を目的別に分けることで一目で使い方が分かるようにした。

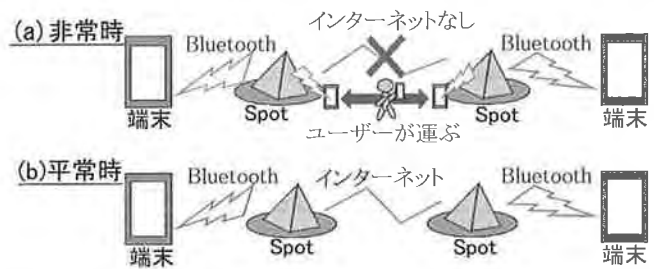


図1 非常時、平常時の通信環境比較

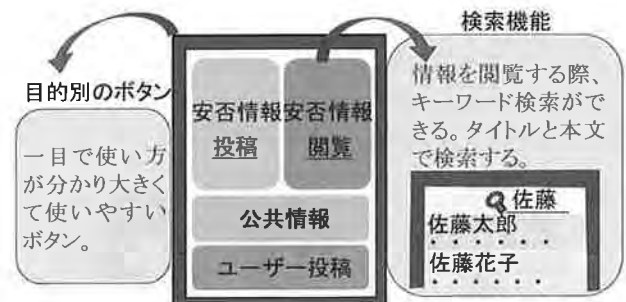


図2 「BBB」のホーム画面イメージ図

提供された多くの情報から素早く特定の情報を見つけるために「BBB」にキーワード検索機能を持たせた(図 2)。これまで災害時の避難所での情報提供は、紙媒体による張り出しや掲示で、視認性が高い一方、必要な情報を見つけるのに時間がかかってしまっていた。検索機能によって、必要な情報をすぐに見つけることができ、この問題を改善した。

3.2 情報の持ち運び

本システムで、ユーザーは spot に情報を投稿したり、spot の情報を閲覧するだけでなく、取得した情報を他の spot に持ち運び、運んだ情報をその spot に投稿することができる。この仕組みによって、図 1(a)に示すような非インターネット環境でも広範囲に情報を拡散することが可能になる。また、避難所間で情報をやりとりしたり、被災地外へ情報を運んだりすることで、避難所間の連携や被災地外からの速やかな支援に繋がる。

3.3 平常時におけるシステム使用

図 1(b)に示すように、平常時には spot 間でのインターネット通信利用も想定している。こうすることで本システムの利用範囲が広がり、平常時にもこのシステムを有効に使うことができる。使用例として、地域の掲示板や回覧板が挙げられる。日常的に本システムに触れることで、非常時に使い方で戸惑うことなくシステムが利用できるようにする。

12 Relief Supplies Relay

鈴 鹿

伊藤 貴哉(4年) 紀平 将史(4年)
 早川 栄作(4年) 道上 将志(4年)
 櫻井 真子(3年) 浦尾 彰(教員)

1. はじめに

東日本大震災発生後、避難所にて物資が足りないという報道が頻繁に行われていたことは記憶に新しいのではないかと思います。しかし、詳しく調べてみると、物資は確かに避難所へは行き届いていませんでしたが、被災地へは十分届いていました。

原因は被災地で物資を受け入れる集積所がありました。大量の物資が一度に届いたため、集積所での仕分け作業が追いついていなかったのです。

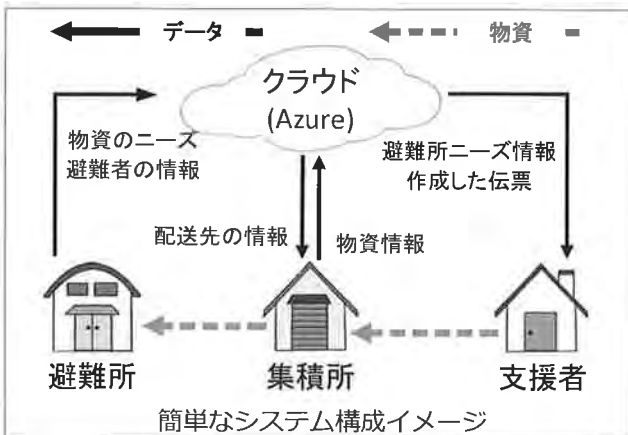
さらに、不要な物資が大量に集積所に届く、避難所で必要とされている物が把握できないといったことが起きたことも、集積所の混乱の原因となりました。



2. システム概要

本システムは避難所にてどのような物資が必要とされているのか、どれだけの方が避難しているかを伝達する避難所端末、集積所にて物資を振り分ける集積所端末、及びこれらを管理する Azure サーバによって構成されます。また、被災地外より物資を送る支援者はサーバが提供する専用の WEB ページにて送る物資、個数を入力し出力された伝票を印刷し送る物資の箱に貼り付けてもらいます。

従来、集積所では送られてきた物資の一つ一つを、箱を開けて中身を確認していました。本システムでは、物資を送る支援者側で、物資の情報を入力することで集積所の仕分け作業の負担を大幅に軽減します。



3. ReliefSuppliesRelay が提供する主な機能

避難所向け ～避難所端末～

避難所状況 伝達機能	避難時にマイナンバーカード(※1)をかざすだけで避難者数を性別、年代とともにサーバへ伝達※2
物資要求 機能	年代、性別を元に予測可能な水や食料などはサーバ内で自動的に要求 一方、乾電池など需要が不定期に発生する物資は付属カタログから物資を選び端末で入力することで要求可能
オフライン 機能	サーバと通信できない場合、付属 SD カードを、物資受け取り時に集積所の方に渡すことで物資要求を伝えることが可能

※1 マイナンバーカードは 2016 年より国から無償で提供され、各種手続きや災害時の活用が予定されています。

※2 マイナンバーカードをお持ちでない場合は生年月日などを端末に手動入力することで対応可能です。

集積所向け ～集積所端末と専用アプリ～

物資 振り分け機能	物資に貼り付けられている伝票を読み込むだけで自動的に配送先を表示 従来集積所で行われていた煩雑な仕分け作業などが大幅に削減され、物資1つの仕分けが約 10 秒で完了します
オフライン 機能	サーバと通信ができない場合は端末内で配送先を決定

支援者向け ～専用 Web ページ～

物資需要 確認機能	避難所で必要とされている物資の種類と数が集計されて確認可能 これにより不要なものや、必要以上の物資を送ることを防ぎます
伝票作成 機能	専用の web ページにて送る物資、個数を入力することで箱に何が入っているかを示す伝票を作成できます 伝票を箱に貼り付けて送ることで、集積所での仕分け作業が簡単になり迅速に避難所へ物資が届きます

4. 終わりに

支援物資のミスマッチは過去何度も起きています。私たちは、本システムを通して支援者・集積所・避難所を結び、ミスマッチを無くしたいと思います。そして、支援者団体や自治体の方と協力して、このシステムをさらにブラッシュアップしていきます。

13 津波避難エキスパート

金 沢

澤田 友樹(5年) 北 直樹(5年)
 中嶋 悠也(5年) 中山 義崇(5年)
 長井 健太(5年) 伊藤 周(教員)

1. はじめに

東北地方太平洋沖地震により、約1万6千人もの犠牲者が出てしまいました。そのうち約93%の人が津波によって亡くなっています。私達はその原因を、津波に対しての避難訓練やその経験が少ないことによるものと考えました。そのため、私達は今後起こりうる津波に対してこの様な事態を避けるため、津波避難体験用システム「津波避難エキスパート」を提案します。

2. システムの概要

本システムは、改造したルームランナーとWiiリモコンとコンパスセンサを用いて、津波警報の発生時から津波到達までの間に安全地帯を目指す「避難訓練」を目的としたシミュレータです。

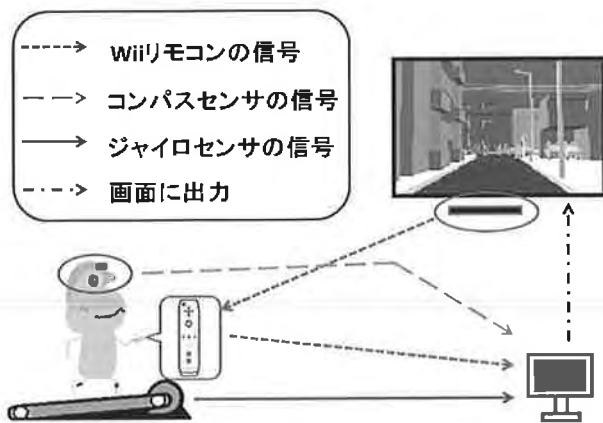


図1 システム概要図

3. システムの構成

ルームランナー上をプレイヤーが走る事で足に付けたジャイロセンサで移動を感知し、パソコンに無線で送信します。その際の移動距離に応じて画面内の景色も進んでいきます。画面上での進行方向はコンパスセンサによって調整します。Wiiリモコンは、画面上に表示された道具の取得等に使用します。道具にWiiリモコンをかざすことで、座標を取得しパソコンにデータを送ります。上記の3つの主な機能を用いて、避難状況を再現する事が出来るシステムとなっています。

4. システムの機能

4.1 ルームランナー

プレイヤーは好きな速度で走り、その速度をジャイロセンサで検知して移動量をパソコンに送ります。これによりプレイヤーの動きを画面上に反映します。

4.2 Wii リモコン

必要な道具を取るときや、ドアを開けるときに使います。Wiiリモコンのポインタを道具に合わせて、道具が入手できます。

4.2 コンパスセンサ

頭に装着したコンパスセンサが方位を感知し、プレイヤーの頭の向きを調べ、画面上の進路に反映します。

5. まとめ

本システムにより、津波に対して実践に近い経験を得ることができます。これにより、津波に対する知識を体で覚えることを期待しています。また、この経験から津波に関心を持つきっかけになればと思います。



図2 実際の様子

14

i-BadgeOVERしまNET

—子供見守り防災システム—

弓削商船

檜垣 俊希(専1年) 岡野さくら(5年)
 亀島加奈恵(5年) 福羅亜利沙(5年)
 村上麻矢加(5年) 田房 友典(教員)

1. はじめに

現在、全国各地で、子供が安心・安全に暮らすために様々な取り組みが行われています。例として、地域の方々による子供の見守りや、携帯電話を活用した位置情報通知システムが挙げられます。しかし、地域の方々による見守りだけでは保護者が確実に子供の状況を把握することはできません。また、ICT を活用するために災害時にも対応できるインフラ整備が進められていますが、離島では未着手であるのが現状です。そこで私たちは、現状の見守りに ICT を取り入れ利便性を高め、さらに災害対策を施したシステム「i-Badge over しま NET」を開発しました。

2. システム概要

本システムは、子供の位置情報を発信するデバイス「i-Badge」と、災害時にも稼働する独立したネットワーク「しま NET」によって構成されます。「i-Badge」は、防犯ボタン、スピーカー、マイコンボード、充電式電池、Xbee（無線通信デバイス）、GPS で構成される子供が容易に持ち歩けるデバイスです。この「i-Badge」を、簡易なネットワーク構成が可能な Zigbee の特性に着目して開発した「しま NET」で稼働させることで、いつ何が起きても、自分の子供の状況を把握できる環境を実現します。

3. 「i-Badge over しま NET」の利用

登校（外出）時、子供に必ず「i-Badge」を持たせる習慣をつけさせることが大切です。

3.1 日常生活の中で

通学路沿いにある家庭の窓際へしま NET（Xbee ルーターモード）を設置し、ネットワークを構築します。このネットワークを通じて、子供が持っている「i-Badge」の位置情報を、保護者のもとへ届けます。

PC やスマホからアクセスできる「B-MAP（位置情報表示アプリケーション）」で保護者は容易に子供の位置情報を確認することができます。また、「B-MAP」には



図1 日常生活の利用

通知設定があり、あらかじめ自宅位置等を登録しておけば、子供の帰宅通知をメールで受け取ることができます。家に持ち帰った「i-Badge」を充電することで、毎日使用を続けることができます（図1）。

3.2 災害が起こったとき（「しま NET」が不通状態）

災害によって「しま NET」が不通状態に陥った場合、R-DEVICE の無作為投入により、「しま NET」を再構築します。R-DEVICE とは、Xbee と電池で構成されるネットワーク復旧デバイスです。「しま NET」の復旧により、「i-Badge」を持っている行方不明の子供を探すことができます。もしも、「しま NET」が再構築できなかった場合は、B-SEARCH(i-Badge 検索デバイス)を用いて、「i-Badge」の電波強度によって位置を検索したり、不通の「しま NET」に接続し GPS 情報を取得したりすることで、安否確認ができない子供の検索を行えます。

4. おわりに

災害時は勿論、停電時でさえ現在のネットワークは島の中で利用できないと予測できます。私たちは、災害時にも運用でき、容易に再構築可能な「i-Badge over しま NET」を提案します。「しま NET」は、同時に 6 万以上のデバイスと接続可能で、今後、新たな Badge の開発により、独居老人や痴ほう症患者の見守り、船釣りをする家族の見守り、遠隔監視等、さまざまな応用例を考えることもできるのです。

15

Sneaker@mail

—心配なあの人に届ける便り—

沖 縄

近藤 史麻(5年) 久保田明成(5年)
下地 春希(5年) 川満 大輝(5年)
貝盛 陽平(1年) 鈴木 大作(教員)

1. はじめに

近年では災害に巻き込まれたとき、地理的距離を超えて安否の確認ができる、電話やメール等の ICT サービスを多くの人々が活用するようになりました。しかし、東日本大震災では通信インフラが破壊された地域においては、メール等の既存の ICT サービスのほとんどが利用できず、安否確認が迅速にできませんでした。

そこで、私達は災害などにより通信インフラが破壊された場合でもメールを使って大切な人の安否確認ができる「Sneaker@Mail」を提案します。

2. システム概要

Sneaker@Mail は、人がデータを持ち運ぶスニーカーネットを使い、メールアドレスで安否確認ができる環境を提供するシステムです。インターネット上に設置された中央システムと各避難所に設置された避難所システムで構成され、主に以下の様な役割を果たします。

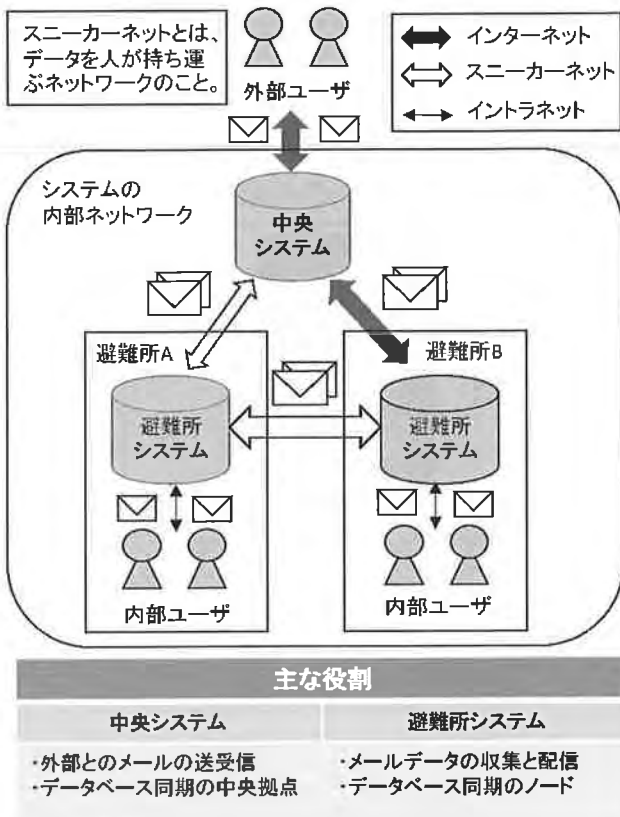


図1 システム構成と主な役割

3. 主な機能

3.1 メールを送受信

避難所にいるユーザは専用のクライアントアプリでメールを作成し送信すると避難所システムにメールが蓄積されます。蓄積されたメールデータは、インターネットやスニーカーネットを通じて、中央システムに運ばれ、そこから宛先のメールアドレスにメールが送信されます。中央システムから送信されたメールの送信元メールアドレスにはユーザ個別の ID が割り当てられており、そのアドレス宛に返信すれば、避難所にいる特定ユーザへメールを送ることもできます。

3.2 簡単ユーザ登録と簡単ログイン

Sneaker@Mail は、NFC タグを使うと簡単にユーザ登録とログインができます。登録の際は、未使用の NFC タグを NFC リーダにかざすことで、自動で NFC タグにシステムが発行したユーザ ID とパスワードが書き込まれ、ログインの際に NFC リーダにタグをかざすだけで、簡単にログインが出来ます。

3.3 簡単同期

簡単同期は、システムのデータベースの同期処理を自動化し、だれでも簡単に同期処理が行えるようになる機能です。スニーカーネットを通じて同期処理をする際には USB フラッシュメモリを使い、簡単に同期処理を行うことができます。また、インターネットに接続できる環境であれば、中央サーバに接続し、同期処理を行います。この機能により、システムの運用面での技術的負担を大きく減らすことができます。

4. 終わりに

Sneaker@Mail は、通信インフラが破壊され、既存のメールサービスが利用できない状況下において、被災者やその関係者が安否確認において抱える不安を解消できることが期待されます。

1. はじめに

2011年3月11日に東日本大震災が発生し、多くの人々が犠牲となった。この中で行方不明者の方の人数は2014年3月11日の時点で2633人であり、死者と行方不明者の方々を合わせたうちの14%である。そして、3年たった今でも行方不明の家族や友人を探している人は多くいる。この現状を受けて、これからは少しでも捜索が容易になるように、GPS腕時計である「キズナ時計」の活用を提案する。

今回提案するシステムは、キズナ時計とそれに対応するアプリケーションをインストールしたデバイスを使用することにより、キズナ時計をつけている人の位置を特定することができるものである。

2. システム概要

キズナ時計は搭載しているGPSモジュールにより取得した位置情報を定期的にサーバーに送信し、蓄積する。そして、これらの位置情報はスマートフォンやパソコン等の外部デバイスに対応するアプリケーションをインストールし使用することで、GoogleMapと連携して確認することができ、日常と直前の情報から大まかな位置の推測が可能である。

また、探索者が一定の距離まで近づくと腕時計内に内蔵したスピーカーにより音が発し、詳細な位置の確認が可能になると考える。

この二段階の捜索手段により、今までは困難であった一般の方々による行方不明者の捜索が容易にできるようになる。(図1)

それぞれの通信方法として、キズナ時計からサーバーへ位置情報を送信する際にはWiFi通信を行い、WiFiに接続されている間は位置情報を送信し続ける。

そしてサーバーに蓄積された情報はインターネットを介して確認することができる。キズナ時計と外部のデバイスとは、テザリングを行うことで一定の距離

にいるかいないかを判断する。

他の機能として、キズナ時計にはLEDチップのライトを内蔵されていることで明かりをとることができます。そのため、緊急時に視覚情報の確保を行うことに役立てることができる。

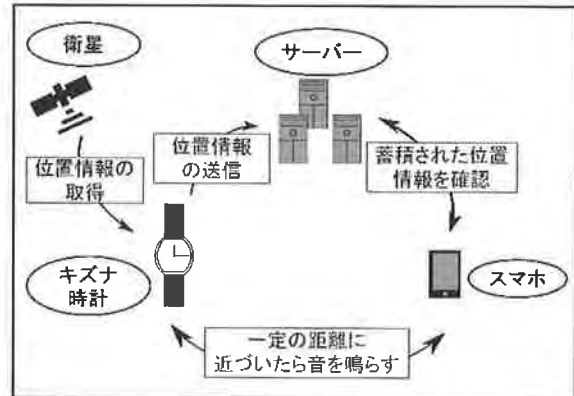


図1 全体の構成図

3. アプリケーション

キズナ時計に対応するアプリケーションはアンドロイド上で動作し、GooglePlayに登録し、無料で提供する。これによって、キズナ時計を購入した人だけでなく、家族などの他にキズナ時計を所持している人の位置情報を確認することが可能である。(図2)



図2 アプリのイメージ図

4. まとめ

「キズナ時計」は単なる時計機能だけでなく、GPSと内蔵スピーカーによる二段階の捜索手段を可能にすることで、行政機関による行方不明者の捜索だけでなく、一般の人達が自分で捜索することを容易にできる。

17

MAP FIXER

—災害時に役立つ町内道路地図アプリ—

熊本(八代)

木村 匠(5年) 関原 至音(4年)
 湯舟 武龍(3年) 岩下 将大(3年)
 本山 和輝(1年) 小島 俊輔(教員)

1. はじめに

災害の影響で既存の地図と実際の道路状況が異なると、避難や物資の輸送に時間がかかります。また、道路状況をリアルタイムに反映した地図を迅速に作ることは難しく、路地などの細い道まで対応することは困難です。

そこで、地図編集の知識がなくても、Android 端末で道路状況を発信できるアプリケーション「MAPFIXER」を開発しました。

2. アプリについて

「MAPFIXER」のアプリには、地図編集と地図閲覧の2つの機能があります。

2.1 地図編集

地図の画像データに線を引くだけで、地図編集の知識がなくても簡単に情報を発信できます(図1)。道路状況に合わせて4色の線を使い分けます(図2)。オフラインでも地図編集が可能で、その場合はオンライン時に自動でアップロードします。

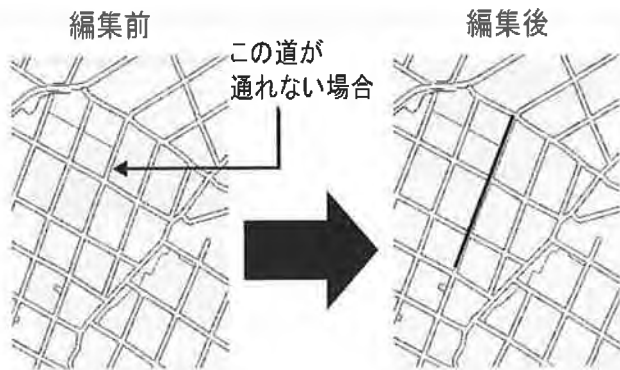


図1 「MAPFIXER」のイメージ

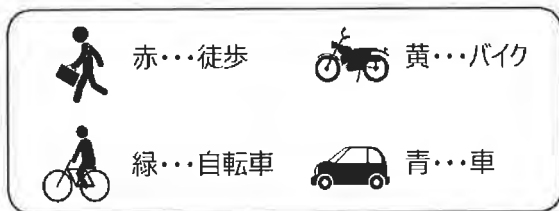


図2 線の種類

2.2 地図閲覧

他のユーザが書き込んだ情報は、サーバを通して閲覧できます。オフライン時は、以前に閲覧した地図の中で最新のものを表示します。

3. 信頼性の確保

災害時の情報には、より高い信頼性が必要になります。「MAPFIXER」では、ユーザ登録の機能を設けることによって情報の信頼性を確保しています。また、登録しているユーザが引いた線のみを表示することにより、見やすく、信頼性のある地図を提供できます。

4. システム構成

「MAPFIXER」は Android 端末で動作するアプリとサーバで構成されています。このサーバは、地図の画像データ (OpenStreetMap) や編集された地図のデータ、ユーザ登録のデータを格納しています。初回起動時に地図をダウンロードし、また、アプリケーションの起動時に更新された地図データを取得します(図3)。

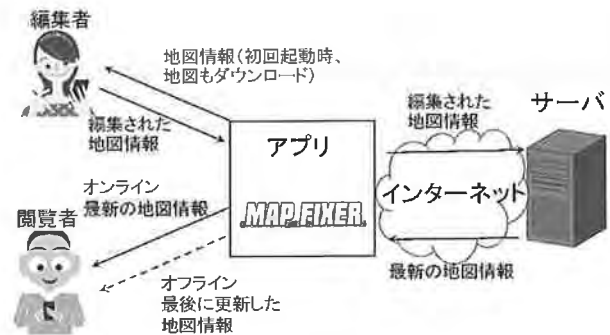


図3 システム構成図

5. まとめ

「MAPFIXER」により、災害の影響で通れない道が把握でき、スムーズな移動を可能になります。また、その場で簡単に編集できるので、誰でも迅速に情報を発信できます。

「MAPFIXER」を利用させていただく事で、ボランティアの方々の支援活動や物資搬送などのサポートをできることを願っています。

18

つながってタワー

—津波避難タワー間を結ぶ安心防災システム—

高知

佐々木 渉(5年) 島内 良章(5年)
南 光成(5年) 森國 健吾(5年)
今井 一雅(教員)

1. はじめに

2011年3月に起きた東日本大震災での被害は、9割以上の犠牲者が津波によるものでした。今後想定される南海トラフ地震でも高い津波が来ると考えられており、その対策として高知県では、2015年度までに100基の津波避難タワーが建設されることになっています。

そこで私たちは、地震発生時に海岸沿いに設置されている携帯基地局に被害が及び、モバイル回線が使用できない状況下でも、津波避難タワーに避難してきた人が携帯情報端末で確実に安否情報の通信ができる「つながってタワー」システムの開発を行いました。

2. システムの構成

図1のように各避難タワーを指向性アンテナで無線LANに接続し、複数の避難タワー間のLANを構築します。このLAN内にサーバを設置し、データベースサーバ、SIPサーバの機能を持たせ、LAN内でのチャットやSIP電話機による安否確認のデータ通信を可能にします。また、中核となるメインタワーには衛星インターネット用パラボラアンテナを設置し、衛星インターネットによるVPNを介して市役所との通信も可能とするシステムとなっています。そして、各避難タワーに避難してきた人の持つスマートフォンがiBeaconに反応して、利用者に安否情報の送信を促し、簡単な動作で安否情報を衛星インターネット経由で市役所に送ることのできる世界初のシステムを構築しました。

3. 津波避難タワー用スマートフォンアプリの開発

私たちは、津波避難タワーに避難してきた人に活用してもらえるようなスマートフォンのiOSアプリ「つながってタワー」を開発しました。開発したアプリには図2のような機能を搭載しています。災害時のみならず、平常時での使用もでき、多くの人に使ってもらえるアプリとなっています。



図2 「つながってタワー」アプリの機能一覧

4. おわりに

本システムにより、地震によるモバイル回線切断時においても、津波避難タワー間での安否情報の共有や、市役所などの対策本部への安否情報の通知がリアルタイムで可能となります。タワーの存在が日々の安心につながるように、本システムは強力にサポートします。

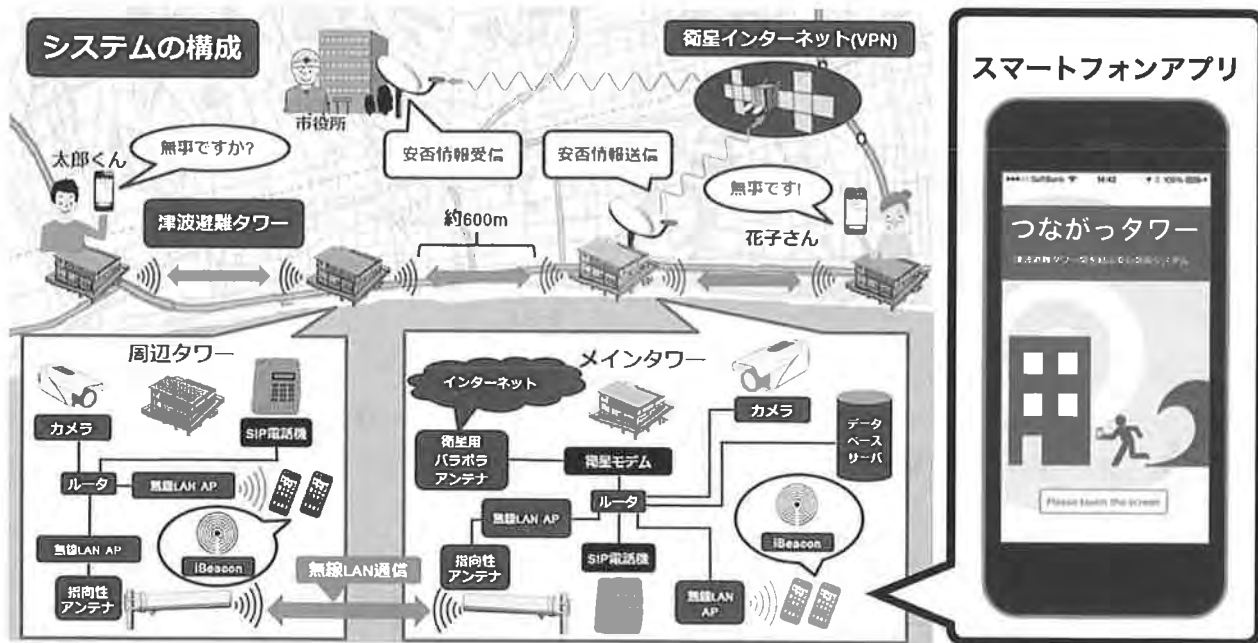


図1 津波避難タワー間を結ぶ「つながってタワー」システムの概要

19

人（ヒート）マップ

—みんなで作る情報マップ—

鳥羽商船

伊藤由祐紀(4年) 竹口 優里(4年)
 谷口 昂汰(4年) 萩原 海仁(4年)
 濱口 堅太(3年) 江崎 修央(教員)

1. 「人(ヒート)マップとは

近年、日本ではゲリラ豪雨や局所的な大雪などの異常気象が発生しています。わずかな時間で道路が冠水したり、急な土砂災害による被害が後を絶ちません。そこで、私達は、地図上にさまざまな SNS の情報を表示し、二次被害を防ぐ「人(ヒート)マップ」を開発しました。

「人(ヒート)マップ」を使えば、その場にいないと分からない状況を容易に知ることができ、有事の際に「待機する」や「出かける」といった判断ができます。

「人(ヒート)マップ」では、Facebook のチェックイン数の変化をヒートマップで示すことで、ある時間にどれくらいの人が集まっていたか分かります。また、Twitter のツイートを行った位置に吹き出しを表示することによって、その場所で何が起こっていたのかが推測できます。これらの情報は 2 4 時間前まで遡って確認可能です。

2. 利用する状況

2.1 急な大雪の場合に子供を迎えに行く

大雨や大雪によって帰れなくなった子供を迎えに行くこうとするとき周辺状況を把握できます。「渋滞の少ない所や被害の少ない道を通って安全に迎えに行く」、もしくは「どのような経路を通っても危険なため、自宅で待機する」などの判断が可能となります(図1)。



図1. 送迎時の利用例

2.2 土砂災害の場合に避難所に避難する

土砂災害が発生し、自宅から避難所に避難しようとするときに発生当時から現在の状況を確認できます。タイムバーを動かし、過去のヒートマップやツイートから周辺の状況を把握することで、「被害の少ない道を通って避難所に避難する」、「避難経路の被害が大きいため、自宅や近くの高い建物に避難する」などの判断が可能となります。

平成26年8月に広島県で起きた土砂災害の際には、災害発生時の交通状況や避難指示・避難勧告の情報から、災害後の避難所での状況やボランティアの活動内容などを知ることができました(図2)。



図2. 避難時の利用例

3. 実現方法

「人(ヒート)マップ」は、自動収集プログラムによって、Facebook と Twitter からデータを収集し、解析したものを、「人(ヒート)マップ」のデータベースに記録し、ブラウザで表示できるようにします(図3)。

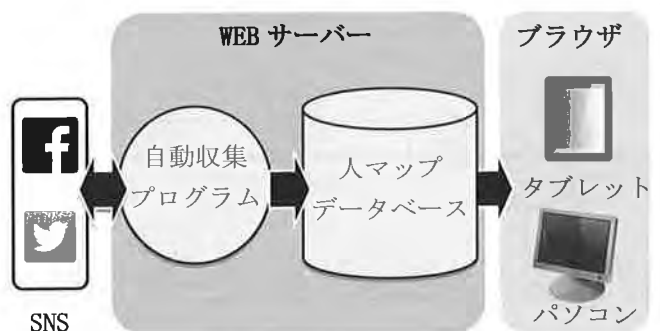


図3. システム構成図

1. はじめに

日本は、比較的自然災害の多い国であり、大規模な災害の影響として通信インフラの機能停止は避けられない事象となってきた。そこで、従来の通信インフラが機能していない時に、誰もが簡単に必要な情報を共有する事の出来るツール「WT」を提供する。WTは、近年急速に普及して来ているスマートフォンに搭載されている近距離通信のBluetoothを利用し、パケットリレーのように安否情報や簡易メッセージを拡散することで、被災地で携帯キャリアが復旧するまでの通信をサポートします。

2. システム概要

WTは自分の置かれている状況を素早く広げて行く事が出来る。アプリの初回起動時に割り振られるIDを他の人と認証し合うことでその人のメッセージを受信した際に表示させる事ができる。WTのメッセージは、宛先名・ID・安否情報・送信時間などを、第三者に見られないように暗号化して蓄積し共有・拡散する。データの蓄積されたスマートフォンが、Raspberry Piによって構成された専用の小型サーバ(WTルータ)に接続できる場所に移動すると自動的にアプリ専用のサ

ーバーにデータが暗号化され送信される。端末内のデータは一定期間が過ぎると削除される。サーバーに上げられたデータは専用のウェブサイトから確認したいユーザーの連絡先を入力する事で、パソコンやスマートフォンなどで確認する事ができる。

3. ビジネスモデル

WTは多くの人々が利用してすれ違いが成立しなければ、最大の特徴を生かす事ができない。それを改善するために以下のビジネスモデルを提案したい。各地域と提携し、地域に根差した防災アプリの中にWTを組み込んで運用することで、利用者を増やすことが期待できる。さらに各企業と提携し、各企業が販売している電子機器などにWTをベースとしたアプリケーションを導入してもらう事で、WTの利用人数を増やしデータ共有の効率を上げる。

4. まとめ

以上の事より、WTを本全国各地のユーザーに利用してもらうことで、災害時に発見が遅れた人・出来なかった人々を今まで以上に発見できやすくなる。

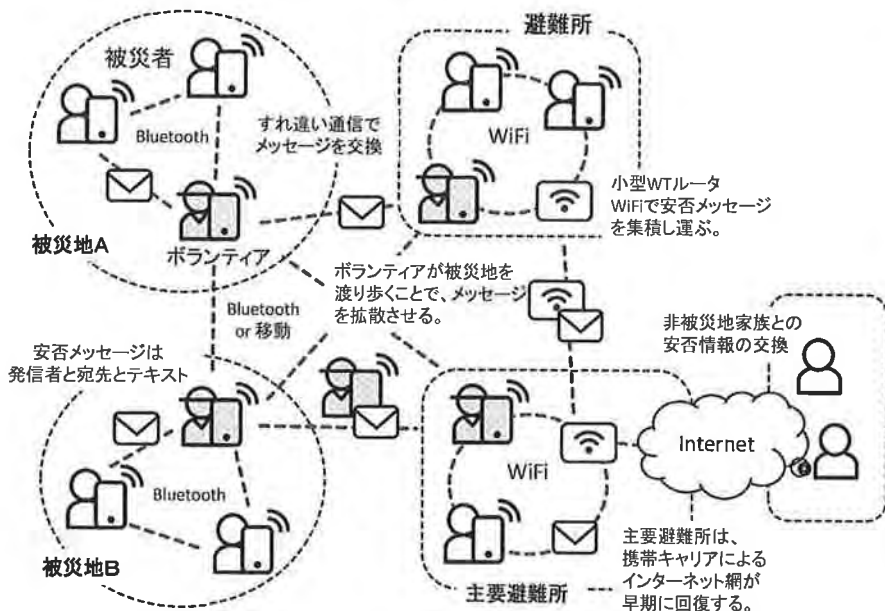


図1 構成とデータの流れ

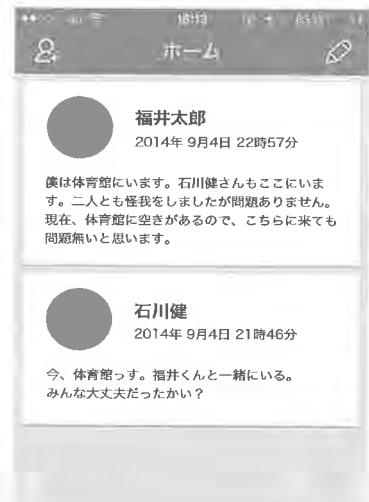


図2 アプリ動作画面

21

SKcAP

—空から支える復興支援—

宇 部

金子 昂稔(4年) 西村 優佑(4年)
伊藤 賢也(4年) 田代 翔也(4年)
久継 宏樹(4年) 田辺 誠(教員)

1. はじめに

5月に行われた高専プロコン連携シンポジウムにおいて、東日本大震災の災害復旧に携わられた大船渡市職員の方から話を聞く機会がありました。災害対策本部の設置直後は、被害状況の情報収集を人が歩きまわって行っていたとのことでした。

災害当時の状況を考慮すると、人が歩いて情報を収集することは危険が伴うと同時に、情報収集の範囲も限られます。もっと安全に多くの情報を迅速かつ正確に入手する手段を確立することにより、復旧や復興がよりスムーズに進むことが期待されます。

2. システム概要

「SKcAP」はマルチコプターとPC上のソフトウェアから構成されます。ソフトウェアは、ハザードマップ作製ソフト、およびマルチコプターの撮影した全方位画像を展開処理するソフトウェアの2種類を提供します。SKcAPを各自治体に設置することにより、災害後の安全確認や危険な場所の迅速な把握を支援します。



図.1 全体概要



図.2 マルチコプター

3. 機能

3.1 ソフトウェア

ソフトウェアは地図上をマウスクリックすることにより、マルチコプターの飛行ルートを設定します。設定内容をマルチコプターに送信し、自動飛行を行いながら経路上で写真の撮影を行います。飛行後、撮影さ

れた全方位画像にパノラマ展開処理を施し、GPS データをもとにオフラインの地図上に配置します。地図上に配置された画像をクリックすると 3D 空間でパノラマ展開画像が表示されます。カメラ視線をマウスで自由に変えることにより被災状況を詳細に確認できます。

ユーザーが瓦礫や土砂崩れのある危険な場所を地図上にマークできる機能も提供します。被災状況の確認や支援検討に役立てることができます。

3.2 マルチコプター

マルチコプターは自作し、全方位カメラを搭載します。ソフトウェア上で設定された経路上を 3~5m の低空で自動飛行し、360° パノラマ写真をインターバルで撮影し戻ってきます。更に、マルチコプターにスピーカーを搭載することにより、広報車の代わりに災害情報を伝えながら飛行します。

GPSにはJAXAの準天頂衛星みちびき(QZSS)の信号を取得できるものを使い、位置精度を飛躍的に高くすることができました。

4. 終わりに

本システムは災害状況がひどく、人間が入れなくなってしまった場所を安全かつ正確に、更に迅速に被災状況を確認することを目指し開発しました。

「SKcAP」で二次災害を防ぎ、災害状況の把握のサポートを行い、多くの人の手助けになると考えています。

22

(未定)

ペトロナス
工科大学

Muhammad Izzuddin Eshak
Mohd Syazwan Zakaria
Faizal Ahmad Fadzil (教員)

MEMO

A series of horizontal dashed lines for writing.

自由部門本選参加作品

発表順	タイトル	高専・大学名	指導教員	参加学生
1	すくえあ	香川(詫間)	金澤 啓三	金子 高大, 瀧下 祥, 東山 幸弘, 山崎 啓太, 山下 昂太
2	Writening —自分と 世界と つながる手帳—	津 山	寺元 貴幸	末田 卓巳, 佐伯 典祐, 岸本 大知, 奥田 智大
3	TeleRokuro —マルチプレイヤー3DCGビルダー—	熊本(熊本)	孫 寧平	村上 雅貴, 八木 智徳, 田中 亨, 山下 陽平, 中神 楓子
4	御座敷トラベラー —メガネをかければ今ここから広がるセカイ—	徳 山	高山 泰博	小松 弘人, 宮本 颯, 片井 拓弥
5	CT・MRIによる骨粗鬆症の3次元解析	新 居 浜	平野 雅嗣	芝山 晃弘, 伊藤 紫芳
6	NaVi —Navigate by Vibration—	長 野	伊藤 祥一	宮前 海里, 小泉 拓海, 牛澤 葵, 酒井 聖矢, 長谷川 就
7	postKey —会議室らくらくシェアシステム—	鈴 鹿	田添 丈博	大森 智喜, 小野せりな, 佐藤 幹人, 堀 未来, 黒澤 預生
8	STEP APP!!	熊本(八代)	開 豊	岩本 優, 上田 尚人, 渡邊 兼誠, 西崎 友輔, 松岡 雄太
9	あみがみ様 —3D編物デザイン支援システム—	米 子	松本 正己	足立奈々美, 生田 悠華, 谷口 正明, 山崎 千裕
10	CosmoSweeper —スペースデブリ回収体験システム—	八 戸	釜谷 博行	奈良 公明, 竹林 雅人, 永田 大也, 鈴木 康太, 唐牛良太郎
11	cosmo craft	久 留 米	嶋田 英樹	辻 巧斗, 千北 一期, 船越 南斗, 久保田祥平, 江崎 喬祐
12	はなまるフォーム —スポーツ上達サポートアプリ—	鳥羽商船	都築 啓太	小野 陸翔, 池田 友美, 小山 紗希, 勝田 百香, 河口 祭
13	Code on the Tablet	弓削商船	長尾 和彦	古谷 勇樹, 林 真史, 山本 隆弘, 井上 香澄, 松川 瑞樹
14	Rapid House —サーバ自動チューニングシステム—	釧 路	神谷 昭基	森越 友祐, 尾崎 一幸, 加藤 頑馬, 風間 健祐, 真田慧一朗
15	すごろく観光 —ゲーム型 観光サポートアプリ—	金 沢	藤沢 武	小野 叶太, 遠矢 勇樹, 外松 俊尚, 川上 悠太, 山崎 寛人
16	おそらく王手 —AR量子将棋盤—	福 島	小泉 康一	青木 僚平, 岩本慎太郎, 小林 尚輝, 川崎 裕幸, 間野目 大
17	ポケットレジスタ	阿 南	児島 雄志	鈴江 翔樹, 瀧本 大介, 森井 理智, 岡田 大樹, 本浄 定俊
18	SNS炎上報知器 —個人情報流出・炎上防止見守りシステム—	東 京	松林 勝志	松林 圭, 松原 良和, 五味 京祐, 古川 和祈, 松尾 祐佳
19	Remoconia —ひとつの家庭にひとつのリモコン—	沖 縄	正木 忠勝	山本 翔翔, 辺土名朝飛, 知念 響紀, 當間 環, 當山 大騎
20	ネッチュー —熱中症対策システム—	舞 鶴	船木 英岳	治居 直哉, 渡部 翔大, 藤村 匡弘, 石本 浩気, 浦 優輝
21	(未定)	ペトロナス 工科大学	Ahmad Izuddin Zainal Abidin	Mohamad Ali Hajian Maleki, Muhahmmad Fadzil Mohd Jusri

1. はじめに

扇風機や掃除機、航空機、風力発電など、歴史的にも私たちの生活は「風」に支えられてきました。人々の生活においてあたりまえの存在である「風」。しかし、コンピュータなどのデジタルな面での利用は、あまり進んでいません。そこで私たちは、「風」を情報として入力し、「風」の表情を感じさせるシステム“SCREEN feels AIR. (すくえあ)”を提案します。

2. システム概要

“すくえあ”は風を検知するスクリーンに向かって息などの実際の「風」を送り、コンピュータとのインタラクティブを楽しむことを目的とした、エンターテインメントシステムです。息だけでなく、うちわやエアスプレーなど「風」を起こすものであれば、どんなものでも入力として使用することが可能です。

2.1 風圧センサアレイ「すくえ Array」

本システムの要となる自作2次元風圧計測デバイス「すくえ Array」は、磁石を取り付けたフィルムとホール素子との距離によって変動する磁界強度を風圧として検出します。風を検出する仕組みを図1に示します。このセンサを格子状に配置し、各センサから計測された風圧を風圧分布としてパソコンに送信します。

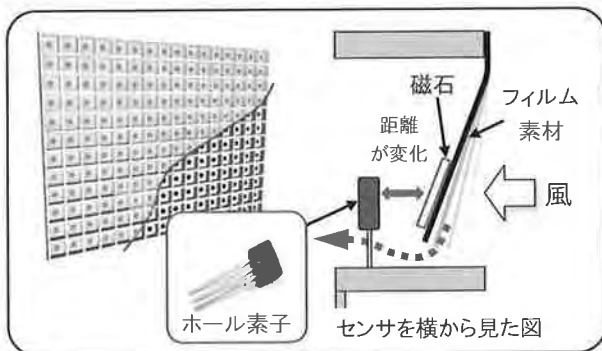


図1 風を検出する仕組み

2.2 気流シミュレーション

入力された風圧を基に、気流のシミュレーションを行いアプリケーションに利用します。離散化には、格子ボルツマン法を用い、GPGPUによって並列計算し、リアルタイムにシミュレーションします。

3. システムの機能

■わくわく実験室モード

格子ボルツマン法を用いた気流のシミュレーションを基に「風」を使った実験をコンピュータ上でを行い、3次元CGによって可視化するモードです。

■クリエイティブアートモード

「風」特有の流れが、神秘的な作品を作る「吹き絵」などのアートを体験することができるモードです。

■パーティゲームモード

「すくえ Array」の入力特性を生かしたミニゲームを複数人で体験するモードです。

4. 処理の流れ

本システムの処理の流れを図2に示します。網目状のスクリーンを通過した風を2次元の風圧分布としてコンピュータに送信し、コンピュータでの処理の結果を映像としてスクリーンに投影します。

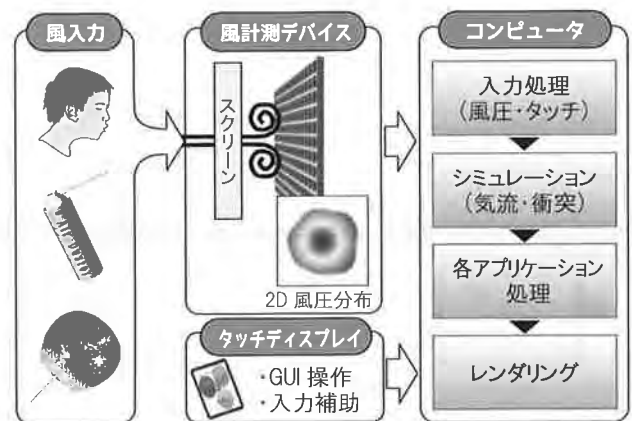


図2 システムの処理フロー

5. システムの応用例

本システムの特徴は、「風」を入力として使用することです。そのため、例えば料理中や骨折などで手が動かせない状況の時でも、活用できる新しいインターフェイスとしての応用が考えられます。

6. おわりに

“すくえあ”。それは「風」を感じる驚きのシステム。スクリーンの向こうに広がる、新しい「風」の世界を体験してみませんか。

1. はじめに

近年、スマートフォンを手帳のように使う人が多くなりました。しかし、今も紙の手帳を利用している人は多く存在します。

その理由は紙とペンで素早く書けることと、「紙に書く」行為の持つ物質性ではないでしょうか。

そこで私たちは、紙の手帳を活かしたインテリジェントな手帳システム「Writening」を提案します。

またさらに、手書き文字の良さを活かせる手書き SNS「Writr」も同時に提案します。

2. システム概要

ユーザは、Writening で提供される専用デジタルペンと専用手帳のセットを今までの手帳と同じように使います。

デジタルペンで書かれた筆跡はスマートフォンで解析され、これから得られるたくさんの情報をもとに Writening はインテリジェントな機能を提供します。

3. システム構成

図1にシステムの構成図を示します。

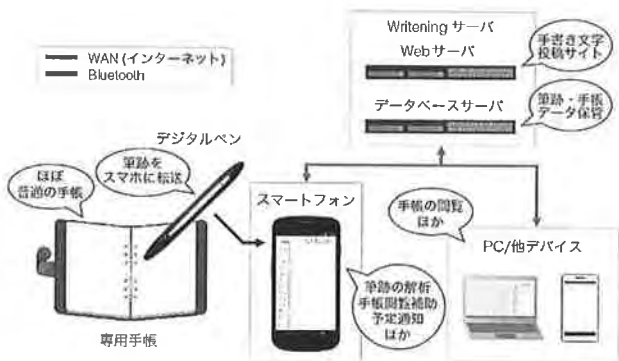


図1 システム構成

3.1 手帳

デジタルペンが読み取る特殊なパターンが印刷された、専用のリフィルで構成されるシステム手帳です。見た目も使用感も通常の手帳と同じように作られており、需要に応じて多種多様なサイズとデザインに対応できます。

3.2 スマートフォン

デジタルペンから Bluetooth を介してやってくる筆跡をバックグラウンドで随時解析し、自律動作できない手帳に代わり後述する予定通知などを提供します。メモや予定の絞り込み検索など、あくまで紙の手帳を支える存在として使われます。

4. 基本機能

4.1 予定のリマインダ通知 (スマートフォン)

従来の手帳は、予定を書き込むことはできても、間近の予定を通知してくれることはありませんでした。Writening では、予定があった事そのものを忘れてしまうような方でも安心の通知機能をスマートフォンを介して提供します。

既に存在するスケジュール管理サービスと同じように、間近の予定のリマインダを手帳に書いた予定に対しても得ることができます。通知は予定が書き込まれると自動でセットされるので、何ら余計な手間はかかりません。

4.2 絞り込み検索 (スマートフォン)

とっさのメモを書く際にはよく、メモ欄の空いたスペースに走り書きしてしまうものです。スマートフォンで“いつ・どこで・何をしていた時”といった条件でメモを絞り込み表示できるので、「あの日ばらばらに書いたメモをすべて取り出したい」・「毎週の定例会議に関するメモだけ見たい」などの需要に対応します。

4.2 他デバイスでの閲覧

Writening では OCR によるテキスト変換に加えて、PC などの画面上でメモを閲覧する簡易的な機能を提供します。

5. システムの応用

5.1 手書き SNS「Writr」

データになった手書き文字を手のひらの中だけに留めるのはもったいないと考え、私達は手書き文を投稿しユーザ同士で交流できる全く新しい SNS「Writr」を提案します。専用手帳の記入欄に専用ペンで書(描)けるものなら何でも投稿でき、閲覧はスマートフォンの Writening アプリ内ですることができます。

角ばった文字や丸い文字、小さなイラスト。Writr はテキストだけの既存の SNS にはない「気持ち」を伝えることができます。

6. おわりに

Writening は、今まで相容れない存在だった「手書き」と「デジタル」の双方に、未だかつて無い架け橋を渡す存在です。手帳にこだわり続けている方も、スマートフォンを手帳代わりにしている方も、今こそ究極の手帳を手にしてみませんか？

1. 離れていても共同で立体モデルを組めるシステム

3DCG モデルの作成には、専用ソフトが必要です。しかしながら現在の CG ソフトはワンライセンスでワークスペースになっており、常に作業用のパソコンを持ち運ばなければなりません。そこで、データベースを介してデータをやり取りすることで、グループで3DCG モデルを製作でき、かつグループ間で物理的に距離が離れていても共同して製作できるシステム「TeleRokuro」を開発しました。

表1. 開発環境

CPU	Intel® Coer™ i5-4570
GPU	GeForce GT 640
Memory	4.00GB
Programming Language and API	Java jdk1.7.0_51, OpenGL1.2, JDBC : mysql-connector-java-5.1.27
OS	Windows 8.1
Database	MySQL 5.6

2. 「TeleRokuro」の使い方

直接会わずに1つの作品を共同で制作する3DCGデザイナーや一般の愛好者は「TeleRokuro」を使って、相手との距離が遠くてもインターネットを介してものづくりが可能です。また、子供たちが積み木やドミノなどを作ったりして遊ぶこともできます(図1)。

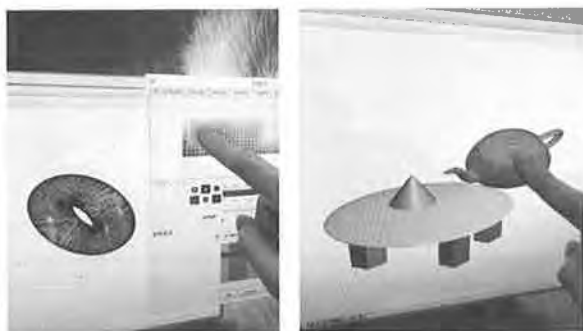


図1 操作画面

各自の端末で TeleRokuro にログインしたクライアントたちは、製作グループを作成したり、参加したり

して、3DCG モデルの製作に取り組みます。

グループ内では、おのおの描画操作が行え、グループ全員の画面にはその操作がリアルタイムで反映されます。

3. システムの構成

ビルダーサーバーには、2つのデータベースを構築します。すべてのユーザー情報を登録し管理します。更に、描画されるオブジェクトのモデリングとレンダリング情報をリアルタイムで管理、更新、定刻に各クライアントと通信します。

「TeleRokuro」にログインしたクライアントたちは、各自の端末で同時に同じビルダーへアクセスし、独自の描画操作を行います。描画したモデルの形状、色彩、動きなどをアクションごとに自動でビルダーサーバーへ送信します。最新のモデル描画状況が0.01秒刻みにキャンバス上に反映されます。

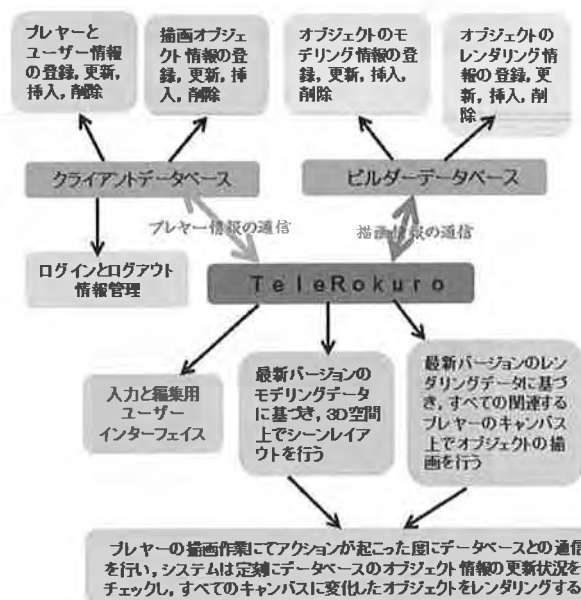


図2 システム構成図

4. おわりに

みなさんも、「TeleRokuro」を使って、離れた人と楽しく3DCGモデルを製作してみませんか？

4

御座敷トラベラー

—メガネをかければ今ここから広がるセカイ—

徳 山

小松 弘人(5年) 宮本 颯(5年)
片井 拓弥(5年) 高山 泰博(教員)

1. はじめに

高齢化の進行とともに介助や支援を要する人が増加しており、その多くの人々が目的地への移動が難しい点と金銭面の余裕がない点から、旅行に行くことが困難な現状にあります。

そこで、自宅で椅子に座ったまま気軽に旅行を疑似体験できるシステムを開発しました。

2. システムの概要

Google Map 上の好きな地点を選択すると、ヘッドマウントディスプレイ (HMD) にその場所の Google Street View が表示されます。図 1 のようにユーザが頭を上下左右に動かすと、その動きに連動して Google Street View の視点が変わり、その場にいるかのような臨場感を演出します。

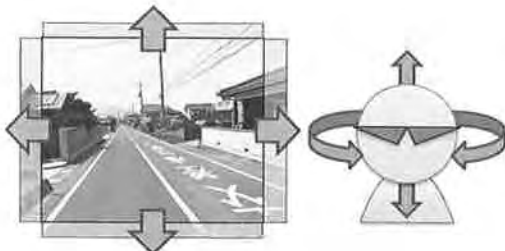


図 1 頭の動きとの連動

また、手元のコントローラのスティックを動かすだけで誰でも簡単に Google Street View の中を自由に移動できます。



図 2 コントローラの操作

他にも、Google Map 上でルートを描き、そのルートを自動で巡る機能もあります。コントローラによる移動操作が困難な人でも景色を眺めることを堪能することができます。

これらの機能により世界中の様々な場所で実際に旅行している疑似体験を味わうことができます。

3. システムの構成

本システムの構成は図 3 のようになっています。



図 3 システム構成

・ヘッドマウントディスプレイ (HMD)

眼鏡をかけたまま装着でき、首や頭への負担が少ない軽量なものを uses。60kHz 帯の WirelessHD (TM) によって PC と通信します。

・コントローラ

指先の操作だけで簡単に扱えるコントローラを uses。肘掛に固定し HMD を装着した状態でコントローラを見失う問題に対処しました。Bluetooth によって PC と接続します。

・センサ・Arduino

頭の動きを取得するために HMD に 6 軸センサを搭載します。Arduino を介して PC と Bluetooth で通信を行います。

・PC

Windows 7 以降が動作する PC を uses。本システムは Google Chrome 上で機能し、JoyToKey でコントローラのキーの割当て、Apache, Processing でセンサからの値の取得を行います。

4. おわりに

支援の必要な人が負担なく気軽に旅行を楽しめる環境として本システムが活躍することを期待します。

1. はじめに

近年、骨粗鬆症が高齢女性を中心に年々増加していることをご存知でしょうか？ これは女性ホルモンの分泌低下が引き起こす骨密度の低下が原因であり、早期発見による治療が求められています。

2. システム概要

このシステムは脊椎 CT 画像を読み込み、皮質骨と海綿骨の部分を3次元抽出し、そのヒストグラムと標準偏差及び平均値を解析し、グラフ化して出力するものです。こうして得られた実行結果をもとに患者の骨粗鬆症や骨折可能性について判断し、適切な処置を施すことで患者さんのQOLを向上させることができると考えています。

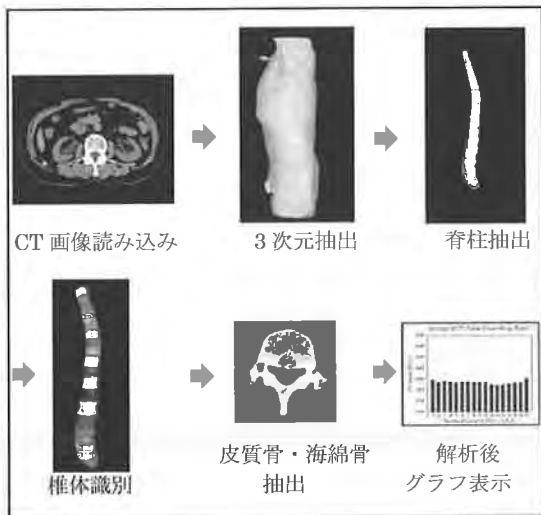


図1. システム構成

3. システム機能

グラフは全部で3種類出力され、それぞれの結果を基に診断を行うことになります。その3種類とはヒストグラム、標準偏差、平均値のことです。

ヒストグラムは横軸 CT 値、縦軸が頻度になっており、椎体ごとにグラフを作成します。しかし標準偏差、平均値の棒グラフは横軸が椎体番号、縦軸が CT 値になっています。そのため胸椎から腰椎までの範囲の椎体番号でグラフを作成しているので、骨のどこが弱いのが分かりやすいです。

標準偏差はデータのばらつきを表す量です。図2より左図と右図では、ばらつきが大きいのは右図であり標準偏差は右図の方が大きくなります。そして網目状に見える骨の

部分を見てみると密度の高い丈夫な骨は左図なので、標準偏差が高いほど骨が弱っているということが分かります。

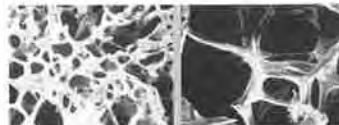


図2. 正常骨と骨粗鬆症症例の比較

(右図が正常骨、左図が骨粗鬆症例)

以下に実行結果の各グラフについて示します。(図3右: 海綿骨、左: 皮質骨)

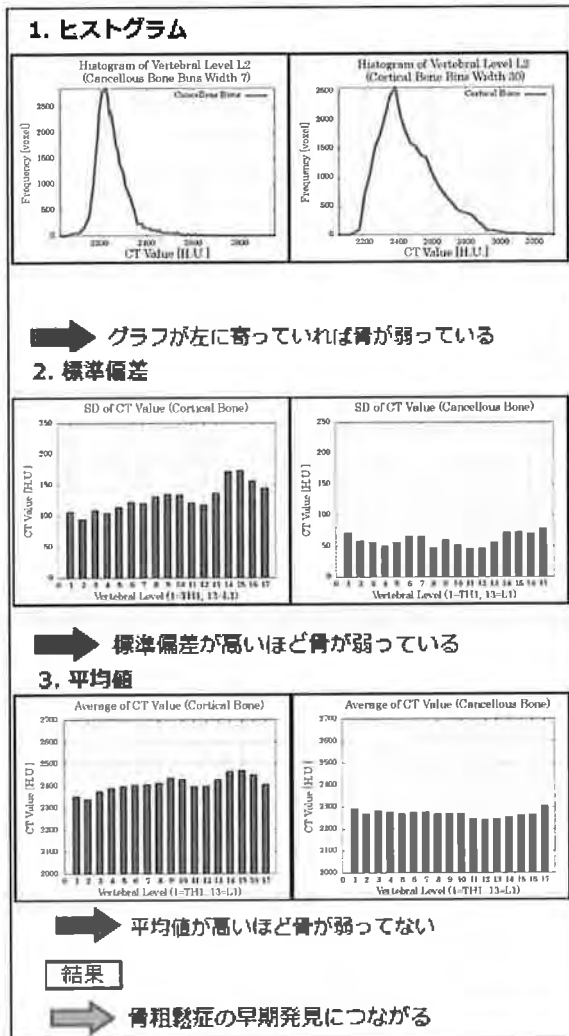


図3. 実行結果

4. おわりに

本システムを少しでも多くの病院でご利用していただき、骨粗鬆症の早期発見、及び膨大な画像を診断しなければならない医師の負担軽減に役立てることができれば幸いです。

1. NaVi とは

現在、日本では「ながらスマホ」による交通事故が増加傾向にあります。しかし、自転車に乗って遠出する際スマホ等で地図を見ながら運転したいという人もいるはず。本システムではリストバンドからの振動を用いて、視覚に頼らないナビゲーションシステムを提案します。利用者が地図を見るために視線を逸らす必要がなく、より安全に目的地まで案内できます。

2. NaVi の基本操作

NaVi の基本操作は3ステップです。

- ① 振動モータ内蔵のリストバンド(ナビリスト)を両腕に装着する。
- ② スマホでアプリを起動し、目的地を検索・設定後案内を開始。
- ③ スマホをポケット等にしまう。

これだけの簡単操作で利用することができます。

3. NaVi の機能

3.1 ナビゲーション機能

分岐点に近づくと、曲がる方向のナビリストが振動します。近づくにつれて図1のように振動パターンが変化し、分岐点の直前では両腕のナビリストが振動し曲がるタイミングをお知らせします。また、案内開始時と目的地への到着時に特殊なパターンでナビリストが振動します。

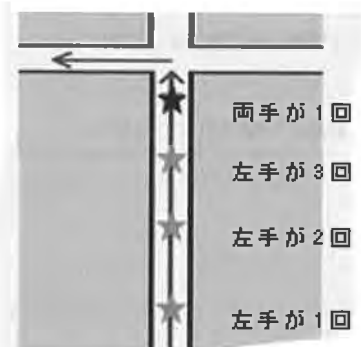


図1 分岐点での振動パターン

3.2 休憩ボタン

長時間の移動の際、途中で休憩をはさみたい場合があります。本システムのナビリストには休憩ボタンが搭載されており、これを押すことで最寄りの休憩ポイントへの案内に一時的に切り替えることができます。休憩ポイントはコンビニ、公衆トイレ等の種類をあらかじめアプリ内で設定することができます。

3.3 共有機能

複数人でサイクリングを行う場合、ルートを共有して走行することができます。誰かが休憩ボタンを押した際はその情報が全員に共有されます。休憩をする際に並走や仲間を振り返る必要がないため安全です。

3.4 記録機能

走行距離や走行時間などを記録します。カレンダーと同期して表示することができるため、長期的な運動計画などにも役立ちます。

4. システムの構成

NaVi のシステム構成は図2のようになっています。NaVi は案内をスタートすると Google Maps から地図データを取得し、Bluetooth を用いて振動するタイミング等の命令をナビリスト内のマイコンに送信します。そして、命令に合わせてモータが振動します。共有機能を使用する際は、サーバを介してスマホ同士の通信を行います。

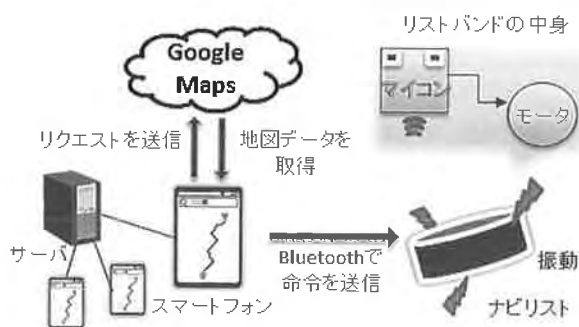


図2 NaVi の構成

5. おわりに

NaVi を用いてより安全で快適な自転車走行を楽しんでみませんか。

1. はじめに

学校などの施設で会議室や講義室を利用したい時、部屋の利用者には

- ・鍵を借りに行くのは手間がかかる
 - ・予約が早い者勝ちなので使いたい時に使えない
- という問題点、また管理者には
- ・使われてない部屋があるのはもったいない
 - ・複数の鍵の管理は紛失の可能性があり大変
- という問題点があります。

postKey は、スマートフォン1つで予約・鍵の開閉を行い、優先度を決定し自動的に部屋を割り振ることで、これらの問題を解決します。

2. postKey の使い方

2.1 部屋の利用者

利用者はまず、アプリケーションをインストールします。その後、管理者に発行してもらったアカウント(後述)でログインして利用を開始します。

部屋を借りたい時は、まずアプリケーションから申請をします。申請が認められたら、予約した時間に部屋の前にいだけで鍵が自動で解錠され、利用することができます。(図1)

2.2 部屋の管理者

管理者はまず、アカウントの作成をします。その後、鍵の解錠用のデバイス(以下 postKey デバイス)の登録、利用者のユーザ発行を行います。あとはシステムが優先度により自動的に部屋を割り振るので、許可/不許可の決定をする必要はありません。

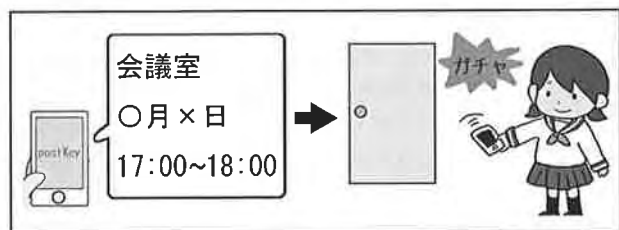


図1 利用者のイメージ図

3. システム構成

postKey は、サーバ、web アプリ、ios アプリ、postKey デバイスにより構成されています。(図2)

管理者は web アプリによりユーザ等の登録・部屋の利用履歴の閲覧をし、利用者は ios アプリにより予約・鍵の解錠を行います。postKey デバイスは、利用者が近づくとスマートフォンと通信し、サーバを介して部屋利用の認証を行います。認証がなされると、postKey デバイスによって部屋の鍵が開きます。

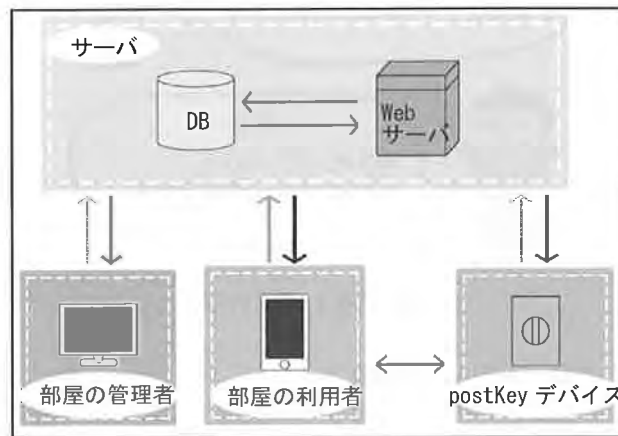


図2 システム構成図

4. 優先度について

postKey では、優先度によって部屋の割り当てを行います。優先度はランク・ポイントから決定されます。

4.1 ランクについて

ランクには教員と学生があり、教員が優先されます。これは、ユーザ発行の際に管理者が設定します。

4.2 ポイントについて

ポイントにはマイナスポイントとプラスポイントがあります。累計利用時間、利用回数が多い場合や、予約をしたのに部屋を利用しなかった場合にマイナスされます。また、1日の初回ログイン時や、あまり使われていない時間帯に部屋を利用するとプラスされます。

ランク・ポイントでの優先度の決定により、皆が公平に部屋を利用することができます。

1. 『STEP APP!!』とは

皆さんは「シニアネット」という NPO 団体をご存知ですか。この団体はシニアが IT を楽しく学ぶことを目的とし、主にシニア向けのパソコン教室を開設しています。しかし、このパソコン教室の受講者や講師はそれぞれ悩みを抱えています(図 1)。

私たちは「分かりやすい資料」は動画であると考え、動画資料を用いた学習と動画資料の作成を支援するアプリケーション『STEP APP!!』を開発しました。



図 1 受講者と講師の悩み

2. 『STEP APP!!』の機能

本アプリケーションは 3 つの機能で、パソコン教室に関わる方々へ支援を行います。

① 動画資料での学習

何度でも見返すことができるので、受講者は自分のペースで学習できます。パソコンの OS やインストールされているソフトウェアのバージョンを登録すると、自分のパソコンに合った動画資料のみを表示します(図 2)。

② 動画資料の作成

講師は講座で教えているときのようにパソコン操作を行います。そのパソコン画面をアプリケーションは自動でキャプチャし、画像を連結して動画を作ります。動画にあわせて説明を入れるだけで、字幕付きの動画が完成します。

③ 円滑な講座進行の補助

講座スケジュールを公開し、受講者に参加の可否を登録してもらうことで、講師は次回の講座に参加する人数を知ることができます。講座で使用する動画資料を関連づけることによって、受講者が予習や復習をしやすくなり、講座を円滑に進めることができます。



図 2 動画資料の再生画面 (開発中)

3. 動画資料の共有

作成された動画資料の共有に、クラウドサービスを利用しています。このためシニアネットが保守を行う必要がなく、負担を減らすことができます。

動画資料はアプリケーションによってクラウド上にアップロードされ、パソコン教室に関わる全ての人のパソコンに共有されます(図 3)。

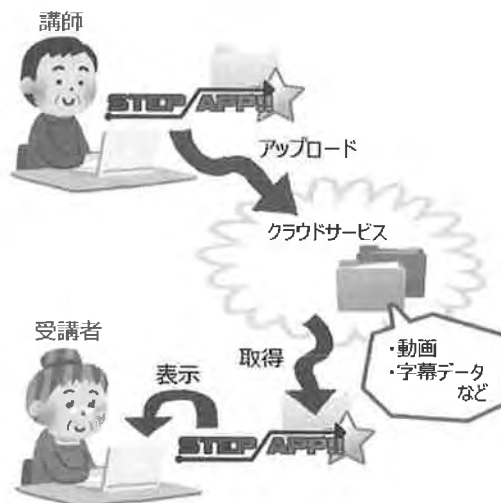


図 3 動画資料の共有

4. まとめ

パソコンは使いこなすと面白いと感じられますが、最初は覚えることが多く、「わたしにはできない」と思われる方も多いです。更に講座進行の遅れが原因で、最後まで説明ができないということも多々あります。

『STEP APP!!』を活用し、自分のペースで学習することで、シニアの方々にも楽しくパソコンを活用していただきたいです。

1. はじめに

編み物、かつてフランスで産業として発展したそれはコースターなどの小物から衣類に至るまで、毛糸1本から作ることができます。使用する毛糸の色や種類、編み方を変えるだけでまた違った作品をつくることもできます。しかし、編み物の設計図「編み図」の見方がわからない、作りたいものの編み図が見つからない、なんてこともよくあります。

そこで私たちは編み図をデータ化し、編集や完成予想図の立体化など、編み物作家さんの作品制作を支援するシステム「あみがみ様」を開発しました。

2. システムの概要

「あみがみ様」は「編み物の初心者から上級者までしっかりサポート」するための機能を搭載します。初心者は編み図から完成イメージを立体化できるビューア機能を、上級者は編み図を新規作成・編集できるエディタ機能を利用することでより良い作品制作が可能になります。

さらに、制作した作品と編み図にコメントをつけてブログ化をサポートするブログ化支援機能も搭載し、自分が作った作品をほかの誰かが作るかもしれません。

また、「あみがみ様」はパソコンだけでなくスマートフォンからもアクセス可能！「あみがみ様」はお出かけ先にもちゃっかり降臨し、支援します。

3. システム構成

図1は「あみがみ様」の構成図です。

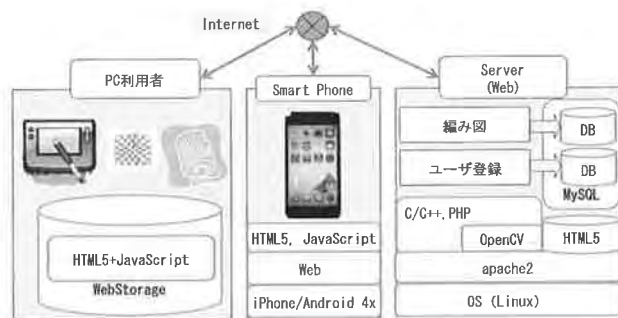


図1 システム構成図



図2 利用画面例

(1) エディタ

エディタ機能は、Webサーバー上でデータベース化することで、糸の種類や編み目などの細かい設定を行います。

(2) ビューア

ビューア機能では正確な三次元図ではなく出来上がりの雰囲気を立て的に表現し、作りたいものの完成時の雰囲気を出来上がりの前に確認します。

(3) マイルーム

マイルーム機能では自分が作った作品を部屋に並べて自由に配置することができます。工夫次第では、森のように統一感のあるルームをつくれます。

(4) Smart Phone

図2は利用画面例です。HTML, PHPを使うことで「あみがみ様」はパソコンにもスマートフォンにも降臨可能にします。

4. おわりに

「あみがみ様」をみて、少しでも「編み物をやってみたい」と思っただけでも幸いです。皆様に「あみがみさま」のご加護がありますように。

1. はじめに

国際宇宙ステーションの船長として若田光一さんが長期滞在する等、近年の日本の宇宙開発には目覚ましい活躍があります。しかし、地球周回軌道上には今非常に厄介なものがあることをご存知でしょうか？これまでの宇宙開発によって生じた宇宙のゴミ「スペースデブリ」です。スペースデブリには人工衛星に衝突する可能性や地上に落ちてくる可能性などの問題があります。また、世間一般に知名度や関心が低いことも問題の1つです。

そこで、我々はスペースデブリの問題を小中学生にも知ってもらうことを目的として、スペースデブリ回収体験システム「CosmoSweeper」を開発しました。

2. システム概要

本システムでは体験者が無人機を遠隔操作してスペースデブリの回収体験をすることができます。回収方法は高速で移動するスペースデブリに近づき、スペースデブリとの相対速度を0にした後、ロボットアームで無人機の中に取り込むことで行います。無人機には燃料による活動制限時間が設定されており、燃料がなくなると動けなくなります。燃料が少なくなってきたら大気圏に突入し、回収したスペースデブリごと燃え尽きることで回収成功となります。一方、燃料がなくなってしまうと、スペースデブリに衝突してしまったりした場合は、回収失敗となります。この場合、無人機は新たなスペースデブリとなってしまいます。開発中の実行画面を図1に示します。

本システムでのスペースデブリの運動や無人機の運動は物理演算モデルを使用して計算しています。また、ロボットアームは国際宇宙ステーション日本モジュール(きぼう)で使用されているロボットアーム^{[1][2]}を参考にしています。このように、本システムは実際に宇宙空間でスペースデブリを回収することを再現しています。

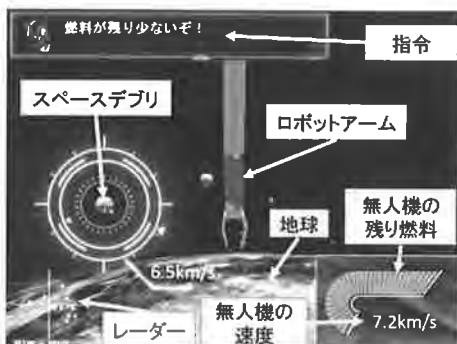


図1 実行画面

3. システム構成

システム構成を図2に示します。

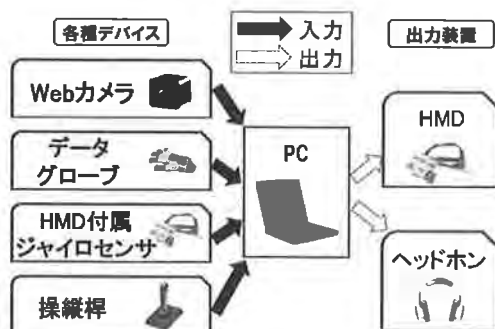


図2 システム構成

- ・Web カメラは、体験者の正面に設置され、手の座標を画像処理により検出します。マニュアルモードのロボットアーム操作に使用します。手の動きに連動してロボットアームが動きます。
- ・HMD 付属ジャイロセンサは、体験者の頭部の傾きを検出します。セミオートモードでロボットアームの操作に使用します。頭部の傾きに連動してロボットアームが動きます。
- ・データグローブは、体験者が装着し、手を握っている状態、開いている状態を検出します。スペースデブリを掴む操作に使用します。
- ・HMD(ヘッドマウントディスプレイ)は、無人機から見た地球周回軌道上の映像を表示します。
- ・操縦桿デバイスは、無人機の操作に使用します。無人機の加減速や移動、視線の変更で使用します。
- ・ヘッドホンは、「人工衛星と衝突しそうなスペースデブリを回収せよ」などの指令を音声出力します。

4. 難易度選択

本システムには難易度の異なる2つのモードがあります。

- ・マニュアルモード
中学生以上を対象とする、リアルな操作で回収体験を行うモードです。このモードでは複数のデバイスを同時に使用します。無人機の操作を操縦桿デバイスで行い、ロボットアームの操作は手を動かすことで行います。
- ・セミオートモード
小学生以下を対象とする、操作を容易にしたモードです。マニュアルモードよりも簡単な操作で回収体験を行えます。無人機の操作はコンピュータが自動で行い、アームの操作は体験者の頭部を動かすことで行います。

5. システム体験の手順

- 1)小学生以下か中学生以上を選びます。
- 2)セミオートモードかマニュアルモードを選びます。
- 3)名前を入力を行います。
- 4)チュートリアルでスペースデブリについての説明と回収体験の操作方法を覚えます。
- 5)地球周回軌道上のスペースデブリ回収体験が始まります。回収体験中には、人工衛星と衝突しそうなスペースデブリの回収などの指令が下されます。
- 6)燃料がなくなる前に大気圏に突入しスペースデブリごと燃え尽きます。これで回収体験は終了です。
- 7)回収体験の結果が表示されます。回収したスペースデブリの数と残りのスペースデブリの数をすることができます。

6. おわりに

スペースデブリは、宇宙開発にとって大きな、かつ、待ったなしの問題です。現在、スペースデブリの回収方法は確立されておらず、まだ研究中です^[2]。その中で我々はJAXAで回収方法の1つとして提案されている構想^[1]を参考にしながら、実現方法を独自に考え本システムを開発しました。本システムをきっかけにスペースデブリの問題について少しでも関心を持っていただけたら幸いです。そして、その子供達が将来、宇宙技術者として活躍して欲しいと思います。

7. 参考文献

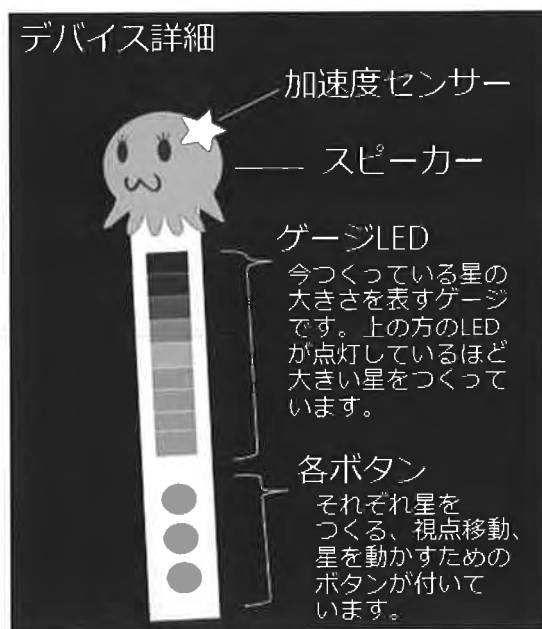
- [1]EVA支援ロボット実証実験(REX-J)の打上げと運用, URL:http://iss.jaxa.jp/kibo/library/press/data/130227rexj-press-lec.pdf(2014.9.4現在)
[2]宇宙博2014-NASA・JAXAの挑戦-, NHK・NHKプロモーション・朝日新聞社発行(2014)

1. 宇宙を創ろう「cosmo craft」

みなさんは宇宙旅行をしてみたいとか、魔法を使うように星を動かして遊んでみたいと考えたことはありませんか？私たちは、そんな夢のような話をプラネタリウムや天体観測などと違う形で楽しむことが出来る宇宙体感型システム「cosmo craft」を作成しました。

2. コスモちゃんステッキ

このシステムには、魔法の杖「コスモちゃんステッキ」が存在します。この「コスモちゃんステッキ」にはマスコットキャラクターのコスモちゃんが付いていて、コスモちゃんがあなたのオリジナルの宇宙を創ることを手伝ってくれます。ボタンを押してコスモちゃんに星を作ってもらったり、ワープさせてくれたりします。



3. 遊び方

3.1 星をつくろう

ステージの中でコスモちゃんステッキのボタンを押し続けると部屋の中に色や大きさの異なる星を創ることができます。好きな場所に好きなように星を作ってみましょう。ここで、生成される星は現実の星と同じく、生まれて間もない星の色は青く、そこから時間がたつにつれて、白、赤へと変化していくでしょう。

3.2 星を打ち上げよう

3.1 で作った星は、コスモちゃんステッキをふって打ち上げることができます。打ち上げられた星はステッキのボタンを押した場所に応じてプロジェクターで天井に投影されます。星はほかの星にぶつかったり、近づいたりすることで万有引力などの様々な物理現象を起こします。自分の好きな場所に星を打ち上げてみましょう。

3.3 星を動かしてみよう

星を打ち上げた宇宙空間にある星を自由自在に動かすことができます。それによって私たちは流星を降らせ、星を衝突させ反応を起こし星空を好きなようにデザインすることができます。星の衝突で起こる美しい輝きはまるできらきらとした宝石のように私たちを魅了するでしょう。そしてこれらの星の現象は星空のデザインだけでなく新しい学習スタイルを提供することにもつながります。



生成できる宇宙の例

4. おわりに

みなさんも「cosmo craft」で自分なりの宇宙を創ってみませんか？

1. 「はなまるフォーム」の概要

趣味や部活動でスポーツをしている人のなかで、「自分のフォームを見直したい」、「あの人の動きに近づきたい」と思ったことがある人は多いのではないのでしょうか？自分のフォームを知ることはスポーツでとても大切なことです。そこで私たちはタブレットを用いて、誰にでも使えるスポーツ支援アプリ「はなまるフォーム」を開発しました。

「はなまるフォーム」には2つの機能があります。それは、自分の数秒前の動きをチェックできる「ふりかえり機能」(図1)と、撮影された上級者の動きを真似して練習できる「お手本機能」です(図2)。



図1 ふりかえり機能



図2 お手本機能

2. 「はなまるフォーム」の使い方

2.1 ふりかえり機能

「ふりかえり機能」では短時間のリプレイを表示することで簡単に自分の動きを見ることができます。遅延させる時間は画面の下にあるスライダーによりいつでも設定することができます(図3)。

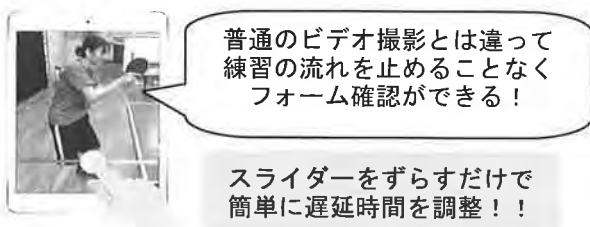


図3 ふりかえり機能の操作画面

2.2 お手本機能

「お手本機能」では上級者を見本として撮影し、画面上でその動きを自分の動きと重ね合わせることができます。これにより、お手本と自分の動きの違いをタブレットで確認しながら繰り返し練習することができます(図4)。

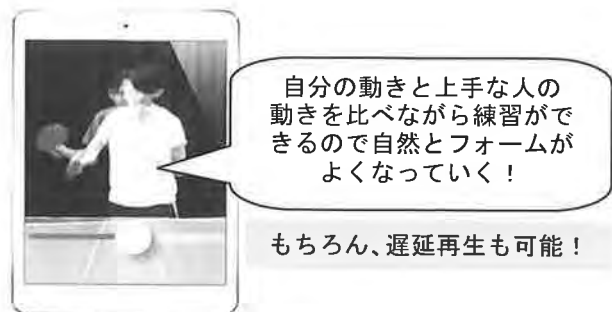


図4 お手本機能の操作画面

3. おわりに

私たちは「はなまるフォーム」を卓球部、バスケ部、少林寺拳法部、バレー部などで実際に使ってもらいました(図5)。「これなら効率よくフォームを確認できる!」、「新しい練習方法で選手のレベルアップができそう」などの反響をいただきました。

「はなまるフォーム」があればビデオカメラで撮影して確認なんて作業はいりません! 「はなまるフォーム」を使ってあなたのスポーツライフに花を咲かせてみませんか?



図5 動画確認の様子

1. はじめに

今日、プログラミング学習の必要性が叫ばれている。しかし、初学者の学習環境は充実しているとはいえない。学校のPCの利用には場所・時間の制約があり、自宅での環境構築には機材と専門知識が求められる。また教師側では問題生成と採点に時間と手間がかかる。

そこで私達は、利用が容易なタブレット端末を用いたプログラミング学習環境と、問題の作成・採点の負担を軽減するシステムを提案する。

2. システムの概要

2.1 コンセプト

場所・時間の制約を受けないタブレット上で動作するJavaプログラミング学習環境と、問題作成・採点機能を持つLMS環境を提供する。

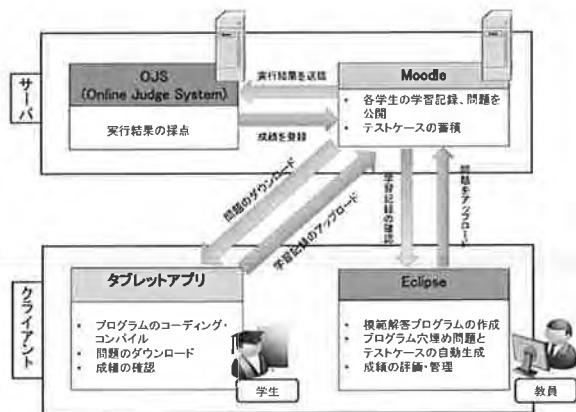


図1. システム構成図

2.2 各システムの機能

2.2.1 問題の自動生成

教員の問題作成を容易にするために、完成したプログラムソースから穴埋め問題・テストケースを自動生成するEclipseプラグインを作成した。

穴埋め問題・テストケースの作成は、完成したプログラムソースの構文解析によって実現している。

2.2.2 LMSサーバ(Moodle)による問題・成績の管理

問題と成績の公開・管理を行うためにLMSサーバを使用する。学生はいつでも学習したい課題をダウンロード

でき、成績確認も可能である。LMSは多くの高専で利用されているMoodleを採用、専用プラグインを作成した。

2.2.3 Java IDE アプリ

場所・時間の制約が無いJavaプログラミング環境を提供するために、タブレット上のJava IDEを開発した。

本アプリは、従来のアプリに前例のない**端末内でのコンパイル・実行を実現**させている。

エディタ部では文法エラーチェック、シンタックスハイライト、コード補完を実装している。また、キーボードがないタブレットの操作性を改善するため、カーソルキー移動とズーム機能を加えた。

2.2.4 OJS (Online Judge System)

問題の採点を自動化するために、OJSを作成した。OJSとは問題の自動採点を行うシステムである。LMSから送られたJavaソースを自動で実行し、テストケースの評価によって採点を実現している。

3. 利用者からのフィードバック

本システムを実際に利用してもらい、学生が抱える

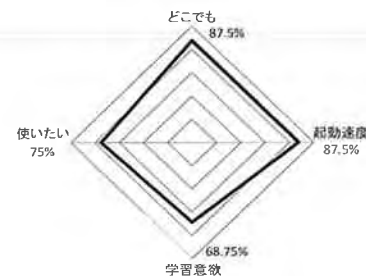


図2. 学生のアンケート結果

問題を解決できているかアンケート調査を行った。学習環境の改善に繋がっていることが確認された。

4. まとめ

本システムによって、学生は時間・場所に制限されないJavaプログラミング学習をすることが可能に、教員は問題の作成・採点が可能になった。

本システムは学生の自習はもちろん授業にも活用できる。PC教室に固定されていたプログラミング学習もタブレットの活用によって場所の制約から解放される。今後は高専を超えてプログラミングの輪の拡大を図っていききたい。

1 はじめに

「サーバのチューニング」は「面倒」「よく分からない」ものです。私たちは、そうしたサーバチューニングを自動で行うシステム「Rapid House」を開発しました。

本システムを用いて、「Apache HTTP Server」のチューニングを行った結果、デフォルト設定と比較して約 12.5 倍、手動チューニングとの比較でも約 9.5 倍もの性能を引き出せることを確認できました。

さらに、従来の類似システムの難点であったチューニング時間について、本システムでは、類似した環境の成果を利用することで時間の短縮を行うことができます。

2 本システムの構成

構成図を図 1 に示し、構成要素を紹介します。



図1 システム構成図

2.1 対象サーバ

サーバソフトウェアが実行され、本システムのチューニング対象となるサーバです。SSH(Secure SHell)で操作できる必要があります。

2.2 管理 PC

対象サーバへ SSH 接続する PC です。対象サーバでも自身に SSH 接続できれば、自身が管理 PC になります。

2.3 情報データベース

「こうした環境で、こうしたパラメータで、こうしたスコアが出た」という情報を持ちます。管理 PC は、対象サーバのこれらの情報を取得し、チューニング中に送信します。なお、環境は「CPU 情報(メーカー名, 製品名, アーキテクチャ, 周波数, コア数)」「メモリ容量」「ソフトウェア名」「評価に用いたコマンド」で構成されます。

3 本システムの機能

本システムの機能を紹介します。

3.1 サーバ自動チューニング機能

チューニングを最適化問題とし、山登り法(HC)、遺伝的アルゴリズム(GA)などの最適化アルゴリズムで、サーバのチューニングを自動で行う機能です。

図 2 は、Apache HTTP Server でのチューニング結果で、「GA での各世代最良スコア」、「事前に 3 名により手動チューニングされた後のスコア」と「デフォルト設定のスコア」であり、それぞれは同一構成の仮想マシンで独立させています。また、縦軸の「Scores」は ab コマンドの実行結果「Requests per second」の値で、大きい値が良く、横軸の「Time」は、GA での一世代の処理時間です。

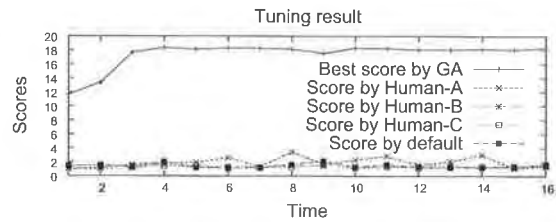


図2 チューニング結果

図 2 より、GA では時間とともにスコアが上がっていることが分かります。このとき、「事前に 3 名により手動チューニングされた後のスコア」の各自の平均の最大は 1.93 で、「デフォルト設定のスコア」の平均が 1.46 でした。これらと、GA の第 16 時刻(世代)での最良スコア 18.3 を比較すると、デフォルト設定では約 12.5 倍、手動チューニングでは約 9.5 倍の性能改善が行えたことが分かります。

3.2 チューニング時間短縮機能

最適化アルゴリズムの初期解、またはその一部を情報データベースから取得した「対象サーバと類似する環境で高いスコアが得られた際のパラメータ」とする機能です。

この「類似する環境」とは、情報データベースの「環境」を要素として持つベクトル x, y において、ユークリッド距離を基にし、独自に設定した「どの違いを重視するか」の重み w を考慮した距離

$$d(x, y, w) = \sqrt{\sum_{i=0}^n w_i (x_i - y_i)^2}$$

が小さいものを指します。なお、文字列の差は「一致の場合は 0」「それ以外は 1」として処理しています。

図 3 は、図 2 と同じ構成で、情報データベースを利用した場合のチューニング結果です。図 3 を図 2 と比較する

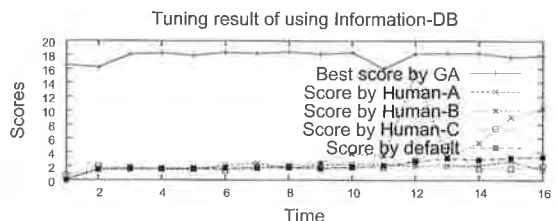


図3 情報データベースを利用したチューニング結果

と、少ない時間で良いスコアが得られており、チューニング時間の短縮が行えたことが分かります。

4 おわりに

図 2 より、デフォルト設定と比較し約 12.5 倍、手動チューニングとの比較でも約 9.5 倍もの性能を引き出せることを確認し、図 3 では、チューニング時間の短縮も行えていることから、本システムは有用です。

「面倒」「よく分からない」で、無駄に電力を消費し、地球を暖めているサーバに貢献できればと存じます。

1. ゲーム型観光サポートアプリ

みなさんは観光をする際どうやって観光名所を探していますか？旅行のプランをどのように決めていますか？どのようにして、目的地までたどり着きますか？いざ、ネットやアプリで調べようと思っても情報が多すぎて分かりづらいものです。そんな悩みを楽しみながら簡単に解消できる便利な Android アプリが「すごろく観光」です。

2. すごろく観光の機能

2.1 概要

すごろく観光に備わっている機能は大きく分けて 4 つあります。まず 1 つ目は「すごろくゲーム」です。すごろくゲームはこのアプリを構成する重要な機能で、ユーザーはすごろくを楽しみながら旅行先の観光情報を調査できます。主に旅行前の情報収集に使えます。2 つ目は「プラン作成」です。自分にあった旅行プランを作成できます。3 つ目の「経路探索」は旅行先で必ず行う現在地から観光地への行き方を調べることができます。4 つ目は「疑似体験」です。この機能は旅行に行きたいが、忙しくて行くことができない人のために、旅行を疑似的に再現します。

2.2 すごろくゲーム

すごろくゲームは、旅行先の地図を利用して、実際の観光情報などを知っていただくことができます。観光名所に対応したすごろくのマスをクリックすることで、名所の観光情報などがわかります。

2.3 プラン作成

プレイヤーは「実際にこの名所に行きたいかも！」と思う名所を追加できる、行きたいかもリストを所持しています。すごろくゲーム上で、名所に対応したマ

スをタップすると名所のメニューが開かれ、行きたいかもリストへ追加することができます。ユーザーの行きたいと思った場所をリストに追加することでユーザー独自のプランを計画することができます。

2.4 経路探索

名所のマスから呼び出せるメニューから経路探索を選択すると Google Map が呼び出され、自分が今いる場所から名所までのルートがわかります。

2.5 疑似体験

名所のマスから呼び出せるメニューから疑似体験を選択することができます。疑似体験を選択すると、Google Map のストリートビューが呼び出され、名所を疑似的に旅行体験できます。

3. システム構成

メニュー画面からすごろくゲームが選ぶことができます。すごろくゲームでは経路探索や疑似体験、プラン作成を選ぶことができ、経路探索では Google Map、疑似体験ではストリートビューがそれぞれ呼び出されます。作成したプランは保存して、後で見返すことができます。

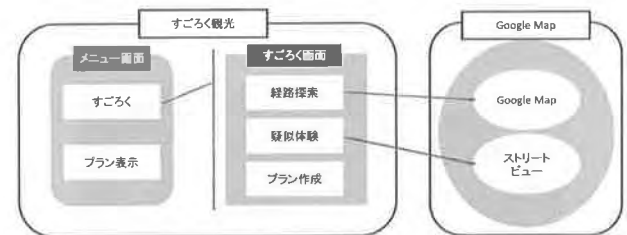


図 1 システム構成図

4. 実行環境

Android OS Ver2.1 以上の Android 端末

5. おわりに

「すごろく観光」でみなさんの旅行がもっと楽しくなることを願っています！！

1. 「AR 量子将棋」とは？

みなさんは量子将棋というゲームを御存じでしょうか？量子将棋とは、最初は全ての駒があらゆる種類の駒である可能性を持っており、駒を動かすたびにその行動からあり得ない駒の可能性を削除していき、最終的に相手の王将を確定させてその王将をとるゲームです。インターネット上の「将棋ったーβ」^[1]でプレイすることが出来ます。このゲームは膨大な情報量が必要なため、今のところパソコンを使用しないとプレイすることが困難です。そこでは私たちは拡張現実（以下ARと省略）を使い実際のボードで量子将棋をプレイできるシステムを開発しました。

2. 機能説明

2.1 駒情報の視覚化

図1はARを使った量子将棋のイメージです。駒の情報を視覚化する方法として、ARマーカーとARToolKitを使用します。ARマーカーとは、図にある駒の上に張られたモザイク状のパターン画像をカメラにかざすことによってマーカーの画像パターンを認識し、既存の画像パターンと照らし合わせ、対応する画像をマーカーのある位置にある画像に重ねて表示することができる技術です。本システムはこの駒にARマーカーを貼り付けることで各駒の情報を投影します。



図1. ARを使用した量子将棋のイメージ

2.2 プレイした棋譜の作成

図2は棋譜作成システムの構成です。駒が1手移動

するたびに盤面の画像を取得、ARtoolkitを用いてARマーカーを検出して座標を出力します。

座標情報を比較し、その値が異なれば「その駒が移動した」と判定し、「棋譜」データへ記録されます。これにより棋譜には駒の移動履歴が残ります。

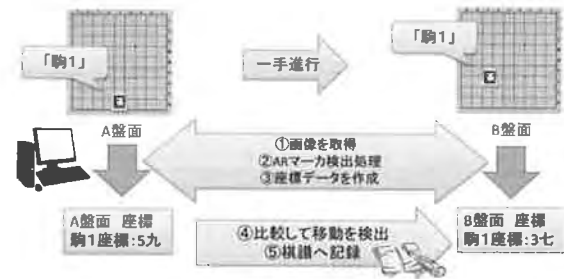


図2. 棋譜作成の構成図

3. システム構成

WebカメラとARマーカーを使ってその駒が何番の駒であるかを判別します。また、その駒の動きと、その駒の可能性をパソコン上のソフトウェアで処理していきます。「AR 量子将棋」はこのように構成されます。図3はシステム構成図です。

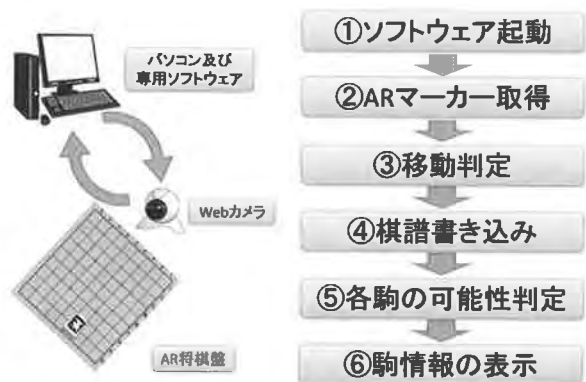


図3. システム構成図

4. おわりに

みなさんも、より戦略性の高い将棋を手軽にプレイしてみませんか？

[1]将棋ったーβ (2014. 8. 29 検索)

<http://shogitter.com/>

1. はじめに

近年、消費税の増税により、物品の値段が上昇しました。これに伴って、値札を税抜表示とする商店が増加し、これまでの値段とのギャップから、「買いすぎてしまった!」「予想よりも高い…」「細かい値段が増えて家計簿をつけるのが大変…」などのお金を多く使いすぎてしまうという事例が多く見られます。そこで、日常生活におけるこれらの問題を是正するべく買い物支援アプリケーション「ポケットレジスタ」を開発しました。

2. システム概要

本システムでは Android OS を搭載した携帯端末を対象とし、4つのステップで構成されています。システムの流れを図1に示します。

- ① カメラで商品の価格を読み取る
- ② 価格を税率・割引率にかけて保存
- ③ 指示を出すと保存された価格を合算して表示
- ④ 家計簿として出力する

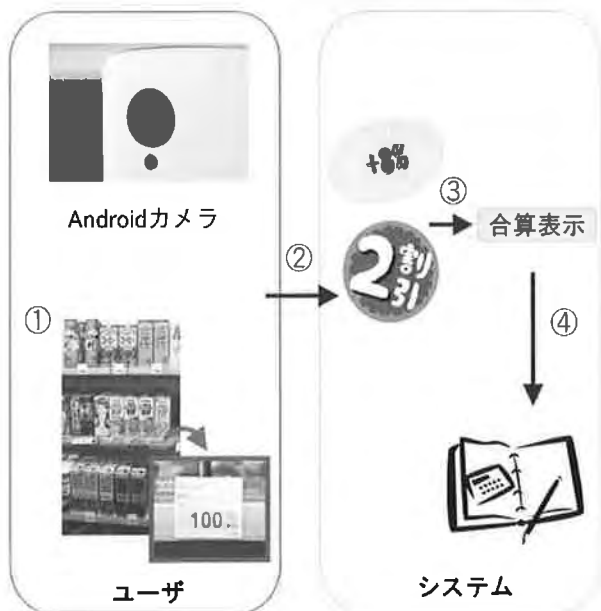


図1. 携帯端末による読み取り・計算

3. 機能説明

3.1 税込価格自動表示機能

商品の値札に携帯端末のカメラをかざすことで、価格を読み取り、税込み・割引価格として表示します。このとき、商店によって税込、税抜のどちらで表示されているか異なるので、それらを切り替えられるようになっています。この値を保持、加算を行い、レジを通る前にお買い物の小計を知ることができます。

3.2 家計簿機能

お買い物をした日付と小計を保存しておき、いつ、どのくらいの買い物をしたかを確認することができます。また、そのデータをPCで管理できるようなファイル形式で出力することもできます。この機能で、家計簿をつける手間を省き、主婦(夫)の方々の負担を軽減し、その他の家事や趣味に充てる時間の増加を図ります。

4. 既存のアプリケーションとの比較

本システムはお買い物中には小計を、後には家計簿を出力し、お買い物における全てのシーンにおいてユーザを支援することができます。「お買い物」という日常生活で多くの比率を占めることに重きを置き、このアプリケーション一つで完結できるようになっています。

5. おわりに

本システムによって、大人には無駄づかいの削減やフリーな時間の増加を、子どもにはお手伝いの機会を提供し、明るい家庭を形作る一つの要因となれば幸いです。

1. はじめに

Twitter など SNS の利用者は爆発的に増えており、それに伴い不適切な投稿内容や写真が第三者により発見・拡散され、炎上する事件（通称“バカッター”等）が増えている。炎上の発端となった投稿内容がマスコミで報道されれば、企業イメージ悪化に伴う損害賠償請求など、取返しのつかない事態に発展してしまう。

図1に典型的な炎上プロセスを示す。不適切な投稿等に対して、投稿から30分ほどで第三者が批判コメントを投稿し、2時間でネット上に投稿が拡散する。2日以内に個人情報が特定され、それ以降、メディアで報道され、大事件化することも多い。

そこで、SNSでの炎上被害などを減らし、安全に利用するためのツール「SNS 炎上報知器」を提案する。

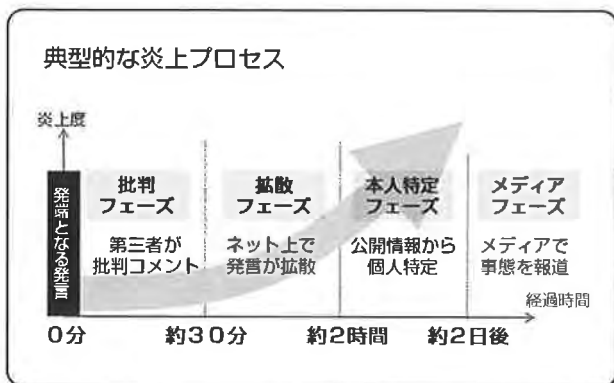


図1 典型的な炎上プロセス

2. システム概要 (図2)

「SNS 炎上検知器」は、Android スマートフォン上で使用し、投稿内容やリツイート・リプライ数を監視することでユーザーに“気づき”の機会を与え、炎上を事前に予防又は対策できるようにする。図1の炎上の各フェーズに対応し、次の機能を開発した。

2.1 批判フェーズ (NGワード・個人情報の検出)

投稿直後30分以内の批判フェーズでは、投稿内容に犯罪予告や誹謗中傷などのNGワードや個人情報が含まれていないか、を検出する。投稿内容に対してリアルタイムに検出を行い、ユーザーに削除・謝罪を促す

ことで、拡散フェーズへの移行を防ぐ。

2.2 拡散フェーズ (炎上感知)

拡散フェーズでは、Twitter や Facebook のリツイート、リプライ、いいね、シェアの一定時間内の増加数をチェックし、異常に増加している場合には炎上の可能性を警告する。

2.3 本人特定フェーズ (事前予防)

本人特定フェーズに移行してしまうと、個人の特定は避けられない。そこで事前予防として定期的にプロフィール等の公開設定をチェックし設定の不備に対して、ユーザーに“気づき”を与える。また一定期間内のTwitter と Facebook の投稿内容から個人情報を特定できる可能性を統計処理・特徴抽出により数値化する。



図2 システム概要

3. 将来のビジネスプラン

将来のビジネスプランとしてはフリーミアム方式を検討している。「SNS 炎上検知器」は一定のユーザー数と評価を獲得するため、一般ユーザーには無償提供する。法人向けには社員や所属タレントの投稿をサーバで一括管理する機能を提供することで収益化を目指す。

4. まとめ

「SNS 炎上検知器」はAndroid スマートフォン上で、投稿内容や公開設定等をリアルタイムに監視し、ユーザーに“気づき”の機会を与えることで炎上を予防又は対策する。将来は法人向けの機能を追加し、収益化を図りたい。

1. はじめに

リモコンは手元で家電を操作することができ便利なものですが、家電の多様化と共にリモコンが家庭内に増え、操作したい家電のリモコンが見つからなかったり、たくさんのリモコンの中からどれか分からなくなる問題が発生します。これらの問題を解決するために学習リモコンやマルチリモコンが製品化されていますが、設定が面倒、操作が煩雑等の問題があり殆ど普及していません。そこで、私達は画像認識技術と AR 技術を応用し、全てのリモコンを一つにまとめ、使いやすさを追求したリモコン「Remoconia」を提案します。

2. システム概要

2-1. Remoconia アプリ

操作したい家電にカメラを向けると、その家電用のリモコンになるスマートフォンアプリです。ユーザーの操作は「Remoconia モジュール」と「Remoconia コンセント」に Bluetooth を使用して指示します。

2-2. Remoconia モジュール

「Remoconia アプリ」からの指示にしたがって、操作する家電の種類やメーカーに対応した赤外線リモコン信号を送信するモジュールです。

2-3. Remoconia コンセント

「Remoconia アプリ」からの指示にしたがって、家電への電力の供給を制御するモジュールです。また、消費電力を測定する機能も備えています。

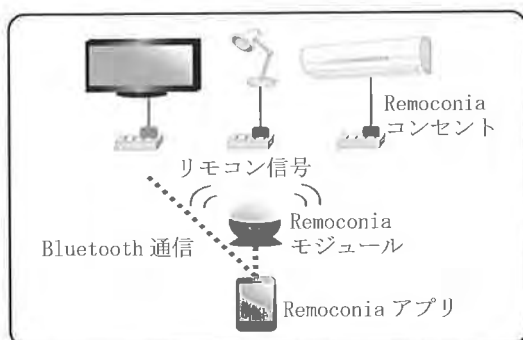


図 1. システム構成図

3. Remoconia の機能

3-1. 簡単操作で家電を登録

操作したい家電にスマートフォンを向け家電とメーカーロゴを登録します。その後は操作したい家電にスマートフォンを向けるだけで「Remoconia アプリ」が自動で家電を認識し、画面に専用のリモコンボタンが表示され家電を操作することが可能になります。

この家電認識を実現するため OpenCV ライブラリを使用しました。家電の種類と形状、メーカーロゴは形状マッチング・テンプレートマッチングで判別し登録します。

3-2. 操作したい家電にスマホを向けるだけの簡単操作で専用リモコンを実現

スマホのカメラを操作したい家電に向けると、カメラから取り込んだ画像と登録した画像データを用いて特徴量マッチング・形状マッチングを行い操作対象となる家電を認識し、家電毎に最適化したリモコンを重ねて表示することによって簡単に操作ができます。



図 2. TV リモコン画面



図 3. 卓上照明リモコン画面

3-3. リモコンに対応していない家電も操作可能

赤外線リモコンに対応していない家電でも、「Remoconia コンセント」を用いることで ON/OFF 機能やタイマー機能が使えるようになります。

3-4. 消費電力の表示

「Remoconia コンセント」で計測した家電の消費電力がスマートフォンの画面越しにリアルタイムに表示されます。

4. おわりに

Remoconia を使うことで家庭にある多くのリモコンは一つに集約され、「一つの家庭に一つのリモコン」が実現します。

1. はじめに

近年、地球温暖化による気温上昇に伴って増加傾向にある熱中症。今年の夏も熱中症で病院に緊急搬送される人が多い年となりました。多い時では一週間におよそ 5200 人もの方が搬送されています。

そこで、熱中症の初期症状を検知し、熱中症が軽度うちに注意を促すシステム「ネッチュー」を開発しました。

2. システム概要

「ネッチュー」はウェアラブル型センサ「ネッチューバンド」を身につけ、Android アプリを起動するだけで使用できます。「ネッチューバンド」は使用者のバイタルデータに異変がある、または Android 端末からの操作があった時のみ、Android 端末と通信し、使用者のバイタル情報を Android 端末に送信します。

受信されたデータを基に独自の判定基準によって、熱中症か発症しているかどうかを判断します。また異常と判定された時、アプリに登録されている相手に現在の状態をメールによって通知し異常を知らせます。

2.1 ウェアラブル端末「ネッチューバンド」について

「ネッチューバンド」は呼吸センサ、体温センサ、脈拍センサ、温度センサ、湿度センサの五つのセンサで構成される端末で通信モジュールとして Bluetooth を使用しています。「ネッチューバンド」はいずれかのセンサが検知した値が独自の基準値を上回った時や、アプリから要求があった時に Android 端末にバイタルデータを送信する機能だけが搭載されています。

2.2 Android アプリについて

アプリには私達が熱中症の状態と判断した複数のデータを基に熱中症であるかを判別する基準が設けられています。受信されたデータをその基準と照らし合わ

せて、熱中症が発症しているかを判別します。熱中症と判別された場合、あらかじめ登録している人にメールで情報を送信します。また、画面には熱中症が発症した時の対応マニュアルが表示されます。

3. 「ネッチュー」のシステム構成

「ネッチュー」はウェアラブル型センサ「ネッチューバンド」と Android 端末によって構成されます。「ネッチューバンド」と Android 端末は Bluetooth で接続されるため、ある程度の距離であれば Android 端末を身につけていなくても使用できます。

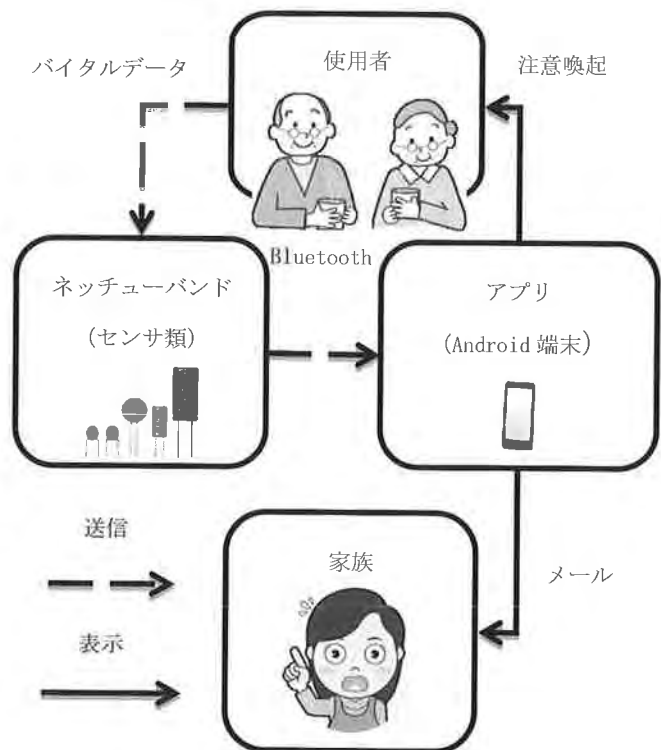


図1 ネッチューのシステム構成図

4. おわりに

今後、増加し続ける熱中症に対して、予防を呼びかけるアプリ、製品は無数にあります。しかし、それでも多くの方が熱中症で搬送されています。しかし、使用者に体の状態を正確に伝える「ネッチュー」では今まで救えなかった熱中症患者を救うことが可能になります。

21

(未定)

ペトロナス
工科大学

Mohamad Ali Hajian Maleki
Muhahmmad Fadzil Mohd Jusri
Ahmad Izuddin Zainal Abidin(教員)

応募全作品

■ 課題部門「防災・減災対策と復興支援」

登録順	タイトル	高専名	指導教員	参加学生
1	HiTo避難所	小山	小林 康浩	佐藤 光, 野中 隆寛, 佐藤 太一, 田崎 洋介, 吉川 健太
2	PoP —災害時被災者情報管理システム—	米子	河野 清尊	大野 貴昭, 嶋本 智矩, 中村紗也佳, 西尾 泰希
3	たたまるハウス	小山	小林 康浩	小久保 隆史, 六本 大志, 中山 和, 中田 雄哉
4	いしきはこぶね —地域で繋げる避難場所案内システム—	鳥羽商船	中井 一文	山本 龍也, 高嶋 大和, 杉野 寿輝, 中野 友貴, 高嶋 峻希
5	人(ヒート)マップ —みんなで作る情報マップ—	鳥羽商船	江崎 修央	伊藤 由紀, 竹口 優里, 谷口 昂汰, 萩原 海仁, 濱口 堅太
6	よっとなび	有明	松野 良信	都松 嘉恋, 井上 志莉子, 竹永 拓海, 松尾 竜世, 古川 拓実
7	Relief Supplies Relay	鈴鹿	浦尾 彰	伊藤 貴哉, 紀平 将史, 早川 栄作, 道上 将志, 櫻井 真子
8	Family Connection Map	茨城	滝沢 陽三	小林 優太, 和田 太一
9	Grouper —集まりを つなぐに—	津山	宮下 卓也	中村 悠生, 須和田 与春, 萩原 涼介, 綾部 敬祐
10	救活!	松江	渡部 徹	廣戸 勇樹, 川上 直人, 福井 悠太
11	緊急災害ファミリーネット	呉	藤井 敏則	市川 晃太朗, 赤羽 真琴, 島本 知輝, 辻井 雄大, 竹山 哲哉
12	津波避難エキスパート	金沢	伊藤 周	澤田 友樹, 北 直樹, 中嶋 悠也, 中山 義崇, 長井 健太
13	DCPA —減災総合支援システム—	有明	松野 良信	待鳥 雄哉, 國松 安一郎, 池村 太郎, 大河 亮, 山田 達郎
14	絆 —情報をより迅速に—	都城	小林 洋介	川崎 浩史, 長友 和也, 清家 竜司
15	BlockSnow —ホワイトアウト回避アプリ—	旭川	森川 一	板坂 優人, 山口 凌, 佐藤 広基
16	i-BadgeOVERしまNET —子供見守り防災システム—	弓削商船	田房 友典	檜垣 俊希, 岡野 さくら, 島本 知輝, 福羅 重利沙, 村上 麻矢加
17	光陰如箭 —A ray of hope—	弓削商船	長尾 和彦	宇崎 裕太, 瀬尾 敦生, 肥田 琢弥, 山本 愛奈
18	Life-LINE —繋ぐ, 繋がる, 繋げ合う—	福井	高久 有一	民谷 慎一郎, 廣島 健亮, 中内 文也
19	WT —つなぐ・つながる パケットを遠くまで届け隊!!!—	福井	斉藤 徹	野村 弘樹, 中後 和希, 野村 信吾
20	Man-Hold —町を守る近未来マンホール—	明石	新井 イスマイル	森 茂史, 森 恵, 西原 大貴, 萩野 秀祐
21	IRIS —災害情報管理システム—	仙台(広瀬)	園田 潤	橋 健陽, 大沼 峻徳, 飯野 直弥, 相沢 幸翼, 小野 幸仁
22	Change —希望が育つ願いの樹—	松江	和田守美穂	今若 悠樹, 吉田 悠花, 岩崎 未来, 古志 俊菜, 土井 一磨
23	マップXD —マップでつなぐ地域コミュニケーション—	一関	小保方幸次	鬼柳 元樹, 佐藤 健太, 下田 将之, 千葉 大輝, 及川 達
24	KImaP —what on earth?—	香川(高松)	重田 和弘	川崎 恭輔, 惠谷 裕香, 十河 慈章, 木村 和寛, 石川 侑扶
25	潮エネ!“安” —スゴいぞ潮流発電・3つの安で世界を変える—	大島商船	北風 裕教	重本 昌也, 村上 秀隆, 柿本 福実, 中野 裕次, BATNASAN, BAASANCHIULUN
26	MAP FIXER —災害時に役立つ町内道路地図アプリ—	熊本(八代)	小島 俊輔	木村 匠, 関原 至音, 湯母 武龍, 岩下 将大, 松田 和輝
27	震災初期避難誘導システム「円盤エリオン」	明石	新井 イスマイル	岩波 慶一朗, 松田 裕貴, 藤 健介, 石岡 匠也, 斉藤 直矢
28	だいわくしょん! —双方向型電子通信システム—	一関	千田 栄幸	村山 福太, 和田 一真, 三宅 知里, 門下 佳樹, 浦郷 亮仁
29	手軽に簡単防災シミュレータ —視覚映像による防災支援システム—	金沢	谷口 萌未	亀谷 大貴, 若井 尊, 梶田 龍生, 直江 一輝, 間瀬 健介
30	スレッチ —避難誘導・行方不明者捜索支援システム—	香川(高松)	村上 幸一	十河 洋大, 平尾 悟, 大森 裕二, 清水 健作, 橋本 賢人
31	つながりタワー —津波避難タワー間を結ぶ安心防災システム—	高知	今井 一雅	佐々木 渉, 島内 良章, 南 光成, 森國 健吾
32	Taskatter —生きてる なう—	鈴鹿	青山 俊弘	石河 純輝, 武田 源生, 川戸 美輝, 川瀬 勇輝, 吉田 淳
33	SALAD —ともにつながる・助けあえる—	岐阜	廣瀬 康之	加納 英樹, 西中 智樹, 伏木ももこ, 塚本 暢弥
34	こことおれーる	和歌山	村田 充利	上野山 大介, 吉屋 孝基, 森 健恭, 和田 賞登, 夏見 勇矢
35	NODE-CONNECTOR —繋ぐローカル災害MAPネット—	神戸市立	朝倉 義裕	宗和 恭平, 田上 空良, 猪之間賢児
36	和。	福島	島村 浩	木村 峻佑, 鈴木 侑斗, 村尾 直柔
37	Vo-Learning —ネットで学ぶボランティア—	石川	小村良太郎	山岸 徹平, 寺西 郁, 坂本 成寛, 宮崎 航輔
38	三次元災害シミュレーター	釧路	神谷 昭基	一條 瑞澄, 齋藤 佑樹, 庄司 怜那, 新畑 幸佑, 佐藤 奎太
39	BBB —非常時の非インターネット環境での情報交換システム—	仙台(広瀬)	穂坂 紀子	高橋 況一, 及川 達希, 東海 佳祐, 後藤 隆太, 齋藤 翼

■ 課題部門「防災・減災対策と復興支援」

登録順	タイトル	高専名	指導教員	参加学生
40	BREAKERS	阿 南	吉田 晋	篠原 諒, 藤原 大輝, 鷺津 京, 谷口 恭, 酒井 和也
41	トリセツ —採りたくなる説明書—	香川(詫間)	宮武 明義	大野健太郎, 藤原 周平, 樋口 祐太, 泉保 孝樹
42	SACK@ジャパン —非常用コンピュータ持出ナップザック—	米 子	松本 正己	山根 大典, 林原 真史
43	すりーくSupport —物資補給支援システム—	舞 鶴	三輪 浩	奥村 直生, 辻 優希, 辰野 渉, 竹村 裕二, 中野 海
44	Go! UP —助かる命を守りたい—	沖 縄	正木 忠勝	大西 諒, 山城 響, 野崎清太郎, 賀数 志乃, 又吉 純次
45	Sneaker@mail —心配なあの人に届ける便り—	沖 縄	鈴木 大作	近藤 史麻, 久保田明成, 下地 春希, 川満 大輝, 貝盛 陽平
46	まごっち! —健康とコミュニティ再生ボランティア支援システム—	八 戸	細越 淳一	竹林 広樹, 武田 瑞生, 大坂 侑平, 大坂 健悟, 外里 有蘭
47	物資Administrator —物資の管理をスマートに—	久 留 米	丸山 延康	竹中 孝介, 石王丸 智, 山村 大介, 横溝 寛人, 中島 暢哉
48	O・MO・I・YA・RI —被災地に届けICTボランティア—	大 島 商 船	浦上美佐子	藤重 幸弘, 金岡 拓哉, 岩見 将平, 藤田 理斗, 中川 碧惟
49	ESCAPE! —3DシミュレーションクイズRPG—	近畿大学高専	宇田 隆幸	山内 翔太, 宮田 侑一, 細見 隼矢
50	Smooth Shopping	福 島	鳥村 浩	山廻達貴裕, 大田 晃平, 佐々木和仁
51	S. EM —Simulated Emergency—	奈 良	中村 善一	大西 将樹, 中村 匠, 増田 秀之, 森 大海, 藤原 詢太
52	SKcAP —空から支える復興支援—	宇 部	田辺 誠	金子 昂稔, 西村 優佑, 伊藤 賢也, 田代 翔也, 久継 宏樹
53	RE.PAD —防災意識向上システム—	舞 鶴	片山 英昭	樋口 拓海, 市川 公亮, 大西 照, 岡原 伸, 小野 数馬
54	キズナ時計 —万が一に備えた腕時計—	新 居 浜	先山 卓朗	山之口 智也, 末光史有土, 宇佐美健太, 田中 宏政
55	DTN通信を用いた災害時の安否及び避難所情報収集システム	富山(射水)	山口 晃史	笠山 北斗, 栗 瑞起, 前田 広夢, 柴田 大希, 藤野 裕時
56	iVent —イベントはじめてみませんか?—	香川(詫間)	宮武 明義	小川 航平, 片岡 麻輝, 竹本 伊吹, 田上 大智, 井村菜々子

■ 自由部門

登録順	タイトル	高専名	指導教員	参加学生
1	Wake Up Water! —水を飲んでお目覚めすっきり!—	鈴 鹿	箕浦 弘人	新美 和生, 平川 貴大, 駒田 航平, 澤 雄太, 倉田 直樹
2	V-Wear —“コレジャナイ!”を無くすWeb試着サービス—	有 明	松野 良信	山口 賢二, 米崎 勝亮, 前田 恵利, 甲田 芹佳, 居石 桃夜
3	むれ —ロボット群の制御システム—	小 山	平田 克己	菊田 佳孝, 関根 将也, 松本 貴志, 大浦 友暉, 岡崎 将也
4	おいしいよ!	小 山	平田 克己	金子 祥治, 石川 颯, 海老原俊輔, 栗田 朋幸, 丸山 颯
5	クレバーM@SK —健康志向マスク型入力インタフェース—	米 子	河野 清尊	塚田 美香, 長谷川陽平, 清水 大輝, 野口 直希, 大口 陽生
6	はこにわーど	鳥羽商船	中井 一文	矢倉 章恵, 栗原 亨徳, 濱口 祐輝, 宮原 渉, 田川 瑞季
7	はなまるフォーム —スポーツ上達サポートアプリ—	鳥羽商船	都築 啓太	小野 陸翔, 池田 友美, 小山 紗希, 勝田 百香, 河口 祭
8	分子結合シミュレーター —くっつけて、回して、いじってみよう—	一 関	千田 栄幸	及川 達也, 堀金 周, 藤田 大輝, 太田 拓海, 三宅 孝平
9	Writing —自分と 世界と つながる手帳—	津 山	寺元 貴幸	末田 卓巳, 佐伯 典祐, 岸本 大知, 奥田 智大
10	Expace —3Dをもっと身近に—	津 山	寺元 貴幸	国定 凌太, 濱本 幸輝, 竹原 翔也, 寺尾 奈々, 安達 哲哉
11	しんちょクール	興	藤井 敏則	作田安希忠, 西山 拓海, 小池 英潔, 岩崎 巧磨, 吉本 海
12	Unidentified Flying Osouji	金 沢	ソングー ロバート	小田 翔一, 黒澤 清和, 佐野 智寛, 成清 敬太, 三澤 裕樹
13	すごろく観光 —ゲーム型 観光サポートアプリ—	金 沢	藤沢 武	小野 叶太, 遠山 勇樹, 外松 俊尚, 川上 悠太, 山崎 寛人
14	SNS炎上報知器 —個人情報流出・炎上防止見守りシステム—	東 京	松林 勝志	松林 圭, 松原 良和, 五味 京祐, 古川 和祐, 松尾 祐佳
15	cosmo craft	久 留 米	嶋田 英樹	辻 巧斗, 千北 一期, 船越 南斗, 久保田祥平, 江崎 喬祐
16	if —可能性の分かれ道—	旭 川	森川 一	鴻野 友樹, 谷村 恭兵, 千葉 大史
17	すまふおーすたでい —情報端末と勉強を繋げる—	有 明	松野 良信	宮崎 大輝, 石川 順平, 井川 大翔, 八木 逸斗, 岩屋 俊祐

自由部門

登録順	タイトル	高専名	指導教員	参加学生
18	おそらく王手 —AR量子将棋盤—	福島	小泉 康一	青木 僚平, 岩本 慎太郎, 小林 尚輝, 川崎 裕幸, 岩間 野目 大
19	Code on the Tablet	弓削商船	長尾 和彦	古谷 勇樹, 林 真史, 山本 隆弘, 井上 香澄, 松川 瑞樹
20	TeleRokuro —マルチプレイヤー3DCGビルダー—	熊本(熊本)	篠 寧平	村上 雅貴, 八木 智徳, 田中 亨, 山下 陽平, 中神 楓子
21	PictoTube —ピクトグラムM@STER—	福井	西 仁司	袖川 瑞, 日下 翼, 山田 友一, 田中 涼太
22	Rapid House —サーバ自動チューニングシステム—	釧路	神谷 昭基	森越 友祐, 尾崎 幸一, 加藤 頑馬, 風間 健祐, 真田 慧一朗
23	STEP APP!!	熊本(八代)	開 豊	岩本 優, 上田 尚人, 渡邊 謙誠, 西崎 友輔, 松岡 雄太
24	postKey —会議室らくらくシェアシステム—	鈴鹿	田添 丈博	大森 智喜, 小野 せりな, 佐藤 幹人, 堀 未米, 黒澤 預生
25	Crowd Garden —ラクして植育!!—	茨城	吉成 偉久	高野 颯人, 磯部 駿人, 川又 亮太, 宮里 宙希
26	あんりみったあ —集中モード, 入ります—	松江	稲葉 洋	金榮 良磨, 堅固 潤也, 青山 南紗, 金長 幸希, 角田 友美
27	Goggle Ocean —寸先は「海」—	熊本(熊本)	藤井 慶	福入 遠太, 委澤 大輝, 川上 誠人, 福田 晃己, 平田 将大
28	チャリit GO	宇部	江原 史朗	岩城 創也, 孫谷 崇之, 三井 悠也, 高橋 瑛介, 畑山 直樹
29	NaVi —Navigate by Vibration—	長野	伊藤 祥一	宮前 海里, 小泉 拓海, 牛澤 葵, 酒井 聖矢, 長谷川 就
30	L×M×S —オープンデータ保管計画—	函館	高橋 直樹	谷 修一, 中島 梢
31	Current Code	仙台(広瀬)	高橋 晶子	佐藤 葉介, 中村 聡邦, 伊勢 竜也
32	Carly! —新時代のキャリアケース—	大阪府大高専	窪田 哲也	伊藤 圭佑, 西村 礼恩, 沖岡 雄希, 小山 涼太, 大和 弘明
33	すくえあ	香川(詫間)	金澤 啓三	金子 高大, 瀧下 祥, 栗山 幸弘, 山崎 啓太, 山下 昂太
34	読書のオトモ	— 関	千田 栄幸	小野 寺和也, 渡邊 祐汰, 泉 望, 永原 基也, 佐藤 永基
35	ポケットレジスタ	阿南	児島 雄志	鈴江 翔樹, 瀧本 大介, 森井 理智, 岡田 大樹, 本浄 定俊
36	御座敷トラベラー —メガネをかければ今ここから広がるセカイ—	徳山	高山 泰博	小松 弘人, 宮本 颯, 片井 拓弥
37	PENカンファレンス	阿南	児島 雄志	多富 信輔, 高橋 遼, 森 公希, 櫻福 智哉, 佐藤 智哉
38	あみがみ様 —3D編物デザイン支援システム—	米子	松本 正己	足立 奈々美, 生田 悠華, 谷口 正明, 山崎 千裕
39	ネツチュー —熱中症対策システム—	舞鶴	船木 英岳	治居 直哉, 渡部 翔太, 藤村 匡弘, 石本 浩気, 浦 優輝
40	CosmoSWeeper —スペースデブリ回収体験システム—	八戸	釜谷 博行	奈良 公明, 竹林 雅人, 永田 大也, 鈴木 康太, 唐生 良太郎
41	Remoconia —ひとつの家庭にひとつのリモコン—	沖縄	正木 忠勝	山本 翔翔, 辺土名 朝飛, 知念 響紀, 當間 環, 當山 大騎
42	monolith —伝道型町おこしシステム—	広島商船	岩切 裕哉	時 友乎, 仲井 友紀, 藤澤 覚司, 中村 果帆, 山根 奈々
43	Session Amigo	沖縄	正木 忠勝	照屋 大地, 名渡山 夏子, 吳屋 寛裕, 奥那城 有, 潮平 諒也
44	単位取得コンシェルジュ —ToBe, OrNotToBe—	舞鶴	片山 英昭	古林 俊祐, 小阪 飛陽, 藤原健太郎, 蔭山海 一郎
45	GIGANE —MEGANEの一步先へ—	福井	斉藤 徹	岩崎 琢真, 小寺 遼太郎, 畠中 達也, 藤田 清司郎
46	HAO —Human Assistance Operator—	北九州	白濱 成希	井川 直也, 木下 陽博
47	のう×これ —脳波これくしょん—	北九州	白濱 成希	吉松 優希, 木津 貴裕
48	actif quest —高専祭を攻略せよ!—	明石	新井 イスマイル	塚本 英成, 和田 佳大, 玉井 雄一郎, 山下 紗由, 泉 将之
49	ハートフルトイレ —トイレが育む真心日本—	奈良	岩田 大志	矢鋪 知哉, 魚谷 果那, 福岡 久和, 原 一彰, 高樋 美奈
50	BEST PARTNER —最も効率的な英単語学習—	福島	島村 浩	永井 智志, 遠藤 優斗, 吉渡 匠汰, 海蔵 育未
51	TIP OFF! —自宅Let'sバスケ—	松江	安井 希子	青木 拓海, 岩田 和俊, 稲葉 光彦, 名原 紫織
52	デジタル香時計 —“時感”という安らぎをあなたに—	新居浜	先山 卓朗	若本 悦子, 加藤 幹也, 菅 淳志, 刈谷 洗一, 高岡 康平
53	E to F —自習のモチベーションを上げるシステム—	仙台(広瀬)	穂坂 紀子	高林 佳稀, 山田 達也, 大子 田克司, 戸嶋 丈士
54	BUS×BAS —バス乗降サポートシステム—	富山(射水)	山口 晃史	平野 万由子, 浦上 拓人, 高井 航平, 大川 大, 並川 遼太郎
55	CT・MRIによる骨粗鬆症の3次元解析	新居浜	平野 雅嗣	芝山 晃弘, 伊藤 紫芳
56	Cyclus —持ち運びの容易な自転車シミュレーター—	奈良	上田 悦子	們原 雄士, 紀木 雅大, 芝脇 智将, 島田 佳哉, 中村 月威

競技部門本選参加作品

■「キオクのかげらⅡ」

登録順	タイトル	高専・大学名	指導教員	参加学生
1	NISHI —Automatic Puzzle Solver—	熊本(熊本)	孫 寧平	上野裕一郎, 宮本 崇平, 西 陽太
2	アカシックレコード —失われた鎮魂曲—	群馬	神長 保仁	小針 和也, 鹿島 大河, 真中 鷹滉
3	画像が、もとに戻るまで、僕は、交換をやめない！	八 戸	細川 靖	平舘 侑樹, 日山 拓海, 道上 和馬
4	ダイオウグソクムシNo.1 —あの日の思い出をもう一度—	鳥羽商船	都築 啓太	宮村騎久也, 坂下 雄摩, PALIN CHOVIWATANA
5	Short Time Artful Program —正当性—	小 山	南斉 清巳	伊東 燦, 寺崎 峻, 河内 祐太
6	賽巡TRIDE改 —ハスケルパズルサウンド—	茨 城	安細 勉	中村 泰大, 木下 郁章, 清水 琢見
7	ダイス&ドラゴンズ	石 川	小村良太郎	中野 毅郎, 木村 優作, 野江 匠
8	SOT理論	秋 田	竹下 大樹	田森 琢陽, 太田 一真, 澁谷 諒祐
9	Reform the picture	呉	藤井 敏則	安岡 健太, 菅 健一郎, 谷口 敏弘
10	画像スクレイプ！産廃と化したサイコロ —産廃なにしてんすか！—	鉧 路	神谷 昭基	谷口 昂大, 寺地 海渡, 森下 真孝
11	パズル&むーちょ	広島商船	大高 洸輝	河村 彩加, 津田 航, 中川 麻美
12	ご注文は原画像ですか？	大島商船	神田 全啓	藤川 晃希, 山重 雄哉, THEDJOISWORO JASON EDBERT
13	解これ！ —いいこと？ 白紙のテキストに解答を刻みなさいっ！—	旭 川	森川 一	青井 亮, 小野 敦夢, 新田 陸
14	まだふみもみず、あまのはしだて —ちくわは無限に存在する—	徳 山	力 規晃	近藤 佑樹, 中島 淳平, 村重 哲史
15	画像！並べずにはいられないっ！	都 城	小林 洋介	坂本 教真, 原田 健太, 中村 大牙
16	—気呵成	弓削商船	長尾 和彦	桶田 雅美, Halim Calvin, 伊藤 宏紀
17	8GBフロッピー	長 岡	竹部 啓輔	高梨 那之, 米山 慧, 中島 英
18	競技ロボ モウダメダー —そうだ、手でやろう—	福 島	小泉 康一	猪野 康弘, 駒木根大海, 馬目 華奈
19	サイコロの霊圧が…消えた…？ —かしこまりノバァンノ(復元) —	鹿児島	武田 和大	福永 彬, 池田 伊織
20	ご注文はパズルですか？	津 山	岡田 正	小林 龍平, 小笠原悠太, 氏平 誠司
21	—筆くん	高 知	谷澤 俊弘	堀本 考宏, 島 仁誠, 水野 裕晴
22	コロココロココロ♪ —シュインシュインシュイン♪—	米 子	倉田 久靖	渡邊 宗成, 勝部 択智, 清水 壮
23	サイコロの画像を並び替えて通信する野鳥の会	仙台(広瀬)	園田 潤	千坂 優佑, 佐久間 諒, 木村 駿
24	逆転再現256 —異議あり！この画像は隣とムジュンしています—	香川(高松)	柿元 健	渡辺 竜, 金丸 将之, 丸山 裕雅
25	Renaissance —そのとき画像は復活されたのです—	サレジオ	内田 健	古川 泰地, 鈴木 裕也, 加藤 慎二
26	人力の神話	明 石	佐村 敏治	中田 季利, 唐澤 弘明, 立川 剛至

■「キオクのかげらⅡ」

登録順	タイトル	高専・大学名	指導教員	参加学生
27	プチトマト	近畿大学高専	政清 史晃	岡森 悠真, 矢羽田浩志, 山内 翔太
28	ご注文はサイコロですか！	熊本(八代)	小島 俊輔	池田 裕季, 田崎あすか, 宮尾 竜矢
29	PIECE×PIECE II —最後のサイコロ美味しかったね—	久 留 米	黒木 祥光	真次 彰平, 吉田涼一郎, 福永 隼也
30	LOVE & PIECE —そんなことよりラーメン食べたい—	木 更 津	丸山真佐夫	江澤 拓哉, 佐藤 陸, 芦田 智季
31	未確認で情報系 —ばいなりわーるど／ぷろぐらみんぐッ！—	金 沢	田村 景明	砺波 大生, 根来 航己, 宮西 洋輔
32	大交換スライドブラザーズX	福 井	村田 知也	山田 涼太, 大柳 慶悟, 宮脇 颯士
33	このパズルを解きたい —パズルだよ！絶対、パズルだよ！—	苫 小 牧	三上 剛	佐藤 陸, 千葉 泰理, 三浦 準也
34	復元レンズ —信じて…こころがときめいた瞬間を—	新 居 浜	占部 弘治	吉田 晟佑, 飛鷹 智浩, 鈴木 龍斗
35	Puzzle & Procons	宇 部	江原 史朗	湯面 翔, 西村 康佑, 貞弘 耕平
36	ご注文はクラムボンですか？	東 京	山下 晃弘	瀧島 和則, 高橋彰太郎, 東 悟大
37	嗚呼、PMANI六段アルゴ難民 —四問目に癖がついてますよ。—	都立(品川)	佐藤 喬	土田 雄輝, 藤野 真人, 鈴木 勇哉
38	海鮮丼のパフェなんだよ！	松 江	橋本 剛	内田 啓太, 青木 勢馬, 岸本 啓太
39	バズトラ！ —パズル・トライアル—	大阪府大高専	窪田 哲也	三上 和馬, 帖佐 克己, 加賀 正樹
40	Puzzle And Procons II	岐 阜	廣瀬 康之	岡地 涼輔, 林 昌哉, 山田 健登
41	PAM —Mr. TangTangと愉快的なアルゴリズム達—	神戸市立	朝倉 義裕	隆辻 秀和, 林 純一, 宗和 恭平
42	このあと減茶苦茶フィックスした —\フィックスした／—	大 分	徳尾 健司	吉田 龍矢, 井野元健史, 平井 雅人
43	オーバータイム —原点回帰のスパイラル—	和 歌 山	青山 敏生	間所 匠, 門脇 奨, 吉田 大河
44	メモリコンストラクタ —たった一つの、冴えた解き方—	有 明	尋木 信一	熊手 主, 船越 紀宏, 待鳥 雄哉
45	画像戦争 —んだよ、位置が分かんねえ—	阿 南	一ノ瀬元喜	里谷 佳紀, 東手 理貴, 居石 峻寛
46	void swap (int, int) ; —なぜかできない—	一 関	管 隆寿	齋藤 真慶, 佐藤 将太, 堀金 周
47	高専のパズル概論 I	長 野	伊藤 祥一	原 大樹, 柄澤 駿太, 高野 航雅
48	新委員長就任 —参勤交代2回目まだですか—	沖 縄	正木 忠勝	日熊 悠太, 西原 希咲, 渡嘉敷真優
49	MasterPiece	佐 世 保	手島 裕詞	城間 優作, 田丸 将寛, 吉田 愛永
50	Rest In Piece (仮)	都立(荒川)	鈴木 弘	金子 孝輝, 藁手 智紀, 濱本 海
51	PICT Hate us —Bravo Java！—	北 九 州	松久保 潤	山下 玲音, 大森 優雅, 片岡 洋揮
52	Restoration —高専生は如何に記憶を取り戻したか—	舞 鶴	伊藤 稔	池本 悠利, 田邊 天音, 新治 拓馬

■「キオクのかけらⅡ」

登録順	タイトル	高専・大学名	指導教員	参加学生
53	00を中心に原画像の隊形にい〜整列!!	香川(詫間)	鈴木 浩司	山崎 佑馬, 高志 克俊, 岡坂 朋和
54	日本サイコロ画像復元協会	沼 津	鈴木 康人	山田 知史, 高原 一真, 野村 允春
55	税込¥3156 —古事記にもそうかかっている—	豊 田	庫本 篤	青井 佑, 高嶋 弘樹, 吉永 颯斗
56	正直パズルだった(・ω・)ノ	奈 良	山口 賢一	関 一至, 栞島 基嵩, 島岡 大遥
57	Stack構造でイけます	鈴 鹿	田添 丈博	小原 佑斗, 駒田 拓也, 中村 晃史
58	冴えないパズルの答え方	仙台(名取)	北島 宏之	西方 聖, 武山 英哲, 北島 彰
59	ピースこれくしょん —ご注文は定数オーダーですか?—	富山(射水)	山口 晃史	富岡 稔, 坂本 和哉, 渡辺 孔英
60	ぶる—まうんてん。	京都大学・ 東京大学	青山 俊弘	増井 隆治(京都大学) 渡辺 謙太(東京大学)
61	ご注文はグイスですか?	豊橋技術 科学大学	梅村 恭司	吉田 智晴, 大友 将宏, 小松 和暉
62	タフ&グローバル	東京大学	黒木 祥光	大城 泰平, 矢野倉伊織, 味曾野雅史
63	VNU HANOI	ハノイ 国家大学	To Van Khnh	Nguyen Duy Khanh, Hoang Gia Minh
64	成都東軟移動革新倶楽部	成都東軟 学院	Zhong Baocai	Xu Tengfei, Wang Di
65	The One Piece	モンゴル科学 技術大学	Khuder Altangerel	Shurenchuluun Erdenebat, Khureltsooj Davaatseren
66	(未定)	ペトロナス 工科大学	Ahmad Izuddin Zainal Abidin	Mohammad Khursani Mohd Shaib, Muhammad Nazmi Mat Asri
67	PIT TEAM	パトゥムワン 工科大学	Wasin Pirom	Krit Klinkaeo, Perachart Krongmalai

司会紹介

高田 信一

石川県在住。プロのナレーターとして活躍中。

第13回石川大会以来、現在に至るまで競技部門の司会・進行を担当。

「はぁじめっ!」「ここまでっ!」のかけ声など、いまや高専プロコン競技部門の名物となっている。

競技部門のルール

競技内容は、1枚の原画像から同サイズに切り分けられた断片画像をバラバラに並べた問題画像を、元の画像に並べ替えるパズルゲームです。並び替えは、隣り合う断片画像同士の入れ替えだけで行わなければなりません。回答時間も重要な要素です。より操作の少ない回答を、より早く回答したチームが勝利となります。時間をかけて操作が少ない回答を求めるのか、それとも操作は多めな回答でも素早く回答するのか、勝負の駆け引きになります。

● 用語など

問題関連

原画像

- ・自然画やイラスト画などのフルカラー画像です。
- ・画像サイズは幅、高さともに最小で32ピクセル、最大で1024ピクセルです。問題によって異なります。

分割

- ・原画像を縦、横それぞれ等間隔に分けます。(分割数は画像サイズの約数になります)
- ・分割数は縦、横ともに最小で2、最大で16です。問題によって異なります。
- ・縦と横の分割数が同じとは限りません。

断片画像

- ・原画像を分割して作成した各々の画像です。
- ・すべての断片画像のサイズは同じです。
- ・断片画像のサイズは幅、高さともに最小で16ピクセル、最大で128ピクセルです。問題によって異なります。
- ・断片画像の幅と高さは同じとは限りません。

断片画像の位置

- ・断片画像の位置は表1のように左上を00とし、そこから右に順に10、20、30、…、下に順に01、02、03、…とします。
- ・断片画像の位置は16進数で表します。分割は最大16分割なので、位置は00からFFです。

表1. 断片画像の位置

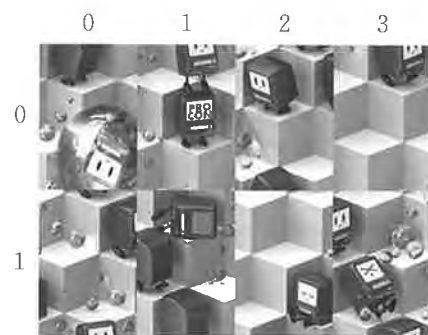
00	10	20	...	E0	F0
01	11	21	...	E1	F1
...
0E	1E	2E	...	EE	FE
0F	1F	2F	...	EF	FF

問題画像

- ・断片画像を無作為に並べ替えた画像です。
- ・断片画像は回転しません。
- ・並べ替えた断片画像を結合して、原画像と同じサイズの1枚の画像とします。



(a) 原画像



(b) 問題画像

図1. (a)原画像を、4×2に分割した(b)問題画像

操作関連

選択

- ・競技者が断片画像の中から1枚の画像を指定することです。
- ・選択した断片画像を選択画像と呼びます。
- ・選択画像は数回変更することができますが、回数が制限されます。問題によって異なりますが最小で2回、最大で16回です。

交換

- ・ 選択画像に隣接する上下左右の4枚の断片画像（選択画像の位置によっては3もしくは2枚の場合もあります）のうちの1枚と選択画像を入れ替えることです。

ライン

- ・ 同一の選択画像で連続した交換操作のことです。

復元

- ・ 選択・交換を行い、断片画像を原画像の位置に戻すことです。
- ・ 選択・交換で並び替えた画像を復元画像と呼びます。
- ・ 復元画像と原画像が一致する復元を、完全復元といいます。
- ・ 原画像によっては同じ断片画像を複数含む場合もありますが、その場合でもそれぞれの断片画像を原画像の元の位置に戻す必要があります。

コスト関連

選択コスト

- ・ 選択回数に選択コスト変換レートを掛けた値です。選択コスト変換レートは1以上300以下の整数で、問題によって異なります。

交換コスト

- ・ 交換回数に交換コスト変換レートを掛けた値です。交換コスト変換レートは1以上100以下の整数で、問題によって異なります。

時間コスト

- ・ 回答を受信するまでの時間（秒数）を100倍した値です。
- ・ 受信するまでの時間にはフォーマットの検証時間も含まれます。
- ・ フォーマットの検証時間は5秒程度を予定しています。回答ごとに異ならないよう競技サーバー側で統一します。

総コスト

- ・ 選択コスト、交換コスト、時間コストの合計です。

有効回答

回答の優先順位を以下のようにして、同一チームが複数の回答を提出した場合は、最も優先順位の高い

回答のみを有効回答とします。

1. 一致断片画像数（原画像との位置が一致した断片画像が多い回答が有効）
2. 総コスト（総コストの少ない回答が有効）
3. 選択コスト（選択回数の少ない回答が有効）
4. 交換コスト（交換回数の少ない回答が有効）

● 制限時間

- ・ 問題ごとに回答の制限時間を定めます。
- ・ 制限時間は1分～15分の予定ですが、実際の制限時間は、試合開始前に連絡します。
- ・ 回答は制限時間内に送信し終わっている必要があります。回答の送信中に制限時間が経過した場合には、その回答は無効になります。

● 試合の進行手順

- ・ 1試合は最大18チーム対戦で行います。
- ・ 1試合は決勝戦以外は3問で行います。
- ・ 1問ごとに定めた数のチームが勝利となり、次の試合に進みます。
- ・ 1試合の問題数および問題ごとの勝利チーム数はあらかじめ告知します。
- ・ 勝利したチームはそれ以降の問題には回答（参加）しません。
- ・ 試合の総合順位は、勝利チームは1問目から順に勝ち抜け順を順位とし、それ以外のチームは、最終問題の順位順にそれ以降の順位とします。

● 順位決定方法

勝敗判定は以下の優先順位で決定します。

1. 一致断片画像数（原画像との位置が一致した断片画像が多いチームが上位）
2. 総コスト（総コストの少ないチームが上位）
3. 選択コスト（選択回数の少ないチームが上位）
4. 交換コスト（交換回数の少ないチームが上位）
5. サイコロの目で勝負（合計が多いチームが上位）

競技部門の組合せ

1 回戦

ブース	第1試合	第2試合	第3試合	第4試合
1	一関	沖縄	苫小牧	長岡
2	釧路	群馬	神戸市立	八戸
3	奈良	香川 (高松)	久留米	仙台 (広瀬)
4	津山	和歌山	秋田	熊本 (熊本)
5	金沢	大分	豊田	米子
6	木更津	岐阜	阿南	鈴鹿
7	大島商船	沼津	有明	近畿大学高専
8	徳山	弓削商船	広島商船	茨城
9	小山	旭川	明石	富山 (射水)
10	福島	香川 (詫間)	仙台 (名取)	都立 (品川)
11	大阪府大高専	佐世保	石川	東京
12	鹿児島	福井	熊本 (八代)	都城
13	北九州	舞鶴	サレジオ	松江
14	宇部	都立 (荒川)	鳥羽商船	呉
15	高知	長野	新居浜	成都東軟学院
16	モンゴル科学技術大学	パトゥムワン工科大学	ハノイ国家大学	ペトロナス工科大学
17	豊橋技術科学大学	東京大学	京都大学・東京大学	
18				

※各試合上位7チームが準決勝へ進出する。8位以下は敗者復活戦へ。

敗者復活戦

ブース	第1試合	第2試合
1	1-1-8	1-2-8
2	1-3-8	1-4-8
3	1-3-9	1-1-9
4	1-4-9	1-2-9
5	1-1-10	1-4-10
6	1-2-10	1-3-10
7	1-2-11	1-1-11
8	1-4-11	1-3-11
9	1-1-12	1-3-12
10	1-2-12	1-4-12
11	1-4-13	1-1-13
12	1-3-13	1-2-13
13	1-1-14	1-2-14
14	1-3-14	1-4-14
15	1-3-15	1-1-15
16		1-2-15
17		
18		

※各試合上位8チームが準決勝へ進出する。

準決勝

ブース	第1試合	第2試合	第3試合
1	モンゴル科学技術大学	ハノイ国家大学	ペトロナス工科大学
2	パトゥムワン工科大学	成都東軟学院	豊橋技術科学大学
3	京都大学・東京大学	東京大学	1-3-1
4	1-1-1	1-2-1	1-2-2
5	1-4-1	1-3-2	1-4-2
6	1-2-3	1-1-2	1-1-3
7	1-3-3	1-4-3	1-2-4
8	1-4-4	1-1-4	1-4-5
9	1-2-5	1-3-4	1-1-5
10	1-3-5	1-2-6	1-3-6
11	1-1-6	1-4-6	1-2-7
12	1-4-7	1-1-7	1-3-7
13	C-1-1	C-2-1	C-1-2
14	C-2-2	C-1-3	C-2-3
15	C-1-4	C-2-4	C-1-6
16	C-2-5	C-1-5	C-1-7
17	C-2-7	C-2-6	C-2-8
18		C-1-8	

※各試合上位6チームが決勝へ進出する。

決勝戦

ブース	第1試合
1	S-1-1
2	S-2-1
3	S-3-1
4	S-1-2
5	S-2-2
6	S-3-2
7	S-1-3
8	S-2-3
9	S-3-3
10	S-1-4
11	S-2-4
12	S-3-4
13	S-1-5
14	S-2-5
15	S-3-5
16	S-1-6
17	S-2-6
18	S-3-6

※X-Y-ZはX回戦-第Y試合-第Z位を表します。

※ただし、X部分のCは敗者復活戦、Sは準決勝を表します。

※1回戦には海外・大学チームが2チームずつオープン参加します。オープン参加のチームの成績は、高専チームの順位には関係ありません。

※準決勝・決勝はNAPROCK国際プロコン (国際大会) を兼ねて実施されます。準決勝・決勝の海外・大学チームは国際大会の公式エントリーで、高専チーム・海外・大学チームの区別なく、試合の成績により決勝進出や準決勝・決勝での国際大会の順位が決まります。

※全国高等専門学校プログラミングコンテストの順位については、海外・大学チームを除きます。

1

NISHI

—Automatic Puzzle Solver—

熊本(熊本)

上野裕一郎(3年) 宮本 崇平(3年)
西 陽太(3年) 孫 寧平(教員)

1. 概要

このプログラムは、以下の3つの部分に分かれている。画像処理により原画像を復元するための”Recover”、パズルを解答するための”Solver”、任意のタイミングでプログラムの意図しない動作の解消、誤った解答を生成した場合に人間が介入するための”NI-SHI”から成り立つものである。

2. ”Recover”

この部分では、断片化され、位置が変更された問題画像を原画像に復元する。各断片において色とエッジの抽出と解析を行う。そのあとに断片間のエッジのマッチングを行い、各々隣接する断片を探し出す。

3. ”Solver”

コストが最小化されるような解を、時間コストが許す範囲で検索する。選択画像を選ぶ際に数学的手法を用い、

プログラムでパズルを解く。

途中で原画像が変更された場合は、現在の解答を放棄し、別の原画像により再び解の算出を開始する。ある程度最適解らしい解が見つければ、それを自動的に”NI-SHI”に送信する。

4. ”NI-SHI”

コンピュータで解けなかった問題や解き間違えた問題がないよう、最終的には人間の手でチェックを行う。 ”Recover”による原画像が間違っていた場合の修正や、現在”Solver”が見つけている解よりも良いものを見つけた場合、それを代わりに送信する。

5. 開発環境

開発環境:Microsoft Visual Studio,Vi IMproved, Emacs

開発言語: C#,C++, .NET Framework 4.5

OS: Microsoft Windows 7 x64

2

アカシックレコード

—失われた鎮魂曲—

群馬

小針 和也(4年) 鹿島 大河(3年)
真中 鷹滉(3年) 神長 保仁(教員)

1. 概要

最初に、原画像を推測し、各断片画像について正しい位置との距離を調べる。次に、この距離が最も長い断片画像を選択し、正しい位置まで最短距離で交換する。このとき、交換の過程で、他の断片画像についての正しい位置までの距離が短くなるように交換経路を選択する。

2. 原画像の推測

問題画像のPPMファイルから各ピクセルのRGB要素を取得し、ひとつの断片画像の1辺と他の全ての断片画像の隣り合う1辺のRGB要素の差(絶対値)を1ピクセルずつ計算する。この要素の差の合計値が最も小さい(断片画像の)辺が正しく隣り合うものとし、これを4辺および全ての断片画像について繰り返す。推測された隣接画像から断片画像の正しい位置を求める。

3. 選択および交換方法

問題画像と推測された原画像を比較し、すべての断片画像について上下方向、左右方向にそれぞれいくつ交換する必要があるかを調べる。そして、上下左右に最も多く交換する必要があるもの(最も長い距離のもの)を選択する。この時、交換するべき方向は上下の一方、左右の一方の2方向以内となり、この2方向にのみ選択画像は交換を行う。選択画像を交換する際に、交換される画像が、選択画像と交換を行うことにより距離が短くなる方向を優先して交換する。交換した後は、交換された画像の距離を更新する。選択画像を推測される位置まで交換し終えたら、新たに選択し、交換を行う。

4. 開発環境

Visual Studio 2012 Ultimate

Visual C++ 2012

3

画像が、もとに戻るまで、僕は、交換をやめない！

八戸

平箱 侑樹(5年) 日山 拓海(4年)
道上 和馬(3年) 細川 靖(教員)

1. システム概要

問題として送られてくる画像を元画像、分割した画像を断片画像、元画像が正しく並べ替えられた画像を目標画像と呼ぶ。



元画像



目標画像

競技にはノートパソコンを三台使用して、画像認識と画像の並べ替えをそれぞれ異なったパソコンを使用して行う。

一台のパソコンは画像認識によって元画像から目標画像を作成し、その情報を他のパソコンに送る。

残りの2台のパソコンは画像認識を行なったパソコンから情報を受け取り、それを元にもたらされたルールに従って元画像を目標画像に並び替える手順を探索する。

2. 画像認識

「グレースケールで画像を読み込み、画像を分割する。全ての断片画像の上下左右の縁を1列ずつ読み込む。他の

画像の縁と比較する。最も誤差の低い画像を隣に置く」これを繰り返し、目標画像を作成する。

目標画像を作成した後、人の手でおかしいところがないか確認し、他のパソコンに目標画像の情報を送る。

3. 並べ替え

受け取った情報には断片画像が目標画像のどの位置に当てはまるかが記されており、この情報を元に画像を並び替えていく。並べ替えは複数のパソコンでそれぞれ違うアルゴリズムにより行う。幅優先探索などを使ってよりコストの少ない手順を探す。

4. 通信

3台のパソコン間の情報の送受は、パソコンを有線で接続して通信を行う。

5. 人が操作できるインタフェース

画像認識、並べ替えの両方の部分でコンピュータだけでなく、人が操作できるようなインタフェースを実装し、人の目で見て間違いを修正する。

6. 開発環境

Microsoft Visual Studio 2013
OpenCV DX ライブラリ

4

ダイオウグソクムシNo.1

—あの日の思い出をもう一度—

鳥羽商船

宮村 騎久也(5年) 坂下 雄摩(5年)
PALIN CHOVIWATANA(4年)
都築 啓太(教員)

1. 概要

問題画像のデータから完成画像を推測する。それぞれの断片画像の正解位置データを交換アルゴリズムを用いて処理し、解答を生成する。

2. 断片画像の原画像における位置推測法

断片画像の四辺を評価用ピクセルとする。初めにある一つの断片画像を選び、他のすべての断片画像の評価用ピクセルと差分をとることで隣接する画像を推測する。ある辺で隣接する断片画像が存在しないと判断されれば原画像を構成する端の断片画像と判断し、位置を順次確定させていく。

3. パズルの交換手順

まず各断片画像の正解位置データから、元の位置への最

短経路を探索する。最短経路の最も長いノードを最初に選択し、交換を始める。片画像の位置関係、選択レート、交換レート、最大選択回数を考慮しファジィ推論を用いて最適な交換手順の推定を行う。ひとつのノードの交換手順が決定したら、次も最短経路の最も長いノードを選択し、交換を始める。これを繰り返すことで断片画像の位置を原画像の位置に戻す交換手順を求める。

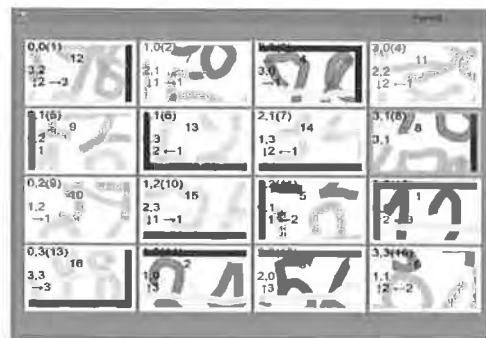


図1. テスト用アプリケーション

5

Short Time Artful Program

—正当性—

小山

伊東 燦(5年) 寺崎 峻(3年)
河内 祐太(3年) 南斉 清巳(教員)

1. 処理の流れ

与えられた問題画像を要素数で分割し、断片画像から原画像を推定する。元の配置から、得られた正しい配置までの経路を求める。また、問題画像の取得と回答の提出を自動で行うようにした。

2. 原画像の推定

断片画像を二値化し、その2つの断片画像の接続部分の2辺について、1ピクセルごとに差を求め、その絶対値の総和を結合コストとする。この結合コストをすべての断片画像のペアについて求める。

結合コストが最も小さくなる断片画像の組み合わせが正しいペアであるから、そのペアについて連結していくことにより原画像を推定する。

3. 経路探索

断片画像を選択して交換する場合に、選択した断片画像の正しい位置までの距離や、選択していない断片画像が交換によって正しい位置に近づく距離などをもとにコストを算出し、そのコストが最小となるような経路を幅優先探索によって求める。

また、すべての経路を探索するためには膨大な量の計算が必要なため、探索する経路のうち、コストが大きいものについては枝刈りを行うことによって、探索経路を一定数にし、それによって処理時間の短縮を図る。

4. 開発環境

Editor: Vim, Emacs

Compiler: GCC, Intel C Compiler

Library: OpenCV, Boost, OpenMP

6

賽巡TRIDE改

—ハスケルパズルサウンド—

茨城

中村 泰大(5年) 木下 郁章(4年)
清水 琢見(4年) 安細 勉(教員)

1. アプローチ

断片画像の本来の位置を判定し、その情報から最良な選択・交換の解を求める、という手続きで問題画像から原画像を復元する。また、高速な計算を実現するために、各手続きを代数学の領域に当てはめ、性質を利用する。

2. 正解位置の判定

ある1つの断片画像において、周囲の断片画像との隣接関係が正解画像における隣接関係と等しい場合、画像の境界付近において色の変化量は限りなく低い。よって、すべての断片画像において、色の変化量が最小になる組み合わせを求めることで、全断片画像の正しい隣接関係を求めることができる。このとき、最小組み合わせを求める方法として、制約ソルバを用いる。

3. 最良手の生成

Algebraic Dynamic Programmingと総称される手法で解法を導出する。解候補の「連結」「和集合」から、最適解のコストの「和」「最小値」にその性質が保存されることを利用し、効率よく解を生成する。より早く解を求めるため、並列化を行った。

4. 問題の選択及び回答

回答する問題の選択は人間が行う。この操作はGUIを用いてマウス操作等で簡単に行われる。問題の選択後、迅速に回答を生成し、自動的にサーバへ回答の送信を行う。

5. 開発言語

Haskell

7

ダイス & ドラゴンス

石川

中野 毅郎(5年) 木村 優作(5年)
野江 匠(5年) 小村良太郎(教員)

1. 概要

このシステムでは、次の流れで解答を行う。まず初めに、画像処理によってそれぞれのピースの正しい位置を求め、完成画像を予想する。その後、ピースを動かす手順を 3 種類の方法を用いて解いていく。

2. 方法

画像の完成には 2 種類の方法を用いる。また、PC を 3 台使用するためピースの並び替えには 3 種類の方法を用意する。3 つの中でコストが一番小さいものを解答として提出する。

2.1a 画像完成

各ピースのエッジ部分の類似性に基づいて、ピースをつなぎあわせて完成画像を作成する。

2.1b 画像完成

2.1a の方法で完成しなかった部分について、手動での並び替えを行い、画像を完成させる。

2.2a ピースの並び替え

目視でピースを選んで並び替える。タッチパネルを使うので効率がいい。

2.2b ピースの並び替え

1 ピースずつ解いていく。時間コストが少なく並び替えコストが高くなる。

2.2c ピースの並び替え

マンハッタン距離の総和を用いて A*アルゴリズムで解いていく。並び替えコストは最小になるがピースが多いと時間内に解けなくなる。

3. 開発環境

・ Visual studio 2010

8

SOT理論

秋田

田森 琢陽(4年) 太田 一真(4年)
澁谷 諒祐(4年) 竹下 大樹(教員)

1. アプローチ

現状普及しているマシンのスペックでは、通常的全探索では遷移が多すぎて計算できない。そこで、操作を制限することで計算量を落とそうと試みた。

2. 問題の変換

この問題は、パズルの探索として見るのではなく、バラバラの数字を決められた順序に直す、つまりソート問題と見るべきである。

これにより操作をうまく制限し、ソートを工夫すれば、十分短い時間で計算を終えることができるであろうと考えている。

3. スライドパズルとの違い

この問題で普通のスライドパズルと異なるのは、自由

に穴(空白)部分を入れ替えられることにある。問題によってその回数は異なるが、これをうまく使う必要がある。しかし、回数は少ないため、これは人間の手で場面を見極めてもよいと考えた。

4. 移動回数の増大

ソート問題として操作を制限することのデメリットに、無意味な移動操作を内包してしまうことがある。よって、手数が重要になり比較的サイズの小さいパズルの場合は、従来のような探索に少しの工夫をほどこしたものをやることにした。

5. 開発環境

Visual Studio 2012-2014 C++
OpenGL

9

Reform the picture

呉

安岡 健太(1年) 菅 健一郎(1年)
谷口 敏弘(1年) 藤井 敏則(教員)

1. アプローチ

サーバーから取得した ppm 画像を適切な形式に一度変換し、アプリケーションに表示させて自動もしくは手で並べ替えを行う。また、画像の各ピクセルに対し、グレースケーリングを行った白黒の画像を用意し、陰影を分かりやすくし、自動で並べ替えするプログラムに送る。

2. プログラムについて

2.1 ppmの形式変換

ppm を一度 bmp に再構築するプログラムを作成する。このときそれぞれのパネル一枚一枚別のファイルとして生成し、並べ替えの処理を簡単にする。また、グレースケーリングした画像を用意し、画像比較を高速化する。

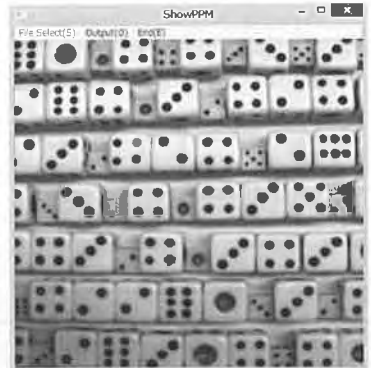
2.2 元画像の生成

元画像を一度生成することで、自動並べ替えを簡潔にする。

る。原画からランダムで一枚とりだし、そこから横1列を生成し、それをくりかえしていき最後に全ての画像をくっつけることで画像の生成を行う。

2.3 自動並べ替え

自動並べ替えを総当りで行うと、分割数が増えるにつれ比較を行う回数が爆発的に増える。このため、優先度を利用したプログラムを作成する。生成した答えと思われる画像の x, y から優先度を求め、そこから最短距離をもとめる。



3. 開発環境

Visual Studio

C++ 2010

10

画像スクレイプ！ 産廃と化したサイコロ

—産廃なにしてんすか！—

釧路

谷口 昂大(3年) 寺地 海渡(3年)
森下 真孝(専1年) 神谷 昭基(教員)

1. アプローチ

本システムでは、バラバラになった問題画像から元画像を推定し、ピースの位置を推定する。推定結果より、 $n \times m$ パズルを定義し、問題で与えられたコストをなるべく最小にするような解法を求める。

2. 元画像の推定

元画像の推定には、ピースの4辺の画素を取得し、それぞれのピース同士の組み合わせで、もっとも辺同士の類似度が高くなるような組み合わせを求めることによって実現する。

3. 並べ替え最適化

$n \times m$ パズルの解を求めるアルゴリズムとして、

欲張り法を用いる。まず、パズル上から四隅に配置すべきピースをとりだす。そのピースの内、正解の配置と移動後の配置のハミング距離が最も小さくなるようなピースを選択し、ピースを正しい位置まで移動させる。この際の移動方法は移動距離が最短となるようなものを選ぶ。選択したピースを正しい位置まで移動したら、そのピースを固定する。固定したピースは壁としてみなし、壁によって生じる領域のうち、隅に位置するものを次の選択するピースの候補とする。

4. 開発環境

Visual C++ 2010

OpenCV

11

パズル&むーちょ

広島商船

河村 彩加(3年) 津田 航(3年)
中川 麻美(3年) 大高 洸輝(教員)

■ 概要

問題画像を読み込み一つずつ色の分析をする。色の関連性を調べ比較し隣り合う画像を判断することで、元の画像を予測していく。予測された画像を基に交換回数が最小になるように画像の交換を行う。各断片画像は、移動させたい画像とその目的地を指定すると、目的地に向かって移動する。その際、画像の位置が明らかに誤っている場合は手動で画像の位置を修正する。なお、自動で回答フォーマットを作成することで高速化を図った。



■ 開発環境

- Visual Studio 2012
- processing

12

ご注文は原画像ですか？

大島商船

藤川 晃希(3年) 山重 雄哉(3年)
THEDJOISWORO JASON EDBERT(3年)
神田 全啓(教員)

1. システム概要

本システムは問題画像をダウンロードし、解答を提出するネットワーク部と、問題画像を解析し断片画像が原画像のどこに位置するのかを決定する位置決定部と、並び替えの処理をする復元操作部によって構成されている。

2.1 原画像の位置決定

問題画像は断片画像を結合して、1枚の画像としてあるので、1枚ずつに分割して処理をする。それぞれの断片画像同士の繋がり度を独自のアルゴリズムを用いて、すべての組み合わせにおいて調べ、表にする。その表より可能性の高い角、辺、内部を順に決定していく。

決定した後、予測した原画像を表示し、人間の目で確認をし、必要であれば人間の手によって変更を行う。

2.2 復元操作

復元操作は選択回数が2回以内で必ず復元できることを利用して、複数の探索アルゴリズムを用いて作った独自の移動方法を試行し、最もよいものを解答として提出する。

3. 開発環境

Visual Studio 2010 C++ Express

Visual Studio 2010 C# Express

OpenCV version 2.4.9

16	2	3	4	➔	1	2	3	4
1	7	10	11		5	6	7	8
14	15	5	12		9	10	11	12
6	8	13	9		13	14	15	16

13

解これ！

—いいこと？ 白紙のテキストに
解答を刻みなさいっ！—

旭川

青井 亮(4年) 小野 敦夢(4年)
新田 陸(2年) 森川 一(教員)

1. システム概要

本システムは、主に画像解析プログラム、狭域探索プログラム、広域探索プログラムの3つのプログラムからなる。

2. 画像解析

画像解析では、全ての断片画像の注目画像と園周辺画像の組み合わせにおける RGB 値の差や、k-means 法を使ったクラスタリングの結果などの評価結果を参考に、画像を再構成する。

3. 狭域探索

完成状態までの推定値によって最短経路を求める A*アルゴリズムを用いて探索を実行する。評価値は、スライドパズルの偶奇性などから算出する。分割数が大きくなるに従って、組み合わせ爆発が発生し計算時間が大幅に増えるため、分割数が一定以上の場合、後述する広域探索プログラムを先に実行する。

4. 広域探索

スライドパズルの持ち手を変更せずに対象のセルを復元する。この際、全てのマスにおいて不利益にならないルートを探し、移動する。行または列を順に復元させていくことで、狭域探索が可能な範囲まで探索対象マップを縮小する。

5. 回答提出方法

提出用クライアントプログラムは配布されるものを流用するが、開発したシステムに埋め込み、問題受信から解析、提出までの一連の流れを自動化する。

6. 開発環境

IDE Visual studio 2014 (C++及びC#)

Xcode 5.1.1

Library openCV v2.49

Boost v1.55.0

14

まだふみもみず、あまのはしだて
—ちくわは無限に存在する—

徳山

近藤 佑樹(5年) 中島 淳平(専2年)
村重 哲史(4年) 力 規晃(教員)

1. 問題解決の手法

問題を、原画像の推定と、推定した原画像を復元する手順の探索に分割して解決する。

2. 原画像の推定

各断片画像に対して、予め計算しておいた評価値を元どの断片画像をどこに接合するか推定する。これに基づき探索を行い、断片画像の元の並びを推定し、原画像を得る。

評価値には、2つの断片画像の接合部分の画素値の差分をとり合計したものを使用する。

3. 復元手順の探索

大きいサイズの問題をある戦略に基づき小さい問題に分解し、得られた小さい問題に対して解を探索する。

問題を小さい問題へ分解する手法の例として外周埋めがある。外周埋めでは、与えられた問題画像の上下左右のうち一辺を選択し、一列だけ原画像と一致させる。一致さ

せた一列を除外したものを新しい問題とする。この操作を、問題のサイズが十分小さくなるまで繰り返す。

4. ユーザインタフェース

図1のような画像の復元を手動でも行えるユーザインタフェースを作成する。

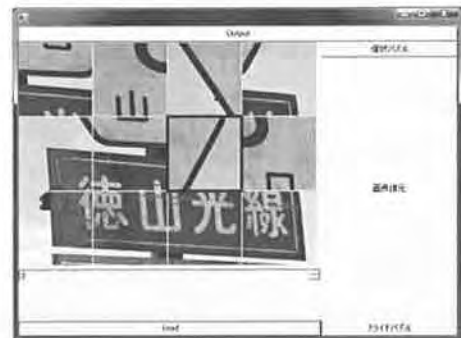


図1 システム画面(開発中のもの)

5. 開発環境

Visual Studio 2010 C#, MinGW

1. 概要

今回の競技は過去にあった競技ルールに、さらに要素を追加したものである。このことから、競技の手順をまとめた単位で分割した。すなわち、問題データを解析する「解析」、解析から得られた結果から並べ替えの最適解を探索する「探索」である

2. 解析の手順

問題データは画素数も分割数も毎回異なるため、解析量を可変することができる仕組みを導入する。その上で、分割された画像の四辺の色情報の一致度を比較していき、一致度が高いものを優先的に配置していく

3. 探索の手順

探索はなるべく多くの分割画像が効率的に正当な位置

に辿り着くように画像の入れ替え手順を模索する

4. エラーについて

解析、探索は基本的に自動で行われるが、特に解析に関しては結果が正答と不一致の部分が出てくることが予想される。これに対して、探索に移る前に人間が訂正できる仕組みを取り入れることで、最終的な回答データの信頼性を高める

5. 開発環境

Visual Studio Express 2013

1. 処理の概要

サーバから取得した問題画像から、元画像の推測、復元手順の探索、回答の自動送信の順序で処理をおこなう。

2. 元画像の推測

断片画像の一致度を計算し、最も高いものから順に候補として割り当て、誤差の最小のものが隣り合った画像となる。評価値は、輝度の比較を簡略化するため各辺の対応するピクセルのRGB値の合計を用いた。この方法では、誤認識があるため人間が介入できるようにしている。

3. 復元手順の探索

全探索は、処理が莫大にかかるなど現実的でないため、最も距離の離れた断片画像を動かして画像の端から順々に決定させる方法を考察した。図1に、評価値及び展開の手順を示す。

- ① 問題画像と元画像の全ての断片画像の距離を計算する。
- ② 完成画像の端の各辺で最も距離の総和が低い(完成に近い)辺を選択する。
- ③ ①の中で距離の最も大きい断片画像を選択する。
- ④ 移動を促すため、②の距離を重み付け(2倍)する。
- ⑤ 全ての断片画像の距離の合計が小さくなるように選択した断片画像③の移動手段を探索する。

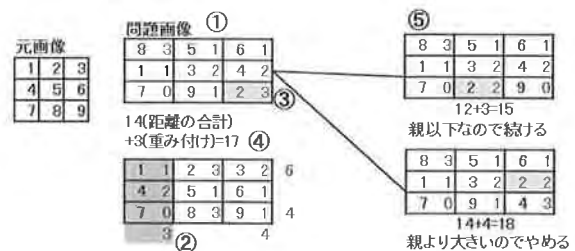


図1. 復元手順

1. アプローチ

写真のような画像を断片化し、それをランダムな順番に並び替えた画像をプログラムで読み込んで、それぞれの断片の本来の位置を推定した後、実際の復元手順を導き出す。

2. 断片画像の本来の位置の推測の仕方

まず、1つの断片画像の右辺について、他の断片画像の左辺と比較を行い、各 y 座標の画素同士の差の 2 乗を計算してその和をとり、それを全ての断片画像の左辺同士と行って、指定した右辺がどの左辺と隣り合うかを調べる。これを全ての断片画像の右辺について行い、各断片画像の左右の位置関係を調べる。上下関係についても、各断片画像の下辺がどの上辺に当てはまるかを同様の手段で計算し、上下の位置関係を調べる。これらから、全ての断片画像の本来の位置を導き出す。

3. 断片画像の復元手順の導出の仕方

問題画像を 2 で求めた断片画像の位置を元に原画像へ復元する。これは任意の選択画像を、反復深化深さ優先探索を用いて復元を行う。また、元の断片画像の配置が奇置換であった場合、原画像への復元が不可能なので、偶置換になるように新たな断片画像を選択する。

4. 間違い訂正について

断片画像の位置の推測が間違えていた場合、マウス操作で断片画像を入れ替えるプログラムを使い、修正を行う。

5. 開発環境

- BCC Developer
- Visual Studio 2013 C++

1. はじめに

今回は画像処理を用いて問題画像から元画像を作り出す画像処理部分と、問題画像から元画像へ戻すまでの手順を考える解答部分の二つに分けて考える。

2. 画像処理部分

隣り合う断片画像の隣接する部分は、正しい場合に最も画素値の差が小さいと考え、一つの断片画像の端と他の断片画像の端の画素値を比較して最小のものを探していくという方法をとる。しかし、これだけでは同じような色を使っているなどといった場合にミスが生まれてしまうため、接合が正しいかどうかの判断の際に画面上に表示させ、人の目からも正誤を判断するようにする。

3. 解答部分

問題画像と元画像での断片画像の位置のずれが最も大きいものを選択画像とし、元の位置まで動かす複数の経路を、交換した断片画像のずれが小さくなったかどうかで評価をしていき、最も評価が良い経路を選ぶ。このとき、ずれの大きさによって重みを変えて評価するようにする。しかし選択回数は上限が設定されているため、最後の選択となった時には全ての断片画像の位置を元の位置に揃えられるようにする。また、分割数が少ない場合は人が経路を考えた方が速い場合もあると考え、人力でも解答を出来るようなインターフェースを設計する。

4. 開発環境

- Visual Studio 2010 C++
- OpenCV

1. システム概要

本システムは、大きく分けて PPM 形式の問題画像からヘッダ情報と画像情報に分けるプログラム群、問題画像における断片画像の本来の位置を特定するプログラム群、その情報を元に原画像を復元するプログラム群からなる。

2. ヘッダ情報と画像情報の分割

PPM 形式の画像中にヘッダ情報(分割数、選択可能回数、各コストの値)と問題画像の情報が入っているので、ヘッダ情報をテキストとして、画像の情報を JPEG や BMP の形式としてそれぞれ分離し、保存する。

3. 原画像における断片画像の位置を特定

2 つの断片画像を比べたときに本来原画像の中で隣接するはずのペアであれば、隣接する部分に急激な画素値の

変化は見られないはずである。この考えから隣接するであろう断片画像のペアを見つけていき、最終的に原画像を復元する。その後、原画像における断片画像の位置の情報を配列として保存する。

4. 原画像を復元

原画像における断片画像の位置情報を受け取り、これを元に 1、2、・・・、n 列目と左の列から順番に復元していく。問題により選択コスト、交換コストが違うのでこれらのコストを加味した上で選択画像や交換画像を決定し総コストの低コスト化を目指す。

5. 開発環境

Microsoft Visual Studio 2013

OpenCV 2.4

1. 動作環境

- OS : Window7,8
- 使用言語 : C#, C++

2. 主な流れ

2.1 画像解析

隣接部分の一致率で断片化された画像同士が隣り合っているかどうかを判断するこの際判定を単純化するため白黒化(2値化)する。白黒化の際、しきい値の設定に判別分析法を用いて、黒か白に偏らないようにして画像を求める。その後左上に来る画像を 1 とし順番を振り分ける。

2.2 画像復元

1 から順に元の位置へ移動させることによって、元の画像に戻していく。

3. 開発環境

Microsoft Visual Studio 2013



開発途中の画像

1. アプローチ

与えられた問題画像を復元し「15 パズル」に類似するパズルの問題に帰着する。その後、最適解を求める。

2. 画像復元

問題画像として原画像を $m \times n$ に分割しランダムに並び替えた画像が与えられる。

$m \times n$ に分割されている画像の各辺の情報を利用し元画像を復元する。

3. 付加番作業

画像復元により復元した元画像を使用し $m \times n$ に分割し左上から順に $1, \dots, m \times n$ まで数字を割り当てる。

その後問題画像の各部分と一致する原画像の番号を割り振る。

4. 最適解探索

前掲の作業により $m \times n$ マスの数字のスライドパズルとほぼ同じような問題となる。

最適解探索のためスライドパズルの移動に深さ制限探索を実施する。

また与えられる移動コストと交換コストの比を利用し短時間で解を見つけ出す。

5. 開発環境

- Java 8
- Eclipse

1. 断片画像の検証

断片画像ごとの左右のコントラスト及び配色パターンを比較、どの断片画像とマッチしているかを把握し、元画像を仮作成する基礎として N 行一列の画像を作成する。

15	19	08	17	02
11	09	0A	0F	0D
03	14	06	0E	18
03	0C	10	12	07
01	04	05	0B	16

→

15	16	17	18	19
0B	0C	0D	0E	0F
06	07	08	09	0A
10	11	12	13	14
01	02	03	04	05

N 行一列の画像ごとに上下の縁のコントラスト及び配色パターンをそれぞれ比較し、元画像を仮作成する。

3. 配置

あらかじめ各断片画像に番号を振り分け、その番号が小さいものから順に最短ルートを解析して配置する。

4. 画像の送信

生成された解答データを即座に送信できるよう自動操作プログラムを作成して必要最小限の操作且つ確実に送信できるように設計する。

5. 開発環境

Microsoft VisualStudio 2010
OpenCV 2.3
Subline Text

2. 元画像データの作成

23

サイコロの画像を並び替えて通信する野鳥の会

仙台(広瀬)

千坂 優佑(5年) 佐久間 諒(4年)
木村 駿(4年) 園田 潤(教員)

1. はじめに

回答プログラムは、まず問題画像から原画像を推測して各断片画像が原画像のどこに位置するかを判断する、次に断片画像を並び替えて復元する手順を導出する、という2つの手順に分かれる。

2. 原画像の推測

各断片画像どうしの「隣接しやすさ」を求め、そこから原画像をジグソーパズルの要領で組み立てる。2つの断片画像の「隣接しやすさ」とは、それらを並べた際、画像としてより自然に繋がる度合いを意味する。

隣接しやすさは、断片画像の各辺についてのある統計学的指標に基づいて算出する。

一致断片画像数をより多くするため、プログラムによって得られた画像は人間が修正することができる。

3. 復元手順の導出

2で求めた、問題画像と原画像の間の断片画像の対応を用いて復元手順を導出する。まず、復元途中の画像と原画像を比較し、それがどれだけ離れているかという値「分散度」を求め、それがより小さくなるような選択・交換の手順を、原画像が得られるまで繰り返す。

選択・交換の手順を求める際は、分散度だけでなく、選択・交換によるコストも考慮して手順を選択する。また、復元における選択回数は限られているため、それを超えないようにする。

4. 開発環境

Visual Studio 2013

C++

24

逆転再現256

—異議あり！この画像は隣とムジュンしています—

香川(高松)

渡辺 竜(専1年) 金丸 将之(4年)
丸山 裕雅(4年) 柿元 健(教員)

1. 問題から原画像の推測

配布されたバイナリデータを基に問題画像を復元し、その問題画像から断片画像の元の位置を推測する。原画像の推測には、断片画像の4辺の色(R, G, B)の類似度を用いる。類似度をみて、お互いに一番類似度が高い画像をつなぎ合わせる。これを繰り返すことによって、問題画像から原画像を推測する。繰り返しても類似度の関係上、一枚の画像とならなかった場合や正しくつながらなかった場合は、手で画像をつなぎ合わせ復元する。

2. 断片画像の交換経路の探索

推測した原画像と問題画像を比較し、断片画像を交換することで、問題画像から原画像を復元する。断片画像の並び替える方法としては、断片画像ごとの最短経路を考え、その最短経路のうち、あらかじめ決めた評価値が最も小さくなるような経路を、その断片画像の並び替え経路とする。

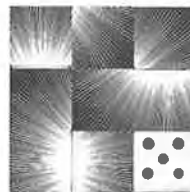
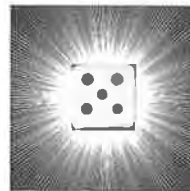
評価値として、原画像との正答数、および原画像とのずれ数を用いる。これを繰り返すことで、原画像の復元を行う。

3. 開発環境

• Visual C++ 2008 Express

• OpenCV

• DX ライブラリ



2
10
7
DRDLURD
10
3
LDD

原画像(左上)、問題画像(左下)および解答(右)

1. システム概要

本システムでは、画像処理を用いて断片画像の元の位置を推測するステップと、問題画像を原画像に復元する手順を探索するステップの2つのステップに分け実現する。

2. 戦略

2.1 画像処理ステップ

断片画像の辺に位置するピクセルの色を他の断片画像の辺と比較し、色の類似度を算出する。その結果から断片画像の元の位置を推測し、復元手順探索ステップへ移る。

2.2 復元手順探索ステップ

画像をノード、選択・交換を枝としたグラフ上で、問題画像から推測された原画像への経路を探索する。この経路が復元の手順となる。

探索にはA*アルゴリズムを使用し、問題の規模に応じて探索時間を短縮するために、探索の深さ・幅の変更、複数

方向での探索を試みる。

3. 使用言語・ライブラリ

- Julia Language 0.3
- Python 2.7.5
- NumPy
- PIL



手動修正のための画像処理ステップ結果表示機能

1. 画像処理

画像処理部では各部分画像が元画像のどの位置にあるのかを推定する。実際的には、各部分画像間に類似度を設定し、それが好ましい値であるペアから順に配置を決定する。類似度は部分画像同士の端部を比較することで決定する。

比較方法は以下の手法を状況により併用している。

- 画像のRGB成分を直接比較
- 画像をグレースケール化したのちに成分を比較
- 画像の成分を偏微分して得られる値や境界検出したものを比較する。

2. 復元手順探索

そもそも復元手順とは、選択・交換の2つの操作の列のことである。これは具体的に言えば、1回の復元手順は{00を選択、01を選択、02を選択、...、FFを選択、選択画像を上を移動、...、選択画像を右に移動}のいずれかになるということである。我々は、それぞれの操作を特に区別せずにこれらの組み合わせと順序を探索していく手法をとることにした。

しかしこの手法は復元手順の種類が多いため、多くのケースにおいて探索空間が途方もなく巨大になってしまう。そこで、探索途中どれだけ探索を進めても解が途方もなく巨大になってしまう。そこで、探索途中どれだけ探索を進めても解があまり改善しないノードを見つける。そのようなノードの探索を打ち切ることで探索空間を小さくすることができ、効率的に良い解を求めることができる。

しかしこの手法には2つの問題点がある。

- 探索にかかる時間が予想しにくい
- 途中までは状態が改善しなかった手順も最適解となりえる

これらの問題は、探索を打ち切る条件を徐々に厳しくし、何度も探索を繰り返す方法をとることで解決できる。探索を打ち切る条件が緩ければ探索は直ちに終了し、また条件が厳しければ途中までの状態が良くない手順を発見することもできるようになるからだ。

1. プログラムの概要

プログラムは、「復元画像予測プログラム」、「選択・交換プログラム」、「回答提出プログラム」から構成される。

「復元画像予測プログラム」は、断片画像の端の画素値を比較し、最も差が少ないものを隣合う画像候補とし、その組み合わせから類推する。正解画像へ順番にIDを付け、局面を保存できるリストを作成する。

「選択・交換プログラム」の選択交換を行うための、評価関数は、選択コスト・交換コスト・時間コスト・各画像のマンハッタン距離を評価パラメータとする。また、同じ画面の堂々巡りを防ぐため、リスト中の同一画面チェックを行う。

「回答提出プログラム」は、「選択・交換プログラム」で、時間内で複数回交換を行った結果のうち、順位決定方法の判定に最も有利な結果を出力する。

2. プログラム間のデータフォーマット仕様

復元画像を構成する断片画像位置のフォーマットは「aabbcc…」で表現する。ただし、aa、bb、ccは問題画像における断片画像の位置を示す。aやbは、'0'～'9'および'A'～'F'の文字であり、コード体系はUTF-8である。すなわち、断片画像1つあたり、2文字でその位置を示すことになる(2文字/位置)。例えば、横が4分割で縦が2分割であるとき、8(4×2=8:断片画像数)×2(文字)の文字列長になる。

3. 開発環境

Microsoft Visual Studio

Visual Basic 2010(GUI用)

Visual C++ 2010(DLLによるハンドラ用)

1. アプローチ

本システムは画像の復元と画像の入れ替えの2つから構成される。分割された断片画像から元の画像を復元し、それぞれの断片画像の正しい位置を記憶する。これをもとに画像の入れ替えを行っていく。

2. 画像の復元

断片画像の端の色情報から隣り合う断片画像を推測する。ただし、これだけでは正確な画像の復元は難しいため、その結果をディスプレイに描画し、人の目による確認と誤りの修正をできるようにした。

3. 画像の入れ替え

画像の入れ替えは、反復深化深さ優先探索を用いて行う。探索をするとき、断片画像の配置から最低でも必要なコストを概算することで枝刈りを行う。概算の方法として、2つの方法を用いる。1つめは、それぞれの断片画像とその

元の位置とのマンハッタン距離を利用したものである。このマンハッタン距離の合計値から最低限必要なコストを求める。2つめは、反転数を用いた方法である。反転数とは、正しい断片画像の位置において左上から順に番号を振っていったときに、現在の断片画像の位置において番号の大小が逆になっているペアの個数をいう。反転数は動かしている断片画像を除けば、上下に動くときのみ変動するため最低限必要な上下動数が求まる。左右の動きも同様に求められる。他にも解を含む可能性のある大雑把な枝刈りも行っている。断片画像の選択をするとき、マンハッタン距離の大きいもののみを選択し、その他を無視することで高速化を図る。解を見つけたら出力し、枝刈りした部分についての探索を続行する。

4. 開発環境

Visual Studio 2010 C++

29

PIECE×PIECE II

—最後のサイコロ美味しかったね—

久留米

真次 彰平(3年) 吉田涼一朗(3年)
福永 隼也(5年) 黒木 祥光(教員)

1. 画像処理

与えられた画像に対して、全ての断片画像の辺の色を比較し、差の少ない並べ方を選択することで、ある程度の精度を持った復元を行う。

また、その配置を 00~FF の値に変換し、それ以降の高速化を図った。

2. 探索

手順の探索には A*探索、反復進化深さ優先探索、ビームサーチ等の探索を元にしたヒューリスティックアルゴリズムを、問題の大きさやコストによって効果的なアルゴリズムを使い分け、時間も考慮しながら解の最適化を行う。

3. 評価

ある状態についての評価は、各断片画像の元の位置まで

のマンハッタン距離を用いて評価する。これに加え、問題分析により見つかった性質などから枝刈りを行い、より高速に、良い解を導出することを図った。

4. 誤りの訂正

Visual C#を用いて Windows Form Application を作成し、問題及びその解を自分たちで視覚的に捉え、さらに画像処理の誤りを人間の手によって修正する。

5. 開発環境

Visual C# 2010
Windows 7/8
GCC

30

LOVE & PIECE

—そんなことよりラーメン食べたい—

木更津

江澤 拓哉(4年) 佐藤 陸(4年)
芦田 智季(4年) 丸山真佐夫(教員)

1. アプローチ

プログラムを画像の配置と画像の並び替えの部分に分けて考える。

1.1 画像の配置

基本的に、各ピースの端のピクセルを比較しその面が隣り合っているかを判断する。この処理を各ピースに対して上下左右の4方向について行う。

画像の配置については、画像の表示をできるようにし、人間の目でも確認できるように実装を行う。

1.2 画像の並び替えについて

画像の並び替えは、15パズルを解読するアルゴリズムをもとに改良を加える事によって実装する。

また、選択と移動のコストを考慮して、最適な移動方法を選択する。

2. 開発環境

エディター : Vim 7.4
使用した言語 : C++, ライブラリ : OpenCV
コンパイラ : GCC(g++)

3. 高速化

高速化を計るために、問題の取得と提出を自動化し、できるだけ人間の手を用いなくてもプログラムが実行できるように工夫をした。

```
sc:31578 in: 180| sc:27279 in:
sc:31699 in: 63| sc:27385 in:
sc:31943 in: 62| sc:27511 in:
sc:32156 in: 61| sc:27782 in:
sc:32185 in: 78| sc:27955 in:
sc:32228 in: 196| sc:28041 in:
sc:32374 in: 79| sc:28195 in:
end calc
```

1. プログラム処理の流れ

このシステムは最初に復元画像を推測し、その画像を元にして、問題画像に達する交換手順を自動的に求めるアルゴリズムを利用している。具体的には、「問題画像の復元 → 断片画像の最短交換手順の導出 → 出力&回答」という流れである。

2. 処理

(1)問題画像の復元

実写画像とイラスト用の問題画像復元プログラムを用意する。初めに、実写画像とイラストを人間が目判断し、実行するプログラムを選択する。その後、明らかに異なる部分を手作業で修正する。

実写画像の処理の場合は、各断片画像の各画素の色の中間値を取得し閾値を定め、類似した色同士の

断片画像の塊を生成する。

イラストの処理の場合は、画像の輪郭を抽出し、輪郭が正しくつながるように判定し画像を復元する。

(2)断片画像の最短交換手順の導出

断片画像の交換手順が最短となるよう問題画像と復元画像から独自のアルゴリズムを使い、最短交換手順を求め、回答フォーマットに従いテキストファイルを出力する。

(3)回答

出力されたテキストをWebフォームから送信する。

3. 開発環境

- OS : Windows , MacOS
- 言語 : Java , Processing

1. 機能の特徴

復元には2種類のアルゴリズムを用いて2重に確認を行い、間違いを減らす。ルート探索は2台のPCそれぞれが一つずつのアルゴリズムを利用して行う。問題のダウンロード・解答の提出は自動的に行う。

2. 元画像の復元について

以下の2種類のアルゴリズムを用いる。

1つは分割画像同士をいったん結合し、直線を検出する方法である。直線が検出されれば、この2つの画像は隣合うものでない。

2つ目は、2つの画像同士の境界ドットのRGBの差の絶対値の和を求め、他のすべての画像よりその値が小さければ隣の画像とする方法である。

最終確認は人の目で行い、正しい並びと判断するまではルート探索は行うが解答の提出は行わない。

3. ルート探索について

以下の2つのアルゴリズムを各一台のPCで実行する。

一つはもとあるべき位置から遠い分割画像を選択し、後退・元の位置にある画像との交換を除いた方向へランダムに進んでゆく。元の位置にある画像と交換するしかない状況になった場合はそちらの方向に進む。

もう一つは元の位置から遠いマスを選択し、自分以外のマスがより近づく方向に進んでいくというものである。この過程で選択する画像もしくは移動方向の選択肢が複数個現れた場合、すべての場合を試しその選択が終了した時点で結果がよいものを解とする。

完成した解答は解答済みのものと比較し、有効回答の決定方法により良いと判断した場合提出する。

4. 開発環境

visual studio 2013 C++, OpenCV, PHP

33

このパズルを解きたい

—パズルだよ！絶対、パズルだよ！—

苦小牧

佐藤 陸(2年) 千葉 泰理(2年)
三浦 準也(2年) 三上 剛(教員)

1. アプローチ

・画像認識

渡された問題画像データを色の差異を認識して並べ替えを行い、原画像の状態に直す。画像を認識しづらい場合、手動で視覚的に並べ替えできるようにする。

・解答探索

認識した問題を複数のアルゴリズムを用いて、問題の最適解またはそれに近い解を導出する。問題によって、A*アルゴリズムや並列処理を用いたアルゴリズムなどから最適と思われるものを選択する。

2. 開発環境

開発言語：C#

開発環境：Visual Studio Ultimate 2013

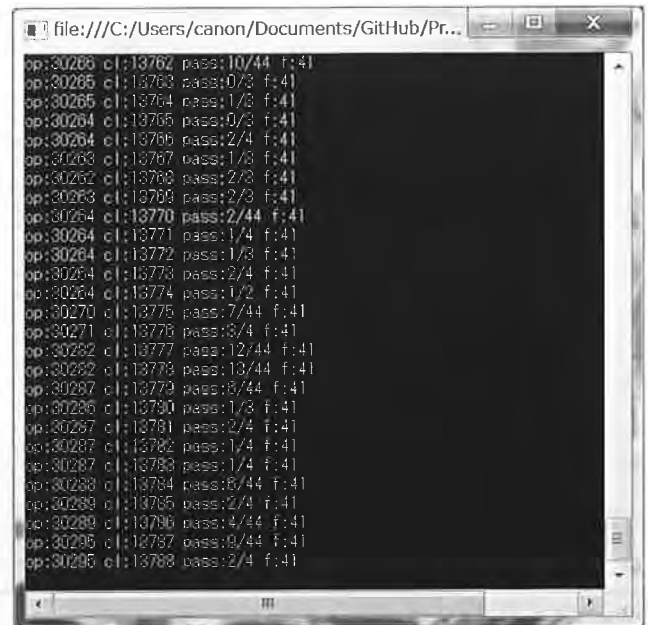


図1 実行時の画面

競技部門

34

復元レンズ

—信じて…ころろがときめいた瞬間を—

新居浜

吉田 晟佑(5年) 飛鷹 智浩(5年)
鈴木 龍斗(2年) 占部 弘治(教員)

1. アプローチ

本システムは大きく分けて2つの工程に分けた。1つは画像解析部、もう1つは選択・交換解析部とした。2つの工程の間にネットワークを介して処理を行う。以後、各解析部とネットワークについての説明をする。

2. 画像解析部

問題画像を原画像に復元する処理を行う。その際、隣接するピクセルの画素値の差分で評価する。各断片画像が移動した距離を計算し、以後画像より扱いやすい数値データを扱う。

3. 選択・交換解析部

実際に断片画像をどのように選択・交換を行うか解析を行う。その際、遺伝的プログラミングと遺伝的アルゴリズムを用いる。また、交換回数が増減に対応するために、可変長の遺伝的アルゴリズムを用いる。

4. ネットワーク

本大会のルールによると、プログラマブルな装置は3台まで用いる事が可能である。3台の装置をネットワークを通じて用いるため、2つの解析部のどちらかを並列で処理を行う事が可能となる。

5. 開発環境

Microsoft Visual Studio 2010
Xcode 5.1.1



1. アプローチ

提示された問題画像を、計算式で区切ることで分割したものと扱い、分割した画像を他の分割した画像と比較することで元の画像に復元する。

その後、復元した元画像と問題画像の位置を比較し、位置が異なっていた場合は異なっていた数に応じて回答を複数検出し、総コストの少ないものを選出する。

2. 画像の復元

分割した画像の境界部分の色を検出し、その境界部分に対応する他の分割した各画像の境界部分との色の距離の合計値を出す。それぞれの色の距離の合計値の中で最も距離に近いものを隣り合う画像とし、それを繰り返すことで復元する。最終的な判断は人が行う。

3. 並び替え

最大交換回数を考慮し、位置が異なっていた数に応じて用意されたパターンに当てはめて修正をしていく。この時点で最大交換回数を超えた回答は候補から取り除き高速化を図る。これによって絞り出された回答に選択コストのレートと交換コストのレートを掛け合わせ、総コストの少ないものを回答とする。

4. 開発環境

Visual Studio 2008 C++

Visual Studio 2010 C++

1. 設計方針

今回解決すべき問題を、「原画像を推測するアルゴリズム」と「選択・交換手順を探索するアルゴリズム」という2つの別の問題があるものとして認識する。

2. アルゴリズム

次の2つのアルゴリズムを有し、それぞれを並行に実行することによって計算パフォーマンスを高める。

2.1. 原画像推測アルゴリズム

2つの断片画像をすべての組み合わせで抽出し、上下左右の端の画素をユークリッド距離で調べて辞書とする。この辞書を用いて、断片画素がもっとも接続しやすい並びかたを求めた結果を原画像と推定する。

2.2. 手順探索アルゴリズム

このアルゴリズムは無限大時間を要すれば、解の1つに到達するものとする。まずは選択回数1回の場合に並び替えが完了する手順を探索し、それを元に選択回数と移動回数の調整を図る。

3. 処理の流れ

1. 試合開始直後、2つのアルゴリズムから1度目に返された解を、正誤に関わらずに提出する。
2. サーバから返却された誤答の数をういた上で、更に正しい原画像推測と手順探索を行う。
3. 2の操作を繰り返し、より良い解答を得る。

4. 開発環境

言語: C++14 ライブラリ: Boost, FLTK, OpenCV

1. 全体の流れ

問題を解くために、PC1 と PC2 の二台のノートパソコンを用いる。まず問題ファイルを PC1 で処理し、画像の解析終了後、PC2 へデータを送る。次に、PC2 で探索をし、答えを見つけ次第、PC1 へデータを送る。送られてきた探索結果を視覚化し、人の目で確認した後解答を提出する。

2. PC1 について

PC1 の役割は問題により与えられるデータを即座に処理し、画像の復元を行いそのデータを PC2 へ送信すること、UI を用いて問題情報の視覚化を行うこと、解答を提出することである。

与えられた問題データから元画像を全探索的に復元する。復元はピクセル単位で断片画像の RGB 値を照合することにより実現する。

UI を使うことで問題を処理していくさまざまな過程で人の手による介入を行うことができる。復元画像に間違いがあった場合に間違いを訂正する作業や探索の方向性を決めることなどが人の手で行えるようになっている。

3. PC2 について

PC2 の役割は PC1 から送られてきた文字データをもとに、貪欲法による全探索、もしくは枝刈り法による探索で、解答を求めることである。送られてきた元画像のデータと照合し、解答が不十分な結果であっても、PC1 に結果のデータを送信する。

4. 開発環境

Visual Studio 2010 C++

Emacs

Eclipse

1. はじめに

今年度の競技内容では、与えられた問題画像を原画像に復元するプロセスと、原画像に復元するために並び替えを行うプロセスがある。問題画像を原画像に復元するプロセスでは画像認識の精度の高さが、並び替えを行うプロセスでは探索の効率の良さが重要である。

2. 原画像の復元

画像を解析して復元する工程では、ある一つの断片画像に連結している断片画像を求めるといった動作を繰り返し行うことによって、原画像を求める。その動作を高速に行うために、次のような手法を採用する。

ある 2 つの断片画像同士が連続であるかどうかを判定する際に、各断片画像のエッジ部分だけを用いて連結性の評価を行う。

3. 並び替え

並び替えを行う工程では、問題の断片画像の数、選択コスト、交換コスト、時間コストによって選択すべき手法も変わる。この中でも、断片画像の数は探索に大きく影響するため、まず断片画像の数から探索アルゴリズムの決定を行い、選択コスト、交換コスト、時間コストによって探索アルゴリズムの細かな調整を行う。

4. 開発環境

言語: C,C++,Python

開発環境 Vim,Emacs

39

パズトラ！
—パズル・トライアル—大阪府立大
高 専三上 和馬(5年) 帖佐 克己(4年)
加賀 正樹(4年) 窪田 哲也(教員)

1. はじめに

この問題は、画像復元パートとパズルを解くパートという2つのパートに分けることができるので、それぞれ別の問題として検討した。

2. 画像復元パート

画像復元パートにおいては、自然画像が与えられるという仮定の元、適切な手法を用いて最も類似度が高い組み合わせを求めるアルゴリズムを考案し、実装した。

3. パズルを解くパート

簡単な考察の結果として、パズルを解くパートは、任意の位置を空きマスとするスライドパズル問題を複数回解く問題に帰着できるので、スライドパズルを解くアルゴリズムを検討した。我々が検討した最適解を求めるアルゴリズムは指数時間アルゴリズムなので、それらについてはキューリストティクスを用いて計算量を改善した。

4. GUI

C++CUI プログラムを実行し、その結果を受け取る C#GUI プログラムを作成した(図1)。



図1 GUI プログラム

5. 開発環境

言語: Japanese/JavaScript/C++/C#/Python/Ruby

ソフト: Windows 付属エディタなど

40

Puzzle And Procons II

岐 阜

岡地 涼輔(3年) 林 昌哉(3年)
山田 健登(3年) 廣瀬 康之(教員)

1. 基本構造

バラバラの画像から最終的に出来上がる画像を解析する。そして最終的な画像をもとに各画像を符号化した情報として扱い、各画像の位置を符号化情報としてパズルを解く。画像読み込み、符号化、出力部分と、パズル解析部分を分けて、より柔軟に対応できるように図る。

2. 画像の並べ替え

前述の、画像の位置を符号化した情報を用いて並べ替えを行う。位置符号は、最終画像を左上から通し番号を付けて各画像を呼称するものである。これを参照することにより課題を単純化し、数字を並べ替えることで、画像の並べ替えを再現することができるとともに、画像の処理と並べ替えの分断を図っている。

3. 試行訂正の処理

3.1 概要

試行処理を行う中で間違いが起こる可能性がある場所は、大きく二ヶ所考えられるため、個別に対応する。

3.2 画像の符号化の検出および訂正

最終的な画像を解析できた時点で、各画像を符号に合わせて、元の位置と最終的な位置に並べ替えた画像を比較表示することにより、未然に防ぐ。間違い部分を手動で訂正できるようにする。

3.3 並べ替えの試行および訂正

並べ替えは数値が正しく並ぶまで繰り返すため基本的に間違いは発生しない。しかし、解を見つけられない可能性がある為、通常の、元の位置までの距離を基準にする並べ替えの他、別の評価基準で並べ替えのモードを複数用意する。

4. 開発環境

Microsoft Visual Studio 2010

1. 問題解決への戦略

今回の問題は、与えられた画像から特定の手順で元の画像を復元するというものであり、これには二つの問題が含まれている。一つは、与えられた画像から原画像の復元であり、一つは与えられた画像から原画像を復元するための交換手順の探索である。これらを解決するプログラムを作成する。

2. 原画像の復元

問題では、あらかじめ断片画像がランダムにシャッフルされた画像が与えられ、どのような原画像であったかは与えられない。そのため、交換手順の探索を行う前に原画像を復元する必要がある。今回は各断片画像についてFFT等の画像処理を行い、各断片画像同士を比較することによって特定の断片画像の原画像内での位置を特定する。

3. 交換手順の探索

与えられた画像からもとの画像を復元する手順は、なるべく選択回数や交換回数が少ないほうが望ましいといえる。そこで今回は、与えられた画像から原画像を復元するための最短経路の探索を行う。ただしそのまま最短経路探索を行おうとすれば、膨大な状態数を探索することになってしまう可能性がある。そこで経路探索を行う際の評価値に手を加えることで、効率的な探索を実現する。

4. 開発環境

Microsoft Visual Studio 2013

OpenCV

1. アプローチ

原画像を推測する為に、ピースのフチ同士を比較し、評価値の大きいものから結合する。また、考えられる復元候補画像を何枚か生成し、その中から別の評価値または目視によって一枚を決定する。実際にピースを並べ替える際は、復元画像を元に起点となるピースを選択し、移動する。

2. 画像の復元

ピースのフチ同士を比較する。具体的には、隣接ピクセルの明度差の総和などを評価値として、フチの連続度を評価する。そして、評価値を降順に整列し、値の大きいほうから結合が可能な場合は順に結合していく。画像全体としての不整合を避けるために、結合するピースの順序を変えて復元候補画像をいくつか生成し、その中から一枚選ぶ。

3. 並べ替え

まず、それぞれのピースが元の場所から何マス離れているかを調べる。ピースはなるべく元の場所に近づくようにしながら動かしていく。

起点となるピースには、移動途中により多くのピースを元の場所に近づけることが可能と思われるピースを設定する。ピースが元の場所から何マス離れているかの情報も、この設定に利用する。起点は自動で判断するが、難しい場合は手動で設定することができる。できるだけピースを角から詰めながら画像を並べ替える。

4. 開発環境

Visual Studio 2013

OpenCV + OpenCVSharp

1. システム概要

本システムは、画像処理、画像解析、復元手順検索に分けられる。

2. 画像処理

サーバーもしくは USB メモリから問題画像をコピーする。コピーした画像ファイルを分割数に応じて分割し、独自の名前を付け保存する。画像解析しやすいように、二値化やグレースケール化などの画像処理を行う。

3. 画像解析

画像処理を行った分割画像ファイルを OpenCV を用いて画像内のオブジェクトの特徴を保存する。保存したオブジェクトの特徴から隣接した画像を検索し、元画像を復元する

4. 復元手順検索

復元された画像を元に復元手順を検索する。検索手法としては、外周から螺旋状に断片画像の位置を一致させていく。これを中央に残る四角形が十分小さくなるまで繰り返す。

その後、残りの四角形を専用のアルゴリズムで残りの復元手順を検索し、その結果を回答として決定する。

5. 開発環境

- ・ VisualStudio 2012
- ・ OpenCV 2.4.6
- ・ OpenCV 2.4.9
- ・ DxDLib

1. アプローチ

問題画像を解析して原画像を推定し、なるべく最短となる手順で並び替えて原画像を復元する。操作手順を回答として提出、サーバからの応答を元に間違い訂正を行う。各処理を切り離すことで並列して開発を行った。

2. 問題画像の解析

問題画像を読み込み、各断片画像の端の R, G, B の各色成分の差が最小となる組み合わせを求め、それを原画像とする。そして、想定した原画像の左上の断片画像から順に番号を割り当て、問題画像での断片画像の配置を割り当てた番号で表現する形式のデータを作成する。またこの時、間違い訂正を考慮していくつか原画像の候補となるデータを作成しておく。

3. 並び替え

先の処理で作成したデータを想定した原画像となるよ

うに並び替える。並び替えのアルゴリズムはメモリ効率がよく、解が存在する場合は最終的に必ず解へとたどり着けるグラフ探索アルゴリズムである反復深化深さ優先探索を用いた。また、分割数が十分に大きい場合は問題を整理していくつかの小さい問題に分割して処理する事で高速化を図った。

4. 間違い訂正

提出した回答が完全な復元画像でなかった場合、別の原画像候補を使用して並び替えと回答を行う。その際、直前の回答と比べて一致していない断片画像の数がサーバからの応答値と等しいものを選択する。

5. 開発環境

- Visual Studio 2010 C++
- Visual Studio 2010 C#

45

画像戦争

—んだよ、位置が分かんねえ—

阿 南

里谷 佳紀(4年) 東手 理貴(5年)
居石 峻寛(4年) 一ノ瀬元喜(教員)

1. はじめに

今回の問題を前半と後半の二つの領域に分割し、それぞれ異なるプログラムを開発する。前半は、復元画像を求めるプログラムで、後半は、最適な選択・交換の手順を求めるプログラムである。

2. 前半部分

問題画像のもととなった復元画像がどのような画像かを求める問題である。断片画像をつなげた時に、どの組み合わせが最も自然かを求める。自然さの指標は、各断片画像をつなげた時、その境界で隣り合うピクセルのRGB値の差をもとに算出する。

しかし、問題画像の情報のみから復元画像を予想しなければならないため、計算結果が間違っただけのものになる可能性がある。その場合は、人間の介入による計算結果の補正を

行い、精度を向上させる。

3. 後半部分

前半部分の結果をもとに、最適な選択・交換の手順を求める問題である。スライドパズルでおなじみ「15パズル」を解く要領で、最初に一つの断片画像を選択して、次々と交換を繰り返す。

問題によっては一度の選択で解くことができず、最大で二度の選択が必要となる。その場合は、一度、適当な断片画像を選択・交換して、あと一度の選択で解けるように断片画像の配置を変更する。

4. 開発環境

Microsoft Visual Studio 2013

Eclipse Luna

46

void swap (int, int) ;

—なぜかできない—

一 関

齋藤 真慶(4年) 佐藤 将太(3年)
堀金 周(2年) 管 隆寿(教員)

1. はじめに

解答の流れとして、問題画像を取得した後、問題画像を分割・解析し、現画像を推測する。そして、推測した現画像をもとに、問題画像を並び替える。

2. 原画像の推測

現画像の推測には、問題画像のヘッダに書かれている分割数とピクセル数をもとに、問題画像を分割し、分割した画像の各辺のRGB値の標準偏差を求める。

そして、求めた標準偏差どうしを比較し、値が近いものどうしを隣接した画像と判断し、原画像を推測する。

さらに、このアルゴリズムでは、近い色どうしであると誤判断が発生する可能性があるため、手動で並び替える機能も実装している。

3. 問題画像の並び替え

並び替えの方針として、時間コストがとても大きいと考えているため、なるべく速く原画像になるように並び替える。

並び替えの方法として、全探索を行う。しかし、全探索では、時間もかかり、メモリも大量に使用するため、反復深化深さ優先探索などのアルゴリズムを利用する。

さらに、枝切りなどの計算量を減らす処理を加えてある。

4. 開発環境

OS : Windows 7、Windows 8

言語 : C++、C#

フレームワーク : .NET Framework 4.5

IDE : Visual Studio 2013

Visual Studio 2012

47 高専のパズル概論 I

長野 原 大樹(3年) 柄澤 駿太(2年)
高野 航雅(1年) 伊藤 祥一(教員)

1. アプローチ

問題画像を受け取り、画像解析により、その画像を復元し、断片画像がどの位置なのかを調べる。その位置を元に断片画像を並び替える。

2. 画像解析

受け取った画像を、指定された数に分割し、色の傾向ごとに分類する。断片画像を1つ選択してその画像に色の傾向が近いものから、左右につながるか試していき、帯を作っていく。帯がひとつ完成したら、次の断片画像を選択し帯を作っていく。必要な数の帯が完成したら、帯を縦方向につなげていき画像を完成させる。

3. 画像の並び替え

A* (エイスター)探索アルゴリズムを用いて探索する。評

価関数にマンハッタン距離の和を用いて、盤面を評価し、コストが低くなるように探索していく。訪れた盤面を保存し、同じ盤面に訪れた時、評価コストが低いなら、再度利用することまた、簡単な盤面の場合は予め解いておくことで、高速化を図った。

4. 間違い訂正について

人間が断片画像を見て、直接修正できるようにすることで間違いに対応する。

5. 開発環境

Visual Studio 2013 C#

OpenCVSharp

Visual Studio 2013 C++

48 新委員長就任 —参勤交代2回目までですか—

沖 日熊 悠太(専1年) 西原 希咲(4年)
渡嘉敷真優(3年) 正木 忠勝(教員)

1. はじめに

今回の競技部門の最大の鍵は、与えられた各問題画像を迅速に復元して、それぞれに適した画像解析アルゴリズムを選択することである。選択する画像解析のアルゴリズムの適性により最終的なコストが大幅に異なる。よって、様々なタイプの問題画像に適応したアルゴリズムを開発した。

2. 問題へのアプローチ

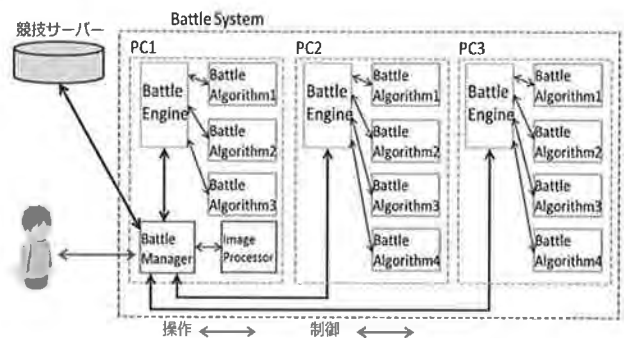
2.1 元画像復元

競技サーバから問題フォーマットを BattleManager が取得し PC1 に問題フォーマットを渡す。ImageProcessor で画像処理を行い断片画像を並び替え原画像を復元し取得する。

2.2 総コストを求める

複製した原画像を元に、人間が各 PC ごとに実行する

BattleAlgorithm を選択する。そのなかから成功した順に競技サーバに回答フォーマットを提出する。



3. 開発環境

コンパイラ: VisualStudio

言語: C++/C#/Python

49 MasterPiece

佐世保

城間 優作(4年) 田丸 将寛(4年)
吉田 愛永(3年) 手島 裕詞(教員)

1. システム

さあ解を導け!! YES⇒幾つもの断片達
正しい元の居場所へ、
さあコマを動かせ!! YES☆たったひとつだけの正解
美しい元の画像へ
MOVE 復元するから SOLVE 解き明かすから
新しい解を求めよう!!
解を初めて求めて 今日までどの位経っただろう
ずっとひとつひとつ繋げよう
解は画像を並べる為に 求めた答だから
きっとこの答が 私の MASTERPIECE

2. アルゴリズム

ねえ… 問題画像を
一度… 復元画像にしてから

場所を特定 そしてよろしく!
COST 交換重視で LINE まとめて動かす
正解の近づく方へ進め ALL PIECE!!
復元画像がどうなるか 誰だって解らないけれど
それは問題画像で決まる事
解は手動で求めるモノじゃなくソフトで求めるモノ
それが答えになる 私が MASTERPIECE

3. 開発環境

DEV 作りたいから ENV 揃えた環境
元々の画像作ろう Visual C++(+) C#!!
開発を始めてから 今日までどの位たっただろう
ずっと一日ずつ組み上げよう
これは画像を並べる為に 生まれたソフトだから
きっとこのソフトで 私の MASTERPIECE

50 Rest In Piece(仮)

都立(荒川)

金子 孝輝(5年) 蓑手 智紀(5年)
濱本 海(5年) 鈴木 弘(教員)

1. 概要

問題解決の手段として主に遺伝的アルゴリズム(以下GA)を用いることとした。GAとは、あるデータを遺伝子情報と見なして交叉・突然変異を行い、問題の最適解を導出する方法である。

2. システムの詳細

2.1 全体の流れ

本システムはまず、問題画像から原画像を導出することを行い、次にその過程を求めるといったアプローチを行い、回答を導き出している。

2.2 原画像の導出について

まず、分割されたピースそれぞれの特徴色を求め、色相からそれぞれのピースの大体の位置を決める。ピースを並べて出来たひとつの画像に対して色の微分と、輪郭線の導出を行い、それぞれの連続性からその画像の尤度を計算する。並び換えを繰り返していき、うち最も尤度の高かった並びを目指すべき並びとする。

2.3 ピースの操作手順について

問題画像から 2.2 項により求めた並びになるような選択・交換の手順を求める。まず乱数によって生成した手順データを遺伝子個体と見なし、この個体を一定数生成する。各個体に選択コスト・交換コストを用いて評価を行い、評価の高い順にソートする。評価の高い個体同士に対して交叉・突然変異の処理を行い、子個体を生成し、評価を行う。評価の高い順から一定数を次の世代に加え、世代交代を行う。以後これを一定世代数繰り返し、そのうち最も評価の高かった操作手順を回答とする。

3. 開発環境

Windows 7 / Mac OS X 10.9 Mavericks
VC++ / Gnu Compiler Collection (C++コンパイラ)

1. システム概要

「正解画像の復元」と「ルートの検索」の二つに分けて開発した。

1.1. 正解画像の復元

正解画像の算出には画像一つ一つのピクセルの RGB を三次元ベクトルとして扱うこととした。隣り合うピースの内積を計算し、そこから色の類似度を算出する。その類似度を基に尤もらしいピースの組み合わせを推定し、正解画像の復元を行う。

1.2. ルートの検索

今回の競技での勝敗の判断で一番に優先されるのは回答までの時間である。そこで交換コストや選択コストの削減は考えず、何よりも回答が算出されるまでの時間の削減

を主軸において作成した。また、原則として一つのピースだけを移動させ、正解の並びを復元する方法を考えた。

最初は正解画像で右下に存在するピースを選択し、そのピースを選択したまま下二行以外を全てそろえる。その際の移動方法はパターン化しているため、共通の手順を踏むことで正解の並びを復元する。

最後の二列は左側からそろえ、右下に 2×2 だけを残す。その場所をそろえるには 12 通りの場合がある。そこで、それぞれの場合に対する解法を個別に適用できるようにした。

2. 開発環境

Visual Studio 2010

gcc4.5.2

jdk1.7.0

1. アプローチ

問題画像から画像処理プログラムを用い正解画像を復元する。その後、パズル回答プログラムによって正解画像に最も早く到達できるような移動プロセスを導出する。パズル回答プログラムには A*アルゴリズムを利用し、計算量を最小限に抑え高速化を図っている。

2. 画像処理プログラム

エッジ処理により画像片毎の輪郭の検出を行い、複数の画像片を組み合わせ、接続点について色の一致度を検出する。最も一致度が高かったものを正しい画像片の組み合わせとして扱う。精度を上げかつ効率を高めるため、画像の復元を最適化問題として GA を使用している。また、評価関数には sobel, DCT, MSDS を使用している。

3. パズル回答プログラム

今回の競技のパズルの移動パターンは無尽蔵に存在するため無駄な計算を削っていく工夫が求められた。本プログラムで用いた A*アルゴリズムは、最短コストがわかっている場合少ない計算量で最適解を求めることが出来るアルゴリズムである。これを利用する事により、1回の移動毎の最適解を求めるだけで良くなり、計算回数を大幅に削減することが出来る。本プログラムでは最短コストとして最大の選択コストと問題パズルのマンハッタン距離の和を用いている。

4. 開発環境

Visual Studio 2010 C++

OpenCV

1. 原画像の復元

各断片画像を管理するため、問題画像上での座標(00~FF)をIDとして割り振る(図1)。

1.1 断片画像の連続性を評価

断片画像同士の対応する辺に対し、画素毎のRGB値の距離の絶対値を評価値として求める。

1.2 原画像の復元

中心となる断片画像を推定し、断片画像の四辺と隣接する断片画像を探索し、接続する(図2)。

接続された断片画像に対し、各辺と隣接する断片画像の選択・配置を繰り返すことで原画像を復元する。

2. 問題画像の並び替え

① 各断片画像に対し、原画像上での位置との距離(行と列の差の絶対値の和)を計算する。

② 距離の大きい断片画像を優先して選択し、「各断片画

像を原画像の位置に移動するのに必要な距離の合計」を基に原画像の位置へのライン情報を評価する。

③ 幅優先探索で、評価値が最小となるライン情報を基に画像の並び替えを行う。

④ 原画像を抽出されるまで①~③のステップを繰り返し、得られたライン情報を解答とする。



図1 断片画像のID



図2 復元手順

1. アプローチ

画像の境界部分で相関係数を計算して、それに基づいて断片画像を配置する。その後で人間が修正を行うことで画像の元の配置を求める。パズルは端から順番に揃える方法を用い、計算の高速化を図った。

2. 画像処理

2.1 相関係数

画像のつながりを判断する方法として、相関係数を用いる。色空間は、問題に合わせて、RGB, HSV を使い分ける。

2.2 相関係数

画像は相関係数の大きい断片画像から順番に並べる。このとき、長方形に並ばないことがあり得るので、人間がGUIで修正を行う。

3. パズル

端から揃える方法を用いて、問題をより小さな問題とみなせるような形に並べ替えながら解いていく。 $m \times n$ の問題を $2 \times n$ にする解法と、 $2 \times n$ の問題を 2×2 にする解法、 2×2 の問題を完成させる解法、 $1 \times n$ を完成させる解法、を組み合わせることで問題を解く。

4. 開発環境・言語

Java

Python

OpenCV

PyCharm

NetBeans

<問題発行>

1. 元画像復元用のインターフェースを作成し、問題画像を半自動で元画像に復元し、断片画像がどのように移動しているかを判別する。

<解答探索>

2. 判別された結果から以下の手法を用いて、元画像を復元するための手順を導出する。

2.1 各断片画像が元画像のどこへ向かえばいいのかについて、マンハッタン距離の総和を算出する。これにより、最低限の移動手数が分かる。

2.2 導出したマンハッタン距離の総和をヒューリスティックとして利用し、反復深化を用いて最短解を

求める。このとき、8パズル程度の状態数の問題についてはあらかじめ解答データベースを作って検索・応答することで時間コストの低減を図る。もし、計算量が膨大な16*16といった問題が出題された場合、本プログラムでは制限時間内の解答が困難と考えられるため、人力で復元率を高められるよう、手動復元用のインターフェースも用意する。

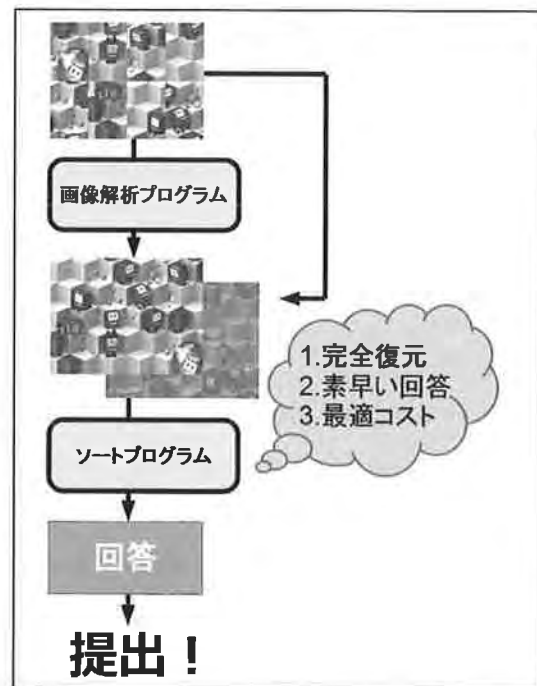
<解答提出>

3. 得られた回答を送信用のプログラムを介して直接サーバーへ送信する。

1. システム概要

まず、各断片画像の辺のピクセル間の色を基本として、各断片画像間の関係性を見出し原画像に近づけていく画像解析プログラムによって問題画像から原画像を割り出します。

次に、原画像と問題画像をソートプログラムによって比べ復元を行います。確実に完全復元画像作することを念頭に置き、問題画像に示されているコスト値から最適で、かつ速くソートできる回答を提出します。ここで最適とは選択コスト、交換コストの合計値が小さいものを指します。回答は、第一に確実に完全復元画像を作ること。次に短い時間で回答できること。最後に最適なコストであることを優先順位として掲げます。回答が確定次第、提出プログラムによって提出します。



システム概要図

1. アプローチ

出題された問題画像より画像処理プログラムで原画像を復元し、断片画像の配置を数字の並びとして出力する。並び替えプログラムで何通りかの解を求め、その手順を出力する。処理されたものをその都度、提出プログラムで提出する。一連の処理は自動化し、人手を介さない。

2. 画像の復元

画像の復元は画像が単純なものであるため、バイナリデータを直接に読み込んで行う。まずは問題画像を断片画像ごとに分解する。断片画像は各辺の縁の色情報に着目して、他の断片画像の各辺との色差を求めることにより復元を試みる。復元された原画像は目視でチェックできるようにする。また原画像が誤っている場合、あるいは復元できなかった場合に、手動に切り替えて復元を試みる。

3. 並び替えの手順

画像の並び替えは複数のアルゴリズムを併用し、それぞれで効率的な手順が求めればその都度出力する。アルゴリズムとしては、最適解を目指して定石通りに手順を分解して幅優先探索などを用いるものと、速さを重視して状態に評価値を与えることによって深さ優先探索などを用いるものと、大きく二つの方法に分けられる。

4. ファイルの提出

ファイルの提出は並び替えが完成し次第自動的に行う。

5. 開発環境

Windows7 Professional Edition

Visual Studio 2010

OpenCV 2.4

1. システム概要

本システムは二つのプログラムに大別される。1つは完成画像の生成で、もう1つはそれに対する経路探索である。

2. 方針

2.1 完成画像の生成

断片画像の位置の特定には様々な画像処理を施して解読にあたる。具体的には、問題画像にBGR比較、Gray scale化、Edge検出、2値化処理などを加え(今後、処理のバリエーションは増える予定)、その結果から断片画像の4辺のそれぞれで1画素ずつ相互に比較し、確率的に位置を判別する。

2.1 ルートの決め方

探索アルゴリズムとして採用予定しているのはダイクストラ法である。これに対していくつかの探索上のルールを加えて、選択回数以内の経路を割り出す。具体的には、断片画像の元の位置からの距離で、ルールに則った最短距離を計算する。断片画像の数が多くなるにつれて処理が重くなるデメリットはあるが、並行処理を用いて解消する。

3. 開発環境

- OS: Windows7, Mac, Linux debian
- 開発言語: C++
- 開発環境: Visual Studio 2013, Vim
- 外部ライブラリ: OpenCV 2.x, OpenCL

1. 概要

問題となる画像を取得し、それに処理を行って原画像を推測したうえで復元手順を探索、解答する。

2. 原画像の推測

画像解析ライブラリである OpenCV を用いて問題画像の境界を求め、断片画像を導き出す。一つの断片画像に注目し、断片画像同士の境界面の差分をもとに隣り合う断片画像を探し出す。その差分が小さい時、正しい組み合わせであるとして、この手順を繰り返し、原画像を推測する。また、この時点で間違った原画像であるかを使用者が判断・修正を行えるようにしておく。

3. 復元手順の探索

それぞれの状態での断片画像の位置から、原画像での位置

までの距離を評価値として用いて、各断片画像の評価値の総和を求め、これが0になるように入れ替えを行っていく。状態として操作履歴、状態推移として断片画像の選択と交換とする幅優先探索を用いることによって、選択コストと交換コストが少なく、かつ評価が高くなるように探索を行う。

4. 開発環境

Visual Studio

Xcode

OpenCV

C++

1. 問題のアプローチ

画像処理によって分割画像の本来の位置を判定するプログラム、最短時間ソルバー、最適解を求めるソルバー、これら3つを使って素早く分割画像の本来の位置を特定し、最適解を導く。

2. 画像処理

分割画像端の類似度情報をマッチングさせて、問題画像の各分割画像が、原画像のどの部分にあたるかを推定する。また、一度答えを送信したときの不一致数を用いて精度を上げる。

3. 最短時間ソルバー

トップダウン型貪欲法を用いて、実行時間の短さを優先した復元を行う。3*3の状態まで帰着できるので、3*3の

最短復元情報を保持しておき、その情報を使って完全復元する。

4. 最適解ソルバー

メタヒューリスティック的手法を用いて、なるべく手数が短くなるような復元を探索する。2. 3. を繰り返し適用し、分割画像の本来の位置が確定したところで、このソルバーに切り替える。時間をどれだけ残しておけるかが勝負の鍵となる。

5. 開発環境

Visual Studio 2013 C++

1. アプローチ

課題を原画像の推測部分と、推測した原画像への交換操作を求める部分に分けて考え、それぞれにおいて貪欲法のアルゴリズムを適用し、時間内に精度の良い解を求めることを目指した。

2. 原画像の推測

1つの断片画像を選択し、残りの断片画像の中で評価関数の値の小さいものを結合していき、原画像を求める。そのとき、 n 個結合する組み合わせの中で有望な候補のみを用いて $n+1$ 個結合する組み合わせを求めることで効率の良い探索を目指す。はじめは縦方向のみ分割数になるまで結合し、その後横方向について分割数になるまで結合という順で行うことにより結合後の画像が四角形になることを保証する。

3. 交換操作の算出

A^* アルゴリズムを用いて、交換操作を求める。ヒューリスティック関数は、推測した画像と問題画像での各断片画像の位置の差(マンハッタン距離)の合計値の関数とした。関数内で交換コストと選択コストを用い、交換コストと選択コストを考慮した探索が行えるようにした。

4. 人力修正処理

プログラムによる原画像の推測の終了後に、人間の目で正しいかどうかを確認し、手で修正が行えるようにした。

5. 開発環境

Windows, Mac, Linux

C++, OpenCV, Boost

Visual studio 2013, gcc, Git

1. 元画像の推定

断片画像同士の距離を定義し、その総和をなるべく最小化する断片画像の組み合わせを見つけることで元画像を推定する。距離としては、断片画像の色空間を適切に変換した上での Euclid 距離や Mahalanobis 距離などを用いる。

2. スライドパズル

本問題は、与えられる任意の問題を完全復元することが可能であるので、完全復元を目指す。具体的には、適切な評価関数とメタヒューリスティックアルゴリズムを用いて解空間の探索を行う。

2.1 評価関数

各断片画像の現在の位置と正しい位置との Manhattan 距離の総和や、残りの選択可能回数などを用いて、現在の盤面の良し悪しを数値で定量的に評価する。

2.2 メタヒューリスティックアルゴリズム

様々なメタヒューリスティックアルゴリズムが存在するが、今回は最良優先探索、ビームサーチ、反復深化 A^* アルゴリズムなどを併用する。

3. ユーザインタフェース

C++用のクロスプラットフォーム UI フレームワークである Qt を用いることで、様々な環境で動かすことのできる GUI を作成した。

4. 開発環境

言語 : C++1y

コンパイラ : clang, gcc, icc

ライブラリ : boost, Qt

エディタ : Sublime Text, Qt Creator, vim

1. Solution idea

We divide the problem into 2 parts. Firstly, our program *locates the correct position of each piece*. Then, we apply several algorithms for *moving pieces to their correct positions*.

2. Locating correct positions of pieces

First, a compatibility function is computed to measure affinity between two neighboring parts. For each way to rearrange these pieces, we call its dissimilarity value as the sum of compatibility value (which is calculated by compatibility function) between every pair of adjacent pieces. Then the solver rearranges the pieces to get the result that has the dissimilarity value as small as possible. To solve the problem, there are two main algorithms:

- Greedy Algorithm
- Genetic Algorithm

3. Movement to the correct positions of pieces

We apply various strategies to solve this problem. After each step, we smaller the problem (by using greedy algorithm such as Parberry or Snake-based algorithm) and when it is small enough, A* search is used to obtain optimal solution.

In addition, we use number of tricks to speed up and smaller answer.

4. Programming language

Our team uses C++ and C# to design the project. Our program is written in .NET framework 4.5.1 and runs in Window 8.1 OS.

1. Introduction

The jigsaw solver is written by Python and Rust. Image Processing module is written by Python language with Numpy library, and Matrix Solver module is written by Rust that is a new Programming language.



Figure 1. Randomly Problem Image



Figure 2. Restored Image

2. Modules

2.1 Image Processing

The input of the program is a problem image. The program will restore the original image using algorithm, and output the matrices of problem image as well as the original image in order to be processed by matrix solver.

According to the image information, our Algorithm is to cut problem image into small pieces. By comparing the gray value differences of the edge of small pieces, we will restore the original image.

For accuracy, the results require the operator to manually inspect and modify what.

2.2 Matrix Solver

The Input of Matrix solver is the output of the image processing module.

By using a search algorithms, Matrix solver continuously selects and exchanges problem matrix until it becomes the original matrix.

To solve the problem that the state space is too large, we choose a heuristic search algorithm. Also, we use parallelization techniques to optimize the processing speed.

1. Introduction

Our program consists of two parts. First part finds correct location for each piece. Second part solves the puzzle itself.

2. Finding correct locations

First, our program take input ppm picture file and description then divide the picture to many pieces. Solve the puzzle using special algorithm to get original picture. Find position of pieces of input picture comparing input picture and original picture. First step output consists of text file that position of piece number.

3. Solving the puzzle

Find optimal position of carry piece from first step output. Then take first and second position for every piece and

distances from the carry piece. First drag carry piece to reposition corner pieces and edge pieces using the shortest path algorithm. Answer is a string that consist of every movement of carry piece. If it is impossible to move, find next carry piece and repeat pervious step.

1. Introduction

This Manuscript is proposed for The 25th Annual Programming Contest Competition Section “Find Your PIECE of Mind II”. The Objective of this competition section is to correct the random puzzle picture from random assignment picture. Therefore, we propose the Hybrid Algorithm based on Greedy Algorithm to use for finding the solution. Our design program for this contest is developed by C# language. This program will generate the correct answer from original and random picture by header condition assignment.

2. Proposed Method

The proposed method is proposed as Fig.1. From Fig.1 we start the program by input the Assignment to find the correct position. Then, the program will use Hybrid Algorithm for selecting and swapping the pieces of picture to correct the

position. The program will iterate the algorithm when the number selection much more than assignment condition. Finally, the correct answer is shown when the program has stopped.

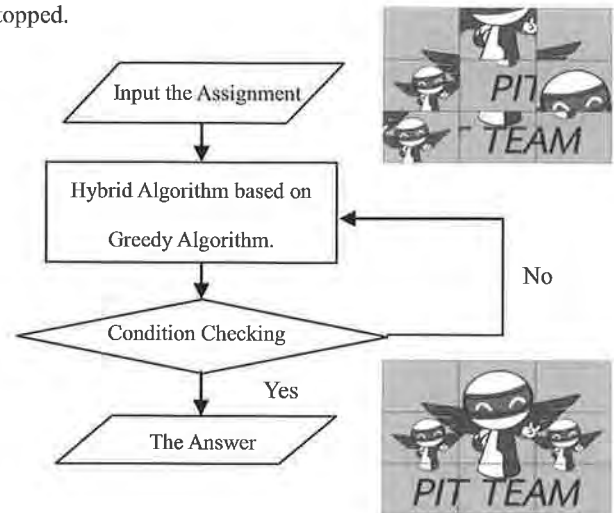


Fig.1 Flow chart of proposed method.

協賛企業広告一覧

第25回プログラミングコンテストでは、全国の企業より多くのご支援をいただきました。衷心より厚くお礼申し上げます。

(敬称は省略させていただきました。なお、数字は広告掲載ページです。)

【特別協賛】

東芝ソリューション(株)	104-105	(株)トヨタコミュニケーションシステム	114-115
アイビーシー(株)	106-107	(株)ドリーム・アーツ	116-117
(株)インテリジェントウェイブ	108-109	ネクストウェア(株)	118-119
さくらインターネット(株)	110-111	富士通(株)	120-121
(株)シーエーシー	112-113	(株)ブロードリーフ	122-123

【一般協賛】

アイフォーコム(株)	124	ソニーデジタルネットワークアプリケーションズ(株)	131
インテル(株)	125	(株)BCN	132
(株)NTTPCコミュニケーションズ	126	(株)フォーラムエイト	133
(株)NTTぷらら	127	富士ソフト(株)	134
(株)オプティム	128	(株)豆蔵ホールディングス	135
(株)ぐるなび	129	(株)メンバーズ	136
セイコーエプソン(株)	130	ヤフー(株)	137

【広告協賛】

(株)アットウェア	138	(株)イーセクター	149
アラクサラネットワークス(株)	139	(株)インフォクラフト	149
エクストリーム ネットワークス(株)	140	(株)ヴァル研究所	150
(株)jig.jp	141	木村情報技術(株)	150
(独)情報処理推進機構	142	国際通信企画(株)	151
(独)情報通信研究機構	143	(株)ソルコム	151
チームラボ(株)	144	(株)ファインディックス	152
(株)デザイン・クリエイション	145	(株)SIエージェンシー	153
日本データパシフィック(株)	146	サイバー大学	153
パナソニック インフォメーションシステムズ(株)	147	蛇の目 ミシン工業(株)	153
(株)ムロドー	148		

【メディアスポンサー】

週刊BCN	154	岩手日日新聞社	154
-------	-----	---------	-----

大会役員・プロコン委員・主管校実行委員会・事務局等

大会役員

大会会長	一般社団法人 全国高等専門学校連合会会長	古屋 一仁	東京工業高等専門学校校長
副会長	一般社団法人 全国高等専門学校連合会副会長	田原 正夫	東京都立産業技術高等専門学校校長
副会長	一般社団法人 全国高等専門学校連合会副会長	小島 知博	サレジオ工業高等専門学校校長
副会長	一般社団法人 全国高等専門学校連合会副会長	新田 保次	鈴鹿工業高等専門学校校長
副会長	特定非営利活動法人 高専プロコン交流育成協会理事長	神沼 靖子	一般社団法人情報処理学会 フェロー
副会長	第26回大会次期主管校校長	黒田 孝春	長野工業高等専門学校校長
副会長	第25回大会主管校校長	柴田 尚志	一関工業高等専門学校校長

プログラミングコンテスト委員会

委員長	柴田 尚志	一関高専	校長
副委員長	長尾 和彦	弓削商船高専	情報工学科教授
副委員長	嶋山 喜彦	一関高専	副校長・学生主事 一般教科人文社会系教授
ブロック委員	佐竹 利文	旭川高専	システム制御情報工学科教授
ブロック委員	千田 栄幸	一関高専	電気情報工学科准教授
ブロック委員	松林 勝志	東京高専	情報工学科教授
ブロック委員	柴田 博司	富山高専 (本郷)	電気制御システム工学科教授
ブロック委員	片山 英昭	舞鶴高専	電気情報工学科教授
ブロック委員	河野 清尊	米子高専	電子制御工学科教授
ブロック委員	今井 一雅	高知高専	電気情報工学科教授
ブロック委員	松野 良信	有明高専	電子情報工学科准教授
企業委員	久保 慎一	ネクストウェア株式会社	
企業委員	田中 達彦	日本マイクロソフト株式会社	
専門委員	今井 一雅	高知高専	電気情報工学科教授 [兼任]
専門委員	吉成 偉久	茨城高専	電気電子システム工学科准教授
専門委員	金寺 登	石川高専	電子情報工学科教授
専門委員	田添 文博	鈴鹿高専	電子情報工学科准教授
専門委員	宮下 卓也	津山高専	情報工学科准教授
専門委員	寺元 貴幸	津山高専	情報工学科教授
専門委員	佐藤 秀一	長岡高専	一般教育科教授
専門委員	井上 泰仁	舞鶴高専	電気情報工学科准教授
専門委員	小保方 幸次	一関高専	制御情報工学科准教授
専門委員	中道 義之	沼津高専	電子制御工学科非常勤講師
専門委員	山下 晃弘	東京高専	情報工学科助教
専門委員	江崎 修央	鳥羽商船高専	制御情報工学科准教授
専門委員	鈴木 宏	長野高専	電気電子工学科教授
専門委員	松林 勝志	東京高専	情報工学科教授 [兼任]
専門委員	小嶋 徹也	東京高専	情報工学科准教授
専門委員	山崎 誠	長岡高専	電気電子システム工学科教授
主管校実務委員	千田 栄幸	一関高専	電気情報工学科准教授 [兼任]
主管校実務委員	菅 隆寿	一関高専	電気情報工学科准教授
主管校実務委員	佐藤 陽悦	一関高専	制御情報工学科准教授
主管校実務委員	照井 教文	一関高専	物質化学工学科准教授
有識者委員	松澤 照男	北陸先端科学技術大学院大学	理事・副学長
有識者委員	伊原 充博	一般財団法人鯨洲会	常務理事
次年度主管校委員	鈴木 宏	長野高専	電気電子工学科教授 [兼任]
前年度主管校委員	佐竹 利文	旭川高専	システム制御情報工学科教授 [兼任]

主管校実行委員会 (一関工業高等専門学校)

委員長	柴田 尚志 (校長)
副委員長	嶋山 喜彦 (副校長・学生主事), 佐藤 昭規 (副校長 (総務担当)), 竹原 克郎 (事務部長)
事務局次長	菅 隆寿
【総務】	明石 尚之, 中山 淳, 竹原 克郎, 加藤 博, 井上 翔, 豊田 計時, 谷林 慧, 河原田 至, 佐々木 晋五, 高橋 道康, 津田 大樹, 田中 邦延, 佐藤 良紀, 及川 尚, 尚, 千葉 知美, 本明 昇, 阿部 恵悦, 佐藤 克八, 高橋 翠
【救護・学生交流企画】	嶋山 喜彦, 原 主祐, 千葉 正義, 津田 大樹, 平野 悦子, 田中 邦延, 佐藤 良紀
【受付】	松本 幸二, 古本 猛憲, 佐藤 克八, 高橋 裕子, 松浦 千春, 関根 孝次, 渡辺 仁史, 及川 尚, 千葉 知美, 本明 昇, 三浦 美鈴, 佐藤 美鈴, 都川 匡史, 貝原 巴樹雄, 中村 美幸
【応接・案内】	佐藤 昭規, 中嶋 剛, 都川 匡史, 貝原 巴樹雄, 中村 貴, 福井 ゆきの
【式典】	二階堂 満, 大嶋 江利子, 大山 稔哉, 中川 裕子, 菅野 俊郎, 戸谷 一英, 佐藤 亮二, 福井 ゆきの, 中村 貴
【輸送・弁当・駐車場】	芳川 直哉, 高橋 寛子
【広報・記録】	白井 仁人, 片方 江, 佐藤 哲郎, 高橋 知邦, 伊藤 瞳, 熊谷 健一, 加藤 卓也, 高園 恵理子, 菊地 雄輔, 佐藤 昌也, 和田 史明, 若嶋 振一郎, 都川 匡史, 澤瀬 薫, 小林 健一, 川合 政人, 滝渡 幸治, 梅野 善雄, 菊地 雄輔
【国際交流】	佐藤 千葉, 和久 八戸, 下川 理英, 大山 稔哉, 佐藤 説夫, 梁川 甲午, 二本柳 譲治
【会場設営】	高橋 龍也, 高橋 龍也
【課題・自由部門】	佐藤 陽悦, 福村 卓也, 三浦 正治, 鈴木 明宏, 千葉 悦弥, 小野 孝文, 小野 宣明, 小岩 俊彦, 佐藤 昌也, 佐々木 亨, 宇野 修子, 高嶋 あつ也, 太田 敏貴
【競技部門】	小保方 幸次, 照井 教文, 三浦 正治, 村上 明, 柴田 勝久, 三浦 弘樹, 高野 淳司, 藤田 実樹, 佐藤 昌也, 和田 史明, 横田 礼, 中村 直也
【学校企画】	郷 富夫, 秋田 敏宏, 佐藤 哲郎, 平林 一隆, 阿部 恵悦, 三浦 美鈴

大会事務局・委員会事務局

〒021-8511 岩手県一関市萩荘字高梨
一関工業高等専門学校 第25回プロコン委員会事務局 学生課学生支援係
Tel:0191-24-4718 Fax:0191-24-4530 E-mail:jimu25@procon.gr.jp

高専プロコン交流育成協会 (NAPROCK) 事務局

〒541-0057 大阪府大阪市中央区北久宝寺4-3-11 ネクストウェアビル
Tel:06-6281-0307 Fax:06-6281-0318
担当 木戸 能史 NAPROCK事務局長
田中 佐和子 NAPROCK事務局次長

全国高等専門学校 第24回プログラミングコンテスト

平成25年10月13日(日) - 14日(月)

主管校：旭川工業高等専門学校 会場：旭川市民文化会館



全国高等専門学校 第25回プログラミングコンテスト

平成26年10月18日(土) - 19日(日)

主管校：一関工業高等専門学校 会場：一関文化センター



岩手県一関市大手町2-16 TEL: 0191-21-2121