

ourcam:

デジタル写真撮影のためのオンサイトプログラミング環境

大島遼^{†1} 笥康明^{†2}

近年、デジタルカメラやスマートフォンの普及に伴い、日々多様な画像が撮影され、さらにアプリケーションを介して加工・共有されている。このような、様々なデジタルカメラやアプリケーションの登場によって写真の歴史がまた大きく塗り替えられようとしている。写真の歴史において、新しいイメージを生み出すための方法論および写真のメディア特性の探求がなされてきた。そして、技術史的な発明と並行して、作家各々が発見・発明した新たな「写真のつくり方」が提案されて、写真の歴史が更新されてきた。このような背景のもと、本研究では、カメラ自体にコンピュータが内蔵され、ネットワークに接続されることを前提として、撮られた写真をネットワーク上で共有するだけでなく、ユーザが「写真のつくり方」を撮影時に試行錯誤することを可能にし、その「つくり方」自体をネットワークを介して共有することを考える。具体的には、デジタルカメラ上でのビジュアルプログラミング言語、プログラムの共有変更機能を有するモバイル上の統合開発環境 ourcam を提案する。本環境により、デジタル・フォトグラフィ特有の撮影手法や、デジタルカメラ上におけるメディアの構築方法が蓄積され、ユーザが任意の場所で思いついたアイデアをその場でプロトタイピング可能とする。また、本環境を用いたユーザスタディによって、ユーザ自身によって「写真のつくり方」が構築可能となったデジタル・フォトグラフィ環境における撮影行為の変容を示す。

ourcam:

On-Site Programming Environment for Digital Photography

RYO OSHIMA^{†1} YASUAKI KAKEHI^{†2}

In these days, the spread of digital cameras and smart phones, a large amount of pictures are taken in our daily lives. Moreover, many of the pictures are processed and shared via various applications. In the history of photograph, photographers and engineers always search for ways to make novel images. In this research, we propose a new method of taking photograph with a camera, which has the computer in itself and is connected to the network. This camera enables users to make "the way of making photograph" with repeated trials and errors on-site, and to share not only the photo itself but also the "way of making photographs" through the network. More concretely, this paper proposes an integrated development environment working on the mobile phone environment named "ourcam", which has a visual programming environment and a program sharing function. This environment stores specific techniques of shooting on digital photography and the way to build the media on the digital camera. It enables prototyping on the fly idea that the user has come up at any location.

1. はじめに

近年、デジタルカメラやスマートフォンの普及に伴い、日々多様な画像が撮影され、さらにアプリケーションを介して加工・共有されている。例えばInstagram[1]は、iPhone[2]、Android OS[3]のスマートフォン上で動作するアプリケーションである。撮影した写真に対してさまざまなデジタル画像処理フィルタを適用することで、既存のカメラやフィルタで撮影された画像を模倣したような質感のデジタル画像が得られ、それをソーシャルネットワーク上にアップロードすることでコミュニケーションを行うことができる。

このようなサービスの中核となるのは写真の加工機能である。誰でも「それなり」の写真が撮れるようになったことが、写真や画像によるコミュニケーションのハードルを下げ、サービスとして成立させている。このように、様々

なデジタルカメラやアプリケーションの登場によって写真の歴史がまた大きく塗り替えられようとしている。

写真の歴史において、新しいイメージを生み出すための方法論および写真のメディア特性の探求がなされてきた。そして、作家各々が発見・発明した新たな「写真の作り方」が提案されてきた。例えば、1830年代にPhotogenic drawing [4]の名の通り「光による描像」を行う写真作品群は、1920年代のPhotogram [5]やRayograph [6]を経て、現代のLightning Fields[7]、Heliography[8]などの作品につながっている。これらは、作家それぞれが発見・発明した新たな画像／写真をつくるための方法論と言える。また、デジタル・フォトグラフィに関しても、Andreas Grusky[9]やThomas Ruff [10]らの作品に見られるように、解像度や写真の流通、編集・加工の方法の作成・選択など、その特有の特性を活かした「写真のつくり方」が示されてきた。

このように、カメラやレンズ、フィルムを含む受光部のさまざまな技術史的な発明と並行して、作家それぞれが発見・発明した「写真のつくり方」が生まれ、写真の歴史が更新されてきたと言える。

^{†1} 慶應義塾大学 SFC 研究所
Keio Research Institute at SFC

^{†2} 慶應義塾大学 環境情報学部
Faculty of Environment and information Studies, Keio University

このような背景のもと、本研究では、カメラ自体にコンピュータが内蔵され、ネットワークに接続されることを前提として、撮られた写真をネットワーク上で共有するだけでなく、ユーザが「写真のつくり方」を撮影時に試行錯誤することを可能にし、その「つくり方」自体をネットワークを介して共有することを考える。

具体的には、デジタルカメラ上でのビジュアルプログラミング言語、プログラムの共有・変更機能を有するモバイル上の統合開発環境ourcamを提案する。デジタル・フォトグラフィ特有の撮影手法や、デジタルカメラ上におけるメディアの構築方法が蓄積され、ユーザが任意の場所で思いついたアイデアをその場でプロトタイプ可能とする。また、本環境を用いたユーザスタディによって、ユーザ自身によって「写真のつくり方」が構築可能となったデジタル・フォトグラフィ環境における撮影行為の変容を示す。

2. 関連研究

2.1 カメラ機能や撮影効果の拡張およびカスタマイズ

近年、コンピュータショナルフォトグラフィの分野を中心に、光学設計とコンピュータによる画像処理技術を組み合わせ、新たな視覚効果や表現を生み出す取り組みが盛んに行われている。代表例として、Lytro [11]では、イメージセンサの前にレンズアレイを設置することで、撮影後に任意の被写体深度の写真を生成できる。このアプローチは、従来の撮影後の画像加工でのエフェクトの幅を広げることができる。これに対し、本研究では、撮影前あるいは撮影時にユーザに、ビジュアルプログラミングを通して映像に施すエフェクトの試行錯誤を可能にするというアプローチを採る。エフェクトがかかった状態で撮影することで、そのエフェクトに合わせた構図の決定や被写体の選定を可能にする。

カメラに搭載されるエフェクトなどの機能をユーザが操作できるようにする取り組みとして、The Franken Camera プロジェクト[12]が挙げられる。これは、カメラに搭載されるソフトウェアをオープンにすることで、プログラムによるカメラパラメータの制御や、画像を用いたアプリケーションの開発を可能にしている。本研究は、撮影端末で動作するプログラミング環境を作ることで、撮影しながらエフェクトを調整あるいは作成することができるという点に特徴がある。

2.2 Creative Coding Community

プログラミング環境の参考事例やコードを集積したものとして Creative Coding Community がある ([13]-[19])。これらは動画や音声、マイクロコンピュータなどを主に扱うプログラミング環境である。プログラミングに熟達したユーザには、高速でアイデアのプロトタイプを行うための道具として、初心者には、制作と技術習得、コンピュー

タとソフトウェアの原理の理解を同時に行うための教育ツールとして利用されている。これらにはプログラミングをスケッチのように短時間で試行の結果をみるものとして扱う Sketch, および他者のプログラムを積極的に引き継ぎ、変更する Fork という概念があり、本研究でもそれらの考え方を取り入れている。

2.3 オンサイトプログラミング環境

室内外を問わず、その場でプログラミングすることに関する研究は主にウェアラブルコンピューティングの分野で行われており、オンサイトプログラミングと呼ばれる ([20]-[23])。これらの研究はいずれもコンテキスト依存プログラムの作成を通じたサービスの作成を目的としており、イベント駆動型ルールを通してプログラミングを行う。本研究は、写真撮影のためのツールであるという点と、作成したプログラムの共有を行えるという点に特徴がある。

3. ourcam の提案

3.1 ourcam の概要

筆者らの提案するourcamは、デジタルカメラ上でのビジュアルプログラミングおよびプログラムの共有・変更機能を有するモバイル端末上の統合開発環境である。これによりユーザがある場所で思いついたアイデアをその場でラピッドプロトタイプすることを可能にする。

また、本システムでは、ユーザはプログラムを適用した状態で撮影を行うことができる。これにより思いついた撮影手法をその場で実践し、それを適用した状態で撮影を行うことができる。これによって撮影手法と被写体の相性をその場で確認でき、高速な試行錯誤が可能となる。さらには、ユーザ自身が考えもしなかった被写体を、構築した手法によって発見することにつながると期待する。

さらに、本システムではユーザの撮影した画像を、その撮影手法と共に他のユーザと共有し、変更可能にする。これにより、多様な撮影手法の中から自分自身の表現に近いものを選んだり、カスタマイズすることが可能になる。

3.2 システム構成

本システムの構成を図1に示す。本システムは、端末としてApple社製iPhoneを用い、その上で動作するアプリケーション、サーバ、写真データベースストレージの3点から構成される。

本環境でのプログラムはスケッチと呼び、写真と紐づけられた形で扱われる。ユーザは、スケッチを適用した状態での撮影、および結果を確認しながらのスケッチの組み替えが可能である。また、スケッチおよび写真を共有する際には、iPhone からサーバにそれらのデータをアップロードする。アップロードされた写真のデータは、プログラムに使用されたメソッド、写真のメタデータと共にデータベー

スに記録される。また、ユーザは端末画面のダウンロードボタンを押すことで、スケッチ内で使用されているコードブロックが使用可能になる。さらに、ユーザが認証した各種ソーシャルサービスへの直接のアップロードが可能である。

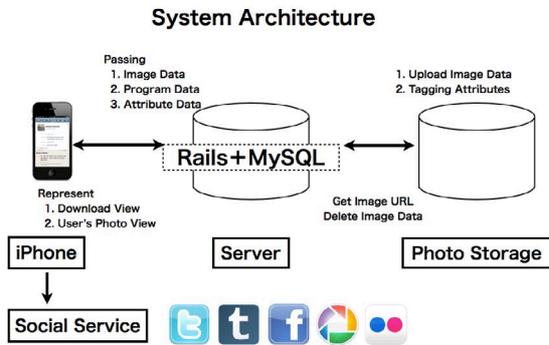


図 1 システム構成

今回は、本システムのプロトタイプとして、iPhone 4 上に iOS4.3, Objective-C++言語を用いた,OpenGL ES ベースの iOS アプリケーションおよび, Mac mini 上の ruby 1.9.3, ruby on rails 3.2, MySQL による Apache サーバによってシステムを構築した。

3.3 ソフトウェアの設計

ourcamでは、撮影の現場で、手軽かつ直感的に映像効果や撮影ルールを設定するために、Codeblock(コードブロック)と呼ばれるメソッドを縦一列に並べることによるビジュアルプログラミングの方法を採る。

本環境のプログラミング環境では、スケッチのダウンロード・アップロード、既存のスケッチの読み込み、そしてプログラミングが可能である。コードブロックは大きくシャッター機能、画像処理機能、共有機能の三種類に分類される。シャッター機能は、時間や物体認識、インターネットからの信号、音、ユーザの行為などの条件が満たされた場合にシャッターを切る機能である。画像処理機能は、イメージフィルタ機能や、複数枚の画像から画像を生成する機能からなる。共有機能は、編集機能、およびアップロード機能からなる。

実装したプログラミング画面を図3(a)で示す。スケッチのダウンロード・アップロード、既存のスケッチの保存、読み込み、そしてプログラミングが可能である。プログラミングの際には、ユーザは、左に表示されたスケッチのコードブロックを、右側の使用するスケッチ部分にドラッグして移動する。コードブロックは上から順に適用される。よって、コードブロックの順序によって異なる結果が得られることがある。

スケッチのデータ構造は、プログラムとパラメータの配列、名前と写真、自身のid、親プログラムのidが構成要素となる(図2)。親プログラムのidをもつことで、プログラムがいかにかForkされてきたのかを追跡できる。

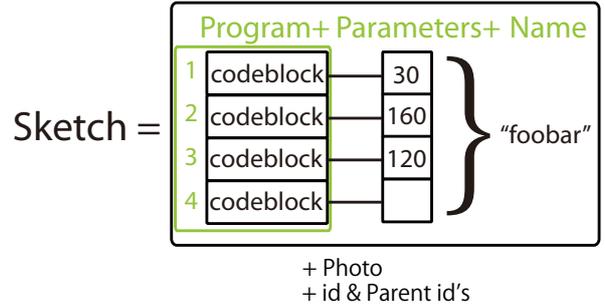
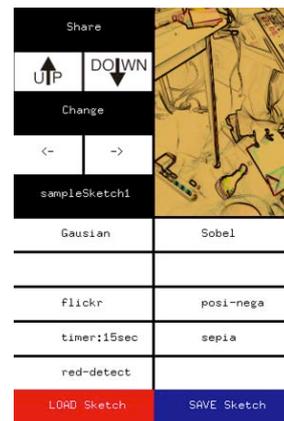
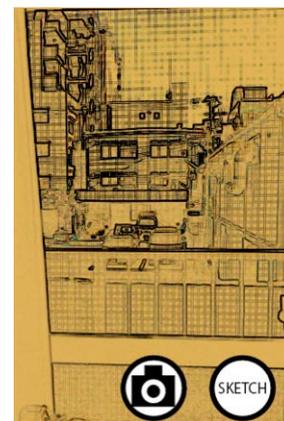


図 2 スケッチの構造



(a) プログラミング時



(b) 撮影時



(c) サーバデータ閲覧時

図 3 ourcam の画面デザイン

ユーザはプログラムの結果を適用した状態の画像を見ながら撮影を行うことができる(図3 (b))。加工や合成手法の制作を前提とした撮影では、編集結果を見るまでが結果が確認できないが、これにより今まで編集プロセスで行われていた、画像に対するフィルタ適用やデータの合成が行われた状態での撮影が可能になる。撮影した際には、撮影された写真と加工されていた場合には加工前の写真とスケッチが、ローカルへの保存と同時にサーバにアップロードされる。

アップロードの際には、スケッチのコードブロックのidとパラメータの値の配列、画像がサーバに渡される。イメージフィルタ機能や画像生成機能が使用されている場合には、コードブロック適用前、適用後の2枚の画像がアップロードされる。使用されていない場合には、1枚の画像がアップロードされる。プログラムにおけるコードブロックはidによって共有される。アップロードされたスケッチはダウンロード可能となる。

ダウンロード画面では、スケッチのリスト画面が表示され(図3 (c))、スケッチを選択すると、そのスケッチを使用して撮影された画像、プログラムの詳細、どのようにスケッチがForkされた結果このプログラムになったかが有向グラフで表示される。ユーザはダウンロードボタンを押すことでこのスケッチをダウンロードし、スケッチ内で使用されているコードブロックが使用可能となる。

4. ユーザスタディ

4.1 ユーザスタディ

本システムを実装し、基礎的なユーザスタディを行った。本ユーザスタディは、ourcamに関するインストラクション、撮影、アンケートおよびインタビューからなる。被験者は19歳から32歳の男女8名(男性4名女性4名)である。被験者は、全員がデジタルカメラやスマートフォンアプリでの撮影を行ったことあり、Photoshopなどの画像編集ソフトを使用したことがある。

(1) インストラクション

実験に先立ち、インストラクションは文章によるourcamの説明、ビデオによる操作方法の教示、口頭での課題の提示からなり、5分程度で行った。ビデオによる操作説明では、アプリケーションの立ち上げ、撮影画面からスケッチ画面への遷移、スケッチの選択方法、スケッチの作成方法、スケッチ画面から撮影画面への遷移、撮影方法を提示した。

以上のインストラクションを行った後で、「30分間で、さまざまなブロックを使って、あなたが面白いと思った写真を撮ってきてください。」という課題を口頭で伝えた。

(2) 撮影

事前の説明の後、被験者は30分間の撮影を行った。撮影に際しては、被験者に観察者が同行し、ハンディカムによるビデオ撮影を行った。観察者は撮影時に、被験者がどの

ようにスケッチを作成したのかを観察するため、画面操作を中心に撮影した。撮影時に観察者は被験者と会話を行ったが、操作に関するアドバイスや説明は行っていない。撮影は慶應義塾大学湘南藤沢キャンパスの構内で行われた。

本ユーザスタディを行った際のアプリケーションの仕様を挙げる。被験者は、サンプルスケッチをもとに、コードブロックを組み替えることで独自の「つくり方」を作成できる。

- ・サンプルスケッチ 5 個
- ・コードブロック 23 個
- ・スケッチの保存、呼び出し機能
- ・アップロード/ダウンロード機能は含めない

(3) インタビュー

撮影を終えた後、それぞれの被験者に対し30分間のインタビューを行った。インタビューは被験者が撮影した写真を見ながらその内容および撮影プロセスについて聴取した。

4.2 結果と考察

以下に被験者の撮影枚数、作成して撮影に利用したスケッチの数、スケッチ作成時の操作回数をまとめる。総操作回数は、・コードブロックを左から右に写す、・コードブロックを右から左に移す、・コードブロックの位置を変える、の3つの操作の合計回数である。観察時に撮影したビデオから目視で計測した。

表1. ユーザスタディで得られたデータ

	撮影枚数 (枚)	利用スケッチ数 (個)	総操作回数 (回)
平均	42.8	16.4	116.7
標準偏差	21.63	3.74	18.67

インタビューで被験者からは、「画面での映り方を考えて被写体を選び、撮影した」という意見が多く得られた。これにより、ourcamを用いた撮影では、被写体を選んで撮影し、それにあった効果を選ぶという従来の被写体の選び方から、作成した効果をもとに周囲を観察し、画面の中で面白く見えるモノを優先し、作成した効果に合った被写体を選ぶという撮影のしかたが多くなされた。これらは、「風景を見て撮ろうというより、かざしながら見ていた」「カメラを縦横に動かしたり、角度を変えたりして、撮影場所を探した」という被験者のコメントにも現れる。

また、他の多かったコメントとして、「同じ被写体についてスケッチを変えて何枚も撮った」「スケッチによって見え方が変わるの面白い」というものが多く、さらに、「特に何ともないものに近づくのが面白かった。ディティールの変化を見るのが面白かった」など、普段は見馴れているた

めに無視してしまう小さな差異に注目する被験者が多くいた。実際に撮影された写真の中でも、被験者の全員が芝や樹木の写真を一回以上撮影し、半数が地面のアスファルトを一回以上撮影するという傾向が見られた。

また、被験者は、スケッチを行う際にそれ以前の撮影で自身が利用したスケッチにて利用したコードブロックをもとに、それらを組み合わせて撮影を行うケースが多くあった。ある被験者は様々なコードブロックをひとつひとつ使っていくことで、それぞれの機能を習得し、それらを組み合わせた。この被験者は図4の上の3枚をそれぞれ撮影した後、撮影タイミングを工夫するとこれらの効果を組み合わせた写真が撮影できるスケッチを作成することに気づき、左下の写真を撮影した。これは撮影を通してコードブロックの意味を理解し、それらを独自に組み合わせた例と言える。

この他、本ユーザスタディを行う中で見られた被験者の特徴的な行動事例を挙げる。まず、ある被験者は、コードブロックと撮影者の動きを組み合わせることで、その場で撮影効果を作りだした。具体的には、図5左の画像で「Time Slice」コードブロックを利用したスケッチを作り撮影を行った際に、カメラを動かすことで「被写体がのびた」かのような効果を持った写真が撮影できることに気づき、何度か被写体を変えて練習を行い、右の画像を撮影した(図5右)。これは、効果が撮影前に見えることで、コードブロックの特徴を理解した上で最適な被写体を探すことができ、さらに自分の動きを加えることで、新たな撮影技法を獲得した例と言える。

別の被験者は「Hadairo」コードブロックと「Laplacian」コードブロックを使ったスケッチを適用して撮影する際、撮影したい範囲全体を撮影すると、範囲の一部を撮影した場合、画像の質感が変わってしまうと感じた。そこでこの被験者は自分自身が今いる位置から後ろに移動するのではなく、腕を水平に動かして2枚の写真を撮影した(図6)。これも撮影技法をユーザが編み出した例と言える。



図4 コードブロックを組み合わせた例

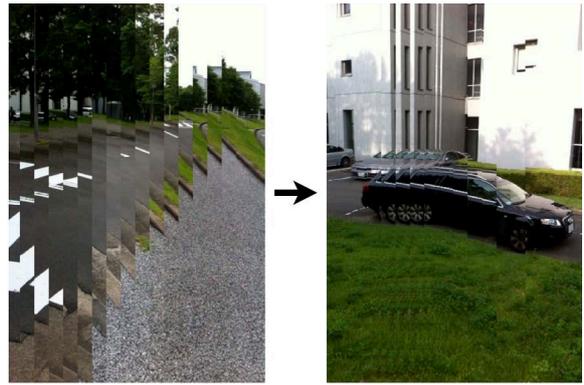


図5 コードブロックと身体の動きを組み合わせさせた例

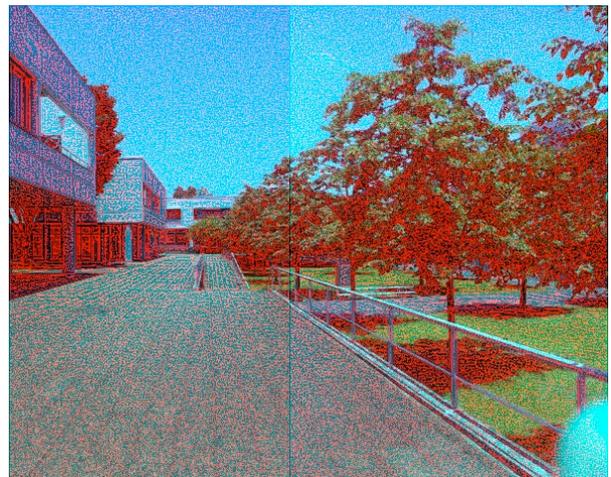


図6 二枚の写真を組み合わせて写真を作成した例



図7 傘の水滴を用いてフィルタを作成した例
(左：元画像、右：スケッチ適用後画像)

もう一つの特徴的な行動として、ある被験者は、撮影当初から「Sobel」コードブロックをもとに様々なスケッチをつくり、写真を撮影した。その中で、偶然持っていたビニール傘が画角に入り、「ビニール傘についた水滴」が、このスケッチの中では風景にドット柄を加える面白い効果として利用できると考えた。最終的に、このユーザは、傘をカメラの前にかざし位置を調整して撮影を行った(図7)。本

例は、変化が起こった状態を見ながらで撮影を行うことで、ourcam内のコードブロックのみならず、自分の身の回りにある実世界オブジェクトをコードブロックのように利用する創意工夫が生まれた例と言える。

5. おわりに

本稿では、デジタル・フォトグラフィのためのオンサイトプログラミング環境 ourcam を提案した。またユーザスタディにおいて、これらの環境を用いた撮影行為において、当環境内の機能を学習して利用するのみならず、実世界オブジェクトを方法の一部として利用する、撮影者自身が発見したスキルと組み合わせる等の試行錯誤を行い、新たな手法を獲得してく様子を示した。

今後は、本環境におけるデータの共有をより長期的に多人数で行い、結果の分析を行いたい。また、本環境ではプログラミングの成果物(写真)と撮影行為、プログラムが分ち難く結びついており、本環境を利用したデータを分析することで、いかにコンテンツと結びついたプログラムが分岐していくのか、その際の要因は何かなど、オープンソースデザインにおけるフォークのされ方や、ウェブを通した創造性の発現の過程の一端を明らかにしていきたい。

謝辞 本研究の一部は、IPA 未踏人材発掘・育成事業の支援を受けて実施した。

参考文献

- 1) Instagram, <http://instagr.am>
- 2) iPhone, <http://apple.com/iphone/>
- 3) Android, <http://www.android.com>
- 4) クエンティン・バジャック著、伊藤俊治監修: 写真の歴史, 創元社(2003).
- 5) 井口 壽乃: モホイ=ナジ 視覚の実験, 国書刊行会(2011).
- 6) Guido Comis, Marco Gramciolli: Man Ray, Skira(2011)
- 7) 杉本博司他: 杉本博司-光の自然, Izu Photo Museum(2009).
- 8) 山崎博: HELIOGRAPHY, 青弓社(1983).
- 9) Andreas Gursky: ArchiTecture, Hatje Cantz Pub(2008).
- 10) Thomas Ruff: jpegs, Aperture(2009).
- 11) Lytro, <https://www.lytro.com/>
- 12) Andrew, A. et al : The Frankencamera: An Experimental Platform for Computational Photography, ACM SIGGRAPH 2010 papers, No. 29, 12pages.(2010)
- 13) MAX/MSP, <http://cycling74.com/products/max/>
- 14) Pd, <http://puredata.info>
- 15) vvvv, <http://vvvv.org>
- 16) Quartz Composer, developer.apple.com/technologies/mac/graphics-and-animation.html
- 17) Processing, <http://processing.org>
- 18) openFrameworks, <http://openframeworks.cc>
- 19) wonderfl, <http://wonderfl.net/>
- 20) 宮前他: バイクレース支援のためのイベント駆動型ウェアラブルシステム, 情報処理学会研究報告, Vol. 2004, No. 44, pp. 53-58 (2004).
- 21) 寺田他: その場プログラミング環境の実現に向けて, 情報処理学会研究報告, Vol. 2007, No. 46, pp. 1-8 (2007)
- 22) 西本他: 携帯電話上でコンテキスト依存プログラムを記述す

るためのビジュアルプログラミング環境, 情報処理学会第70回全国大会講演論文集, Vol. 4, pp. 267-268 (2008)

23) S. Akiyama et al : An On-Site Programming Environment for Wearable Computing , AH '12 Proceedings of the 3rd Augmented Human International Conference, No.1, pp.1-4 (2012)