

筑波大学大学院博士課程

システム情報工学研究科修士論文

公共・他人のカメラを用いて撮影された写真から
ライフログを構築するシステム

閑野 伊織

修士（工学）

（コンピュータサイエンス専攻）

指導教員 田中 二郎

2015年3月

概要

ライフログは、様々なデータを蓄積し、分析することで振り返りや気付きに役立てられている。ライフログとして写真を記録することは楽しかった思い出を振り返る際に重要な要素となる。特に近年ではスマートフォンの普及によって、スマートフォンに搭載されたカメラを用いて今まで以上に多くの人がいつでもどこでも写真を撮影することができるようになった。

そのため、撮影自体は容易になったが、撮影された自身の写っている写真を貰うことや自身が撮影した友人の写っている写真を配ることが困難であり、如何にして自身の写っている写真を集約するかという点が今後の課題となっている。また、集合写真を撮影する際にもシャッターを切る撮影者、もしくは機能が必要となる。シャッターを切る機能として一般的にセルフタイマーが用いられるが、これには確認するためにカメラと撮影地を往復する必要があるため、集合写真の撮影には改良の余地がある。

本研究では、どのカメラを用いるかによらず撮影時に各ユーザを識別することで撮影された写真に写っているユーザの情報をメタデータとして付加する。具体的には撮影前に各ユーザが身につけている腕時計型端末にカラーマーカを表示し、それをカメラに写して識別することでユーザを識別する。識別が完了したユーザには腕時計型端末から振動フィードバックが与えられるため、ユーザは識別が完了したことを知ることができる。一度識別が完了するとカメラが自動でトラッキングを行うため、ユーザは腕時計型端末をカメラにかざす必要がなくなり、好きなポーズにて撮影に臨むことができる。撮影後にユーザ情報の付加された写真を各ユーザのパーソナルクラウドに送信・記録することでライフログを構築する。本研究では以上のシステムを開発した。さらに、公共の場にネットワーク環境を備えたカメラが設置された未来環境を想定し、誰もが自身の端末から操作可能な公共カメラとの連携手法を提案する。腕時計型端末を用いて公共カメラを遠隔操作することにより、集合写真の撮影において撮影者が写真に写れないといった問題を解決することが可能になる。

目次

第1章 序論	1
1.1 研究の背景	1
1.2 本研究が提案するアプローチ	1
1.3 本論文の構成	2
第2章 想定する未来環境	3
2.1 公共カメラの普及した未来	3
2.2 腕時計型ウェアラブル端末の普及した未来	3
2.3 パーソナルクラウドの普及した未来	4
第3章 要件定義	5
3.1 ユーザ識別機能	5
3.1.1 ユーザ識別手法の検討	5
3.1.2 ユーザフィードバック手法の検討	7
3.2 識別したユーザの写っている写真を各パーソナルクラウドに送信する機能	7
第4章 ユーザと写真を紐付けてライフログを蓄積するシステム	8
4.1 概要	8
4.2 使用方法	11
4.3 利用シナリオ	11
4.3.1 通常の利用シナリオ	11
4.3.2 撮影が難しい状況での利用シナリオ	12
4.4 検討事項	12
4.4.1 プライバシー	12
4.4.2 ユーザ識別精度	13
4.4.3 蓄積した写真のブラウジング手法	14
第5章 実装	15
5.1 開発環境	15
5.2 システム構成	15
5.3 カメラ-ユーザ識別部	16
5.3.1 顔の検出	17

	顔が検出できない場面	17
5.3.2	色の抽出	17
	HSV から RGB を算出	20
5.3.3	通信部	22
5.4	クライアント - 腕時計型端末	23
5.4.1	カードの表示	23
5.4.2	通信部	23
5.4.3	振動フィードバック	24
5.5	サーバ	25
5.5.1	位置情報による絞り込み	25
5.5.2	カラーマーカの生成	26
5.5.3	写真のアップロード・閲覧	26
第 6 章	関連研究・関連サービス	27
6.1	写真にメタデータを付加して記録	27
6.2	動的に作成されたマーカの認識	28
6.3	カメラの遠隔操作	28
6.4	撮影した写真の共有	29
6.5	写真に写っている人の顔の識別	29
6.6	ビデオカメラを用いた個人識別型位置検出	29
6.7	カメラが自動で撮影するシステム	30
6.8	写真のブラウジング・探索手法	30
第 7 章	まとめと今後の課題	32
7.1	まとめ	32
7.2	今後の課題	32
	謝辞	33
	参考文献	34

目次

4.1	システム利用の様子	8
4.2	システム概要	9
4.3	公共カメラを用いた場合の初期ステップ	10
4.4	反射防止フィルムを用いた場合と用いない場合の比較	13
5.1	開発に用いた端末	16
5.2	システム構成図	17
5.3	他人のカメラを用いた場合のシステム実行の流れ	18
5.4	公共のカメラを用いた場合のシステム実行の流れ	19
5.5	ユーザ識別のためのマッチング	20
5.6	HSV への変換前	21
5.7	HSV への変換後	22
5.8	腕時計型端末の画面遷移	23
5.9	カラーマーカの一覧	24
5.10	位置情報による絞り込みの様子	26

表目次

3.1 システム要件と識別手法の比較	6
5.1 開発環境	15

第1章 序論

本研究では、撮影される写真に写っているユーザを一意に紐つけることによってライフログを構築する。本章では、現在のカメラ撮影における問題点とその問題を解決するアプローチを述べ、最後に本論文の構成について述べる。

1.1 研究の背景

ライフログは、様々なデータを蓄積し、分析することで振り返りや気付きに役立てられている。ライフログとして写真を記録することは楽しかった思い出を振り返る際に重要な要素となる。特に近年ではデジタルカメラの普及により、撮影したその場で写真を確認することが可能となり、フィルムによる枚数制限もなくなったことから、写真を撮影する機会や撮影する枚数が増えている。さらに、スマートフォンの普及によって、スマートフォンに搭載されたカメラを用いて今まで以上に多くの人がいつでもどこでも撮影することができるようになった。

そのため、撮影自体は容易になったが、撮影された自身の写っている写真を貰うことや自身が撮影した友人の写っている写真を配ることが困難であり、如何にして自身の写っている写真を集約するかという点が今後の課題となっている。また、集合写真を撮影する際にもシャッターを切る撮影者、もしくは機能が必要となる。シャッターを切る機能として一般的にセルフタイマーが用いられるが、これには確認するためにカメラと撮影地を往復する必要があるため、集合写真の撮影には改良の余地がある。

1.2 本研究が提案するアプローチ

本研究では、どのカメラを用いるかによらず撮影時に各ユーザを識別することで撮影された写真に写っているユーザの情報をメタデータとして付加する。撮影後に、ユーザのデータが付加された写真を各ユーザの所有するパーソナルクラウドに記録することによりライフログを構築するシステムを開発する。さらに、防犯カメラや証明写真撮影機のようなカメラが公共の場に設置され、ネットワーク環境を備えた未来環境（以後、公共カメラと呼ぶ）を想定し、誰もが自身の端末から操作可能な公共カメラとの連携手法を提案する。

1.3 本論文の構成

本論文は7章で構成されている。第2章で本研究が想定する未来環境について述べた後、第3章で開発するシステムの要件定義を述べ、第4章では開発したシステムについて概要と使用方法、どのような場面で役立つのかについて利用シナリオを述べる。第5章では実装について延べる。第6章では関連研究について述べ、第7章で本研究のまとめと今後の課題について述べる。

第2章 想定する未来環境

本章では本研究で想定する未来環境について述べる。以下で述べる公共カメラと腕時計型ウェアラブル端末、パーソナルクラウドが普及し、すべての人が使用する未来環境を想定している。

2.1 公共カメラの普及した未来

公共カメラとは、1章で述べたように、公共の場に設置された、誰でも操作可能なカメラであり、顔検出とオートフォーカスは標準搭載されており、ネットワーク環境を備えているものを想定する。これはゲームセンターに設置されているプリント倶楽部（以下プリクラ）や繁華街に設置されている防犯カメラ、証明写真機のようなイメージで至るところに存在し、誰もが自由に使用することができるカメラである。公共カメラというものは私が提案しているにすぎないが、上述のようにプリクラや証明写真では無人の機械によって紙媒体及びデジタル媒体で写真撮影が可能となっていることから実現可能性は十分にあり、本研究によって有用性が示されることによって実現可能性はさらに高まるのではないかと私は考える。

2.2 腕時計型ウェアラブル端末の普及した未来

ウェアラブル端末とは、身体につけスマートフォン等の機器と連携通信することによってユーザの利便性を向上させるデバイスである。現在製品化が進んでいるウェアラブル端末ではスマートウォッチと呼ばれる腕時計型端末、モバイルヘルスケア端末と呼ばれる装着者の健康に関わる情報を収集するブレスレット型の機器、スマートグラスと呼ばれる眼鏡型端末の3つが有名である。2014年の今年はウェアラブル元年と呼ばれる年であり、野村総合研究所の市場調査[1]によると、国内のウェアラブル端末の販売台数は2013年が23万台だったのに対し、2018年には475万台に達すると予測されている。近年では、Google社がシースルー型のGoogle Glassや腕時計型のGoogle Watchを発売し注目されており、眼鏡の販売店であるJINS!社もスマートグラスの開発を行っている¹。腕時計型端末も各社が開発を行っており、今後益々普及すると考えられる。

¹Jins meme, <https://www.jins-jp.com/jinsmeme/product/>

2.3 パーソナルクラウドの普及した未来

パーソナルクラウドとは、外部のサーバ上に存在する個人のストレージのことである。近年 Google 社の Google Drive² や Apple 社の iCloud³, Dropbox 社の同名製品⁴ 等, 様々な企業がパーソナルクラウドを提供している。パーソナルクラウドのメリットとして、自端末には日頃から使用するデータを保存しておき、頻繁に使用するわけではないデータをパーソナルクラウドに置くことで自端末の容量を抑えることができ、必要な時に複数の端末から参照することができる点があげられる。IDC Japan が提供しているクラウドストレージ従量課金サービスの市場調査 [2] によると、2013 年～2018 年の年間平均成長率は 26.7%であると予測されている。本研究ではパーソナルクラウドは今後益々普及すると考え、全ての人がパーソナルクラウドを所有している未来を環境を想定する。

²Google Drive, <http://www.google.com/intl/ja.jp/drive/>

³iCloud, <https://www.apple.com/jp/icloud/>

⁴Dropbox, <https://www.dropbox.com/>

第3章 要件定義

本章では、2章で述べた未来環境を想定した上で、1章で述べた問題とそれを解決するために必要なシステムの要件について定義する。本研究で要求される機能および環境は、写真に写っている被撮影者（以後、ユーザと呼ぶ）を識別する機能、写真を撮影する機能、写真を撮影した端末から識別したユーザのパーソナルクラウドに自動で送信する機能である。

3.1 ユーザ識別機能

3.1.1 ユーザ識別手法の検討

入力された画像から個人を一意に識別する方法は以下の4通りである。

1. 顔の特徴量から識別
2. 固有マーカを用いて識別
3. 赤外線発信機と受信機を用いて識別
4. ウェアラブル端末を用いて識別

このうち、1はGoogle Picasa等のサービスで識別可能なことが知られているが、認識するためにはサングラス等で顔が隠れておらず、髪の長さや髪型、髭等の変化がないこと、表情の変化がないこと等、制約¹が細かく、実用化には課題が多い。2の固有マーカについてはマーカが常に写真に写る点と世界の人口それぞれに固有マーカを割り振ることは難しいが、識別すること自体は1よりもはるかに制約が緩いといえる。3の赤外線発信機と受信機を用いる方法では、被撮影者がカメラから写る位置に赤外線発信機を装着し、赤外線受信機を搭載したカメラを用いて発信機の発する点滅信号を受信することで識別を行う。ただし、赤外線発信機を全てのユーザが装着するにはセットアップの負担が大きく、全世界で用いるには課題が多い上に2同様にカメラに機器が写り込んでしまう。識別が可能なことは不可欠であるが、見栄えの観点から写真に常にマーカや機器が写り込むことは極力避けるべきだと考える。4のウェアラブル端末を用いた識別では3と同様に機器のセットアップコストはかかるものの、ウェアラブル端末は今後普及することにより問題ではなくなると考える。写り込みに関しても、常時身に付ける機器であるため不自然な写り込みではなく問題ないと考えられる。よって、ウェアラブル端末を用いることはユーザ識別に適しているといえる。

¹Wikipedia - 顔認識システム, <http://ja.wikipedia.org/wiki/顔認識システム#.E8.AA.8D.E8.AD.98.E7.8E.87>

上記を整理すると、本研究で開発するシステムの要件は、(a) ユーザ識別のための制約が緩く、かつ (b) 精度が高いこと、また、(c) カメラに識別に用いる機器が不自然に写り込まないことと定義することができる。この要件と上述の方法の比較表を表 3.1 にまとめる。

表 3.1: システム要件と識別手法の比較

	(a) ユーザ識別の制約	(b) ユーザ識別の精度	(c) 写り込み	コスト
1. 顔の特徴量を用いた場合	×	△	無	低い
2. 固有マーカを用いた場合	○	○	有	低い
3. 赤外線を用いた場合	○	△	有	高い
4. ウェアラブル端末を用いた場合	○	△	自然な写り込み	高い

ここで、識別精度が優れている 2 について掘り下げて考える。固有マーカの数的制約に着目すると、写真に写っている人物を一意に識別することが求められているに過ぎないため、全世界の人が一度に使用する場面は想定する必要がないことがわかる。つまり、時間や場所といった制約を加えることによって固有マーカの数的制約は取り払うことができる。つまり、マーカをその時その場所によって動的に作成することで数的制約を解消することが可能なので条件付きではあるが候補に残すことにする。

次に、写真へのマーカの不自然な写り込みについて考える。写り込みを解決するためには以下の 2 つの方法が考えられる。

1. 画像処理を行いマーカを除去する
2. 識別後にトラッキング（追跡）することで撮影時点で写り込まないようにする

画像処理を用いる場合、マーカを完全に除去するためには端末に相応のスペックが必要になる。なぜなら、除去部分を置き換えるための画像をデータベースとして大量に保持しておき、除去部分周辺の画像情報から推定する必要があるためである。そのため、本研究の誰のカメラかによらずという点に反する。トラッキングする場合でも大きな動きを追跡する場合には相応のスペックが必要になるが、集合写真のような動きが少ない場面に絞ることでトラッキングの範囲を大幅に限定することが可能である。そうした場合、トラッキングすることは現在のスマートフォンのスペックで十分可能である。以上から写り込みを解消するためにトラッキングを用いて、識別が完了したことを何らかの形でフィードバックしユーザに気付かせることでマーカが写り込まない状態で撮影が可能となる。

まとめると、特定の時間に特定の場所にいる状況に限定するという点からディスプレイを用いて動的にマーカを生成することが良く、識別完了後はマーカを隠してもトラッキングすることでユーザを見失わないシステムが必要である。また、写真はいつ撮影するかわからないため識別するための装置は常に身につけていることが望ましい。そこで本研究では 2 と 4 を組み合わせて用いることを考えた。ウェアラブル端末に着目し、上記の条件からディスプレイを備えている腕時計型端末を使用することとした。そして腕時計型端末のような表示領域が小さい場合でも認識されやすいカラーマーカを動的に生成することとした。カラーマー

カの精度が得られない場合、どのようなマーカであれば表示領域が小さくても認識されやすいのかといった点を検証し、変更する。

3.1.2 ユーザフィードバック手法の検討

ユーザにフィードバックを行うためには、五感である視覚、聴覚、触覚、味覚、嗅覚を刺激することが考えられる。既存の製品で多く用いられており現実味があるのは視覚、聴覚、触覚である。本研究で用いる腕時計型端末にはディスプレイとバイブレーション機能が付いているため、視覚と触覚のどちらかが好ましい。ここで写真撮影の際に用いることを考えた場合、ディスプレイを見るよりもバイブレーション機能による振動を用いることの方がユーザにとって使いやすくと私は考える。そのため、本研究ではユーザへのフィードバックに振動を用いて触覚フィードバックを用いることとする。

3.2 識別したユーザの写っている写真を各パーソナルクラウドに送信する機能

写真に写っているユーザのパーソナルクラウドに蓄積するために考えられる方法は2通りある。1つ目は撮影端末からサーバに写真および識別したユーザの情報を送信し、サーバから各ユーザのパーソナルクラウドに送信する方法である。2つ目は、撮影端末から直接各ユーザのパーソナルクラウドに送信する方法である。サーバから各パーソナルクラウドに送信する場合、一時的にサーバ上に写真が保存されることになる。これはサーバに不具合が発生した場合やクラッキングされた場合にユーザのプライバシーが脅かされることとなり好ましくない。撮影端末からそれぞれのユーザのパーソナルクラウドに写真を送信する場合、写っているユーザの人数によって通信量は線形的に増加するため、撮影端末の負荷が大きくなる。どちらの場合も一長一短あるものの、通信量は定額制が流行し年々価格が下がってきていることやプライバシーを考慮すると後者が望ましい。よって本研究では、撮影端末から直接各ユーザのパーソナルクラウドに送信することとする。

第4章 ユーザと写真を紐付けてライフログを蓄積するシステム

本章では開発したシステムの概要及び使用方法，使用する場面について述べる。

4.1 概要

本システムは写真撮影時に，被撮影者（以後，ユーザと呼ぶ）がカメラに向かって腕時計型端末をかざすことによって撮影された写真とユーザを結びつけて記録することができるシステムである。具体的には，ユーザが腕時計型端末に表示されているマーカをカメラにかざすことで，予めカメラ端末に保持されているマーカとユーザ ID の対応表と照合する。カメラ端末は照合してユーザを一意に定めた後，撮影された写真をユーザのパーソナルクラウドへ送信・記録する。ユーザは 1～10 人を想定している。腕時計型端末に表示されたマーカとユーザ ID の照合が完了した際には該当ユーザの腕時計型端末を振動させることによってフィードバックを行う。さらに，照合完了後はユーザの顔を自動追跡するため，腕時計型端末をかざしたままである必要はなくなる。そのため，ユーザは好きなポーズの写真を撮影することができる。図 4.1 では腕時計型端末を用いていないが腕時計型端末を用いていると仮定し，カラーマーカを表示した場合の利用の様子を示している。



図 4.1: システム利用の様子

システムの概要図を図4.2に示す。図4.2中の番号では以下の処理を行っている。本システムでは前章で述べた通り、ユーザ識別のためにカラーマーカを用いる。

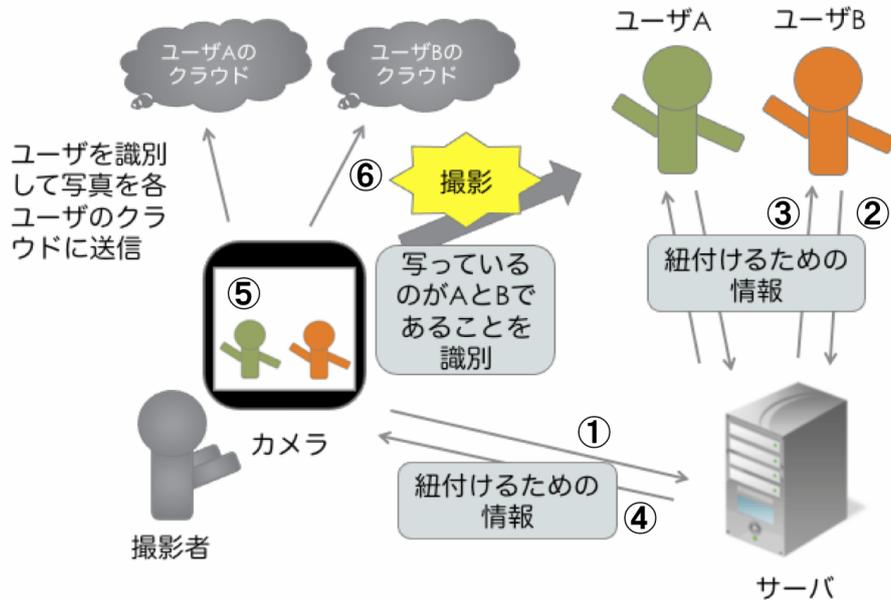


図 4.2: システム概要

1. カメラを起動すると、入力画像を解析し、背景に使われている色を検出する。そして、その色と位置情報をサーバに送信する。
2. ユーザは腕時計型端末を操作し、位置情報をサーバに送信する。
3. サーバはカメラから指定された色と周辺で同時刻に使用されている色を除いたユーザ識別用の色を各ユーザ端末に送信する。
4. サーバはカメラに各ユーザの ID とそれに紐付けた色の情報を送信する。
5. カメラはサーバから受け取った情報を基に写っている画像から検出した顔と指定色を探しユーザをマッチングし、完了したらサーバを経由してユーザに振動フィードバックを与える
6. カメラは撮影後に識別したユーザの ID からそのユーザのクラウドに撮影した写真を送信する

公共カメラを用いる場合は最初に公共カメラの起動を行う必要があることとシャッターを切るタイミングを指定する必要があるため4.2の前にステップが4つ増える(図4.3)。図4.3中の番号では以下の動作・処理と対応している。

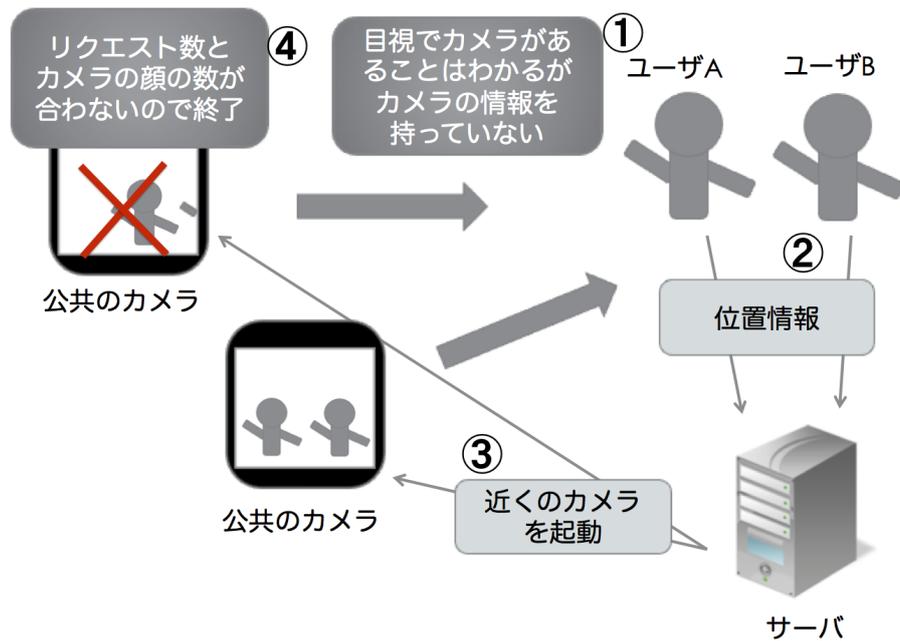


図 4.3: 公共カメラを用いた場合の初期ステップ

1. 公共カメラを用いたいユーザは目視で公共カメラがあることを確認し、カメラの正面に立つ。
2. 各ユーザは腕時計型端末を操作し、位置情報をサーバに送信する。
3. サーバは受信した位置情報の近くにある公共カメラを起動し、各公共カメラにユーザの人数を送信する。
4. 公共カメラは入力画像からユーザの人数を取得し、サーバから受信したユーザ数と比較する。ユーザ数の一致したカメラが撮影に用いられるカメラとなり、ユーザ数の合わないカメラは終了する。

4.2 使用方法

ユーザの目線からシステムの使用方法を述べる。ユーザが行うアクション数は公共カメラを使用する場合と他人のカメラを用いる場合で異なる。公共のカメラを使用する場合には、1. カメラの起動、2. カメラとのマッチング、3. シャッターを切る、の3つのアクションを行う。それぞれがユーザの以下の操作に対応する。

1. 腕時計型端末からアプリを起動する。
2. 腕時計型端末を顔の前にかざす。
3. 腕時計型端末のボタンをタップする。

他人のカメラを用いる場合は、1. カメラの起動と3. シャッターを切るアクションが不要になるので、1つのアクションで使用することが可能である。

ユーザがカメラから認識されたことは腕時計型端末のバイブレーションによってフィードバックを行う。一度認識されると大きく動かない限り自動で追跡されるため、腕時計型端末をかざす必要はなくなりユーザは自由なポーズで撮影を行うことができる。

4.3 利用シナリオ

本システムの利用シナリオを公共カメラを用いる場合と他人のカメラを用いる場合についてそれぞれのシーンを述べる。

4.3.1 通常の利用シナリオ

グループの集合写真を撮影したいが近くに人がいない場合、従来の手法だとセルフタイマーを用いる、もしくはグループの中の誰か1人が撮影するというのが一般的である。本システムを用いることで、新しい集合写真の撮影が可能となる。

公共カメラを用いる場合:

1. 集合写真を撮りたいグループがいた。
2. 周りを見回すと公共カメラがあったので、腕時計型端末からそれを操作して撮影した。
3. 撮影された写真は自動で被撮影者クラウド上に保存されるので、各自が閲覧し満足した。

他人のカメラを用いる場合:

1. 集合写真を撮りたいグループがいた。
2. 周りを見回すと通行人がいたので離れた場所から声をかけて撮影を依頼した。

3. 通行人は自身のスマートフォンを取り出し、グループの集合写真を撮影した後そのまま去って行った。
4. 撮影された写真は自動で被撮影者のクラウド上に保存されるので、各自が閲覧し満足した。

4.3.2 撮影が難しい状況での利用シナリオ

見知らぬ他人に集合写真を撮影してもらう場合、従来の方法では被撮影者が自身の端末を撮影者に渡して、撮影してもらい、それを受け取って確認するというプロセスが存在する。本研究ではそのプロセスが簡潔になると同時に、以下のように被撮影者がカメラを持っている必要がなくなる。

1. 川で遊んでいるグループが写真を撮ろうとしていた。
2. 周りを見回しても公共カメラがなかったため、付近にいた A さんに川の中から声をかけた。
3. A さんは自身のカメラ付端末を取り出し、グループの写真を撮影した。
4. A さんに礼を言うとグループは再び遊び始めたが、その時の写真はしっかりクラウドに記録されていた。

上記の状況では腕時計型端末が防水である場合を想定している。本研究で用いた端末は防水性能である IPX¹ が 7 の防水であるため身につけた状態においても川遊び等が可能である。

4.4 検討事項

本システムを開発する上で留意したことや検討していることについて述べる。

4.4.1 プライバシー

本システムを用いて撮影する場合、撮影端末に写真を残すことで、被撮影者のプライバシーが侵害されてしまう恐れがある。この問題を解決するために、公共カメラでは撮影した写真を各ユーザのクラウドへの送信後、削除する。他人のカメラを用いた場合には、各写真について被撮影者が残す/消すの操作が行えるようにすることを想定している。提案システムでは、写真と写真に写っているユーザが紐付いているので、撮影後であっても残す/消すの操作が可能なのではないかと考えられる。

¹水の浸入に対する試験（防水試験）、<http://www.rime.jp/secint/elecsec/KD3.htm>

4.4.2 ユーザ識別精度

現在、ユーザ識別部の評価が行えていないため、ユーザ識別の精度がどの程度得られるのかという点で不明である。特に、屋外での利用に関して反射により色が認識できないことが考えられる。腕時計型端末は日常的に用いるためもちろん屋外での仕様を想定しており、反射についても日々改良されている²がまだまだ課題は多い。そこで本研究ではこの問題に対して、反射防止フィルムと環境光センサを用いて解決できないかと検討している。反射防止フィルムを用いた場合と用いない場合の比較画像を図4.4に示す³。具体的には、反射防止フィルムを貼ることである程度反射を抑えることが可能であり、端末に備わっている環境光センサ⁴で周囲の明るさを認識し、明るさに対しての識別値を機械学習を用いて調整することで精度が向上するのではないかと考える。精度が得られなかった場合のさらなる検討として、腕時計型端末に付随する加速度センサを用いてリアルタイムに加速度情報を送信、カメラで動きを検出という方法を検討している。ユーザ識別部の実装が済み次第、精度を検証し、比較を行う予定である。



図 4.4: 反射防止フィルムを用いた場合と用いない場合の比較

²レビュー：SmartWatch SW2 -前モデル MN2 との比較-, <http://gadgets.blog.jp/archives/860304.html>

³Mecan Imaging 製品案内 - 弱粘着タイプ反射防止フィルム 携帯・スマホ・タブレット PC 用 AR フィルム, <http://www.mecan.co.jp/Optical-Film/Anti-Reflection/Mobile.html> から引用

⁴SmartWatch3 の仕様, <http://www.sonymobile.co.jp/product/smartwear/swr50/>

4.4.3 蓄積した写真のブラウジング手法

本システムでは、写真に写っている人物や撮影者が取得可能（他人のカメラを用いた場合）なので、従来手法と違ったアプローチからブラウジングが可能になるのではないかと考え、ブラウジングの手法を検討している。例えば、ある特定の人物 A さんが撮影した写真だけを閲覧したいといったニーズに対応できる。また、B さんと C さんが一緒に写っている写真の抽出といったことや自身と D さんが一緒に写っていない写真だけ閲覧したいといったことにも対応可能である。

第5章 実装

5.1 開発環境

システムの開発環境を表 5.1 に示す。本研究では腕時計型端末¹ のアプリケーションを Android 用アプリケーションとして開発する。カメラ端末のアプリケーションを iOS 用アプリケーションとして開発する。開発言語には Java と Objective-C, C++ を用いる。サーバサイドアプリケーションは開発言語に Ruby を用いて、フレームワークとして Ruby on Rails² を用いる。本研究ではクラウドも同様のサーバ環境を用いて開発する。実装に用いた端末を図 5.1 に示す。左上が SmartWatch2, 左下が Xperia, 右の端末が iPad mini である。

表 5.1: 開発環境

機種およびソフトウェア	仕様
SmartWatch2	1.6 インチ液晶ディスプレイ, 220 × 176 pixel
Xperia Android Studio	Android 4.3 ver 0.8.1
iPad mini Xcode	iOS 7.01 ver 5.1
Ubuntu Apache	Ubuntu 10.04.4 Apache/2.2.14

5.2 システム構成

本システムは腕時計型端末, サーバ, カメラの3つから構成されている。システム構成を図 5.2 に示す。システム構成図では上から下に向かって時系列に並んでおり, 青色がデータの流れ, 赤色が動作に対応している。腕時計型端末では位置情報とユーザ ID の送信, マーカの表示, ユーザフィードバックを行っている。サーバは主に腕時計型端末とカメラの中継とマーカの生成を行っている。また, 撮影された写真をクラウドに保存している。カメラではサーバから渡されたマーカ情報とユーザ ID を用いて検出した顔とマッチングを行っている。そしてユーザと検出した顔がマッチングしたことをサーバに伝える。サーバは腕時計型端末にそれを伝える。腕時計型端末はマッチングしたことを振動でユーザにフィードバックする。カメラは撮影した写真を各ユーザのクラウドに送信する。上記をまとめた流れ図を他人のカメラ

¹SmartWatch2, <http://www.sonymobile.co.jp/product/accessories/smartwatch2/>

²Ruby on Rails, <http://rubyonrails.org/>



図 5.1: 開発に用いた端末

ラを用いた場合と公共のカメラを用いた場合に分けて図 5.3 と図 5.4 にそれぞれ示す。両者の違いは公共カメラを用いた場合の流れ図を緑で示した。

5.3 カメラ - ユーザ識別部

ユーザを識別するために、腕時計型端末のディスプレイに色を表示させ、OpenCV³を用いて色を検出する。そしてカメラの顔検出機能を用いて顔を検出、その顔の下にある色と顔をマッチングすることでユーザを識別する。その様子を図 5.5 に示す。図 5.5 では腕時計型端末ではなくスマートフォンを用いている。カメラ付端末搭載の顔検出 API を用いて検出した顔と、OpenCV での色相検出によってリアルタイムで画像を解析する。腕時計型端末のディスプレイに表示するカラーマーカはサーバから取得したものをを用いることによってマッチングを可能としている。カラーマーカ部分については後述する。

³OpenCV, <http://opencv.jp/>

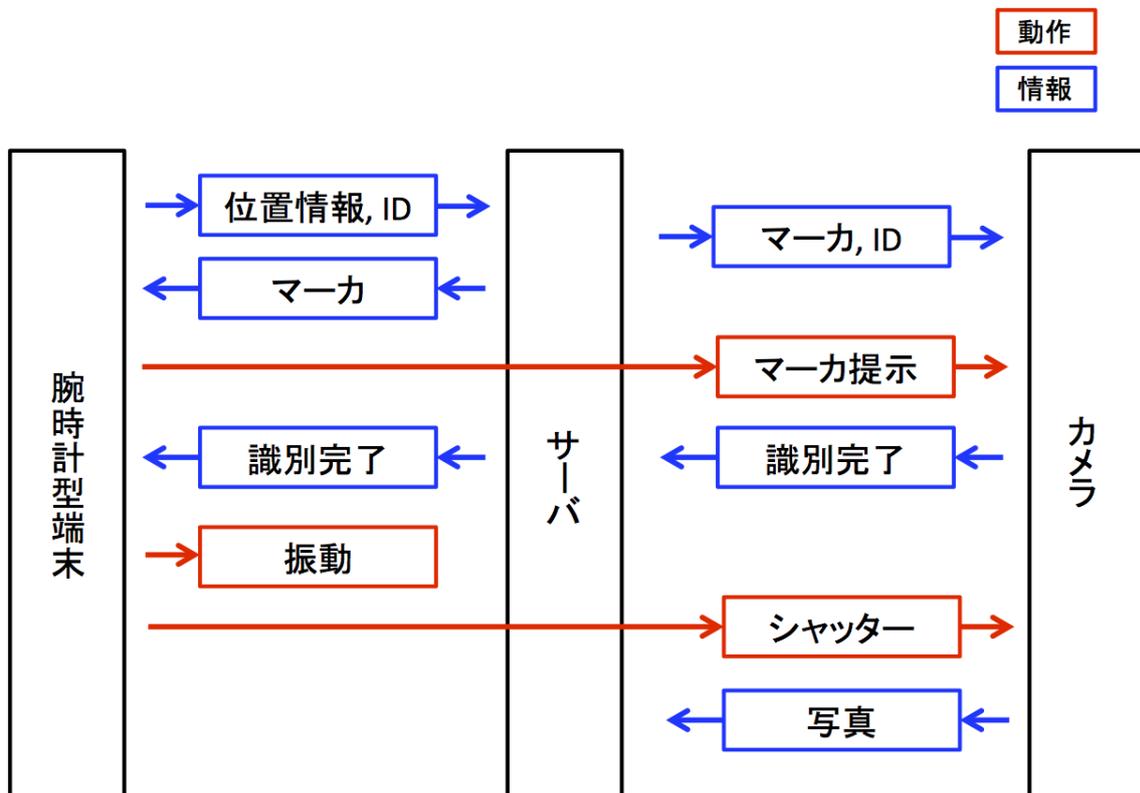


図 5.2: システム構成図

5.3.1 顔の検出

openFrameworks の顔検出メソッド⁴を用いて顔の検出を行っている。メソッドではテンプレートを設定することによって顔以外の形を検出することが可能である。本システムでは標準で用意されている顔のテンプレートを使用した。

顔が検出できない場面

顔の 1/3 ほどが隠れている状態では顔が検出されない。これは上述のテンプレートとマッチングしないためである。

5.3.2 色の抽出

openFrameworks のアドオンである openCV ライブラリを用いて色の抽出を行っている。色の抽出は HSV に変換して二値化を行うため、以下に示す 6 つの手順を踏んでいる。HSV へ

⁴ofxCvHaarFinder, <http://openframeworks.cc/documentation/ofxCvOpenCv/ofxCvHaarFinder.html>

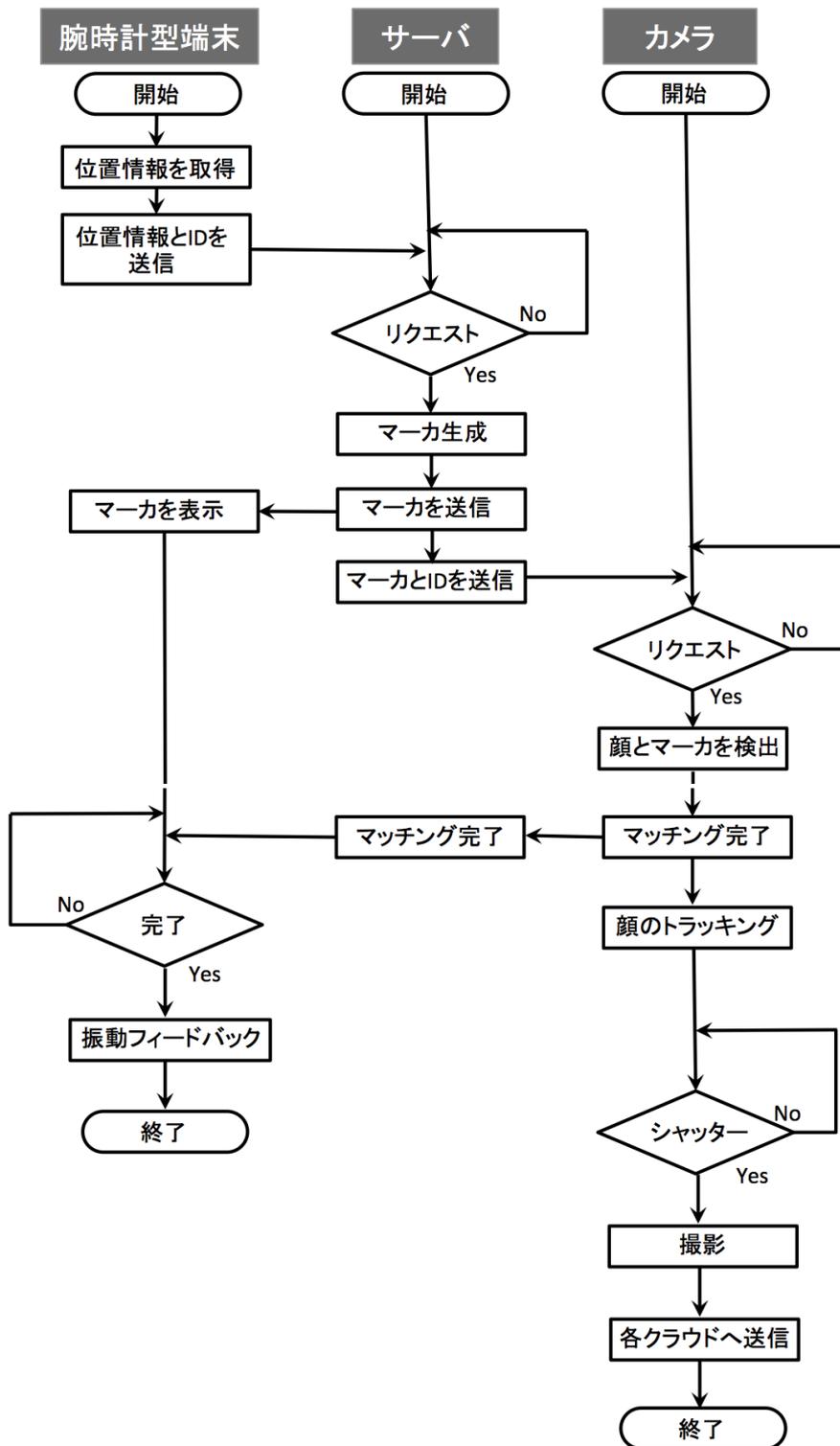


図 5.3: 他人のカメラを用いた場合のシステム実行の流れ

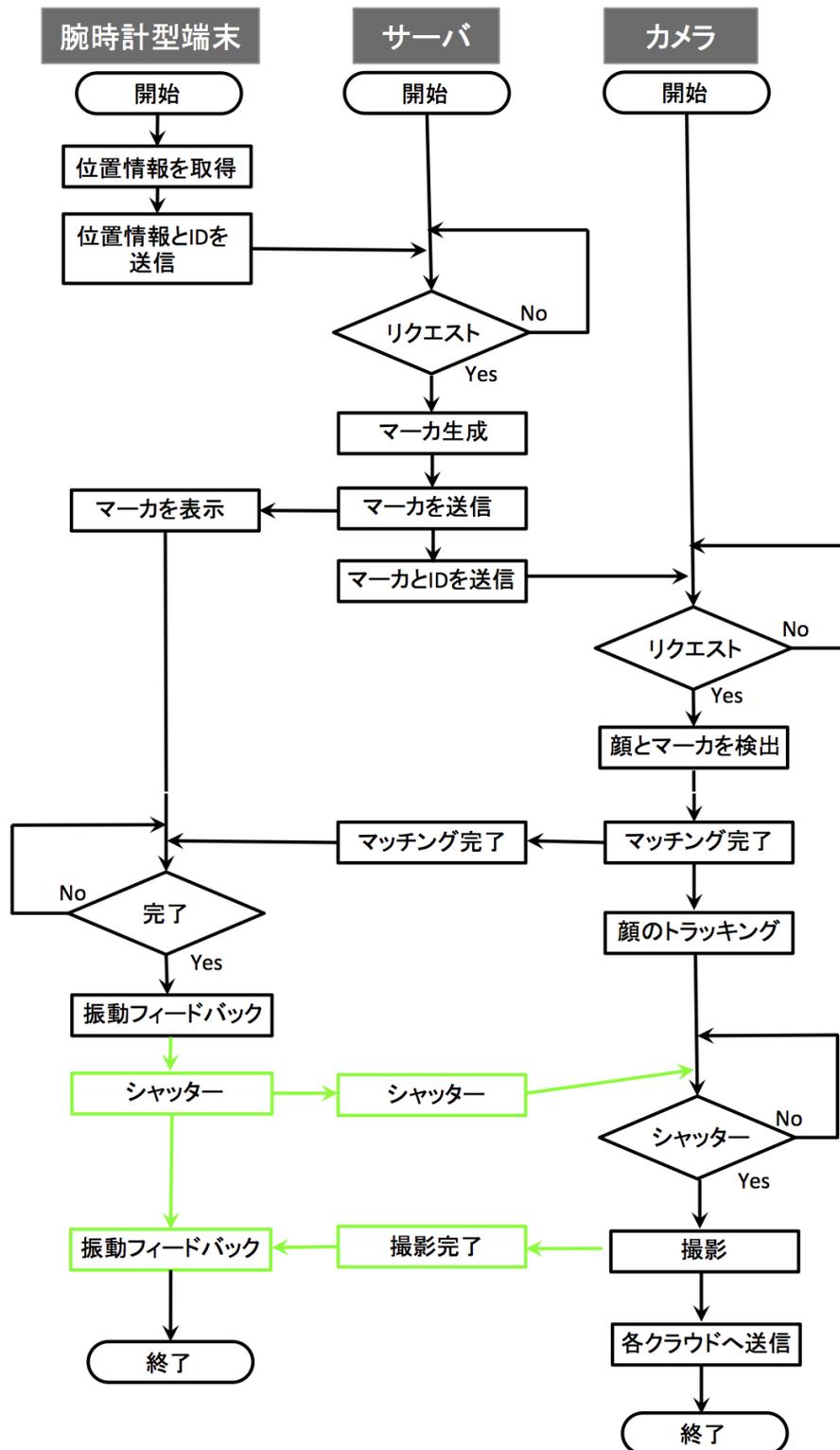


図 5.4: 公共のカメラを用いた場合のシステム実行の流れ



図 5.5: ユーザ識別のためのマッチング

の変換前と変換後の様子を図 5.6, 5.7 にそれぞれ示す。それぞれの図では各ユーザの腕時計型端末のカラーマーカを抽出し、検出した顔と照合を行い、予めサーバから知らされているユーザ ID と紐付けた際の HSV 変換前と HSV 変換後を示す図になっている。左端のユーザはカラーマーカをかざした位置が低かったために検出されなかった。上述の通り、顔の下にカラーマーカを持ってくる必要がある。

1. RGB から HSV に変換
2. HSV 画像を各チャンネルに分解
3. 各チャンネルごとに二値化
4. 各チャンネルの二値化画像の And をとり、マスク画像を作成
5. HSV 画像を各チャンネルの画像を And をとる
6. 各チャンネルの画像を RGB 画像へ戻す

openFrameworks では RGB から HSV への変換はメソッドが用意されているためそれを用いた。各チャンネル毎に抽出したい色とピクセルを比較し、二値化画像を作成する。HSV から RGB へと戻す際には標準のメソッドが見当たらなかったのでウェブサイト⁵を参考に以下のように変換するためのメソッドを作成した。

HSV から RGB を算出

最大値と最小値を求める

本システムでは $S \cdot V$ の範囲が $0 \sim 255$ であるため、以下の式で最大値と最小値を求めることができる。

⁵HSV から RGB への変換計算式, <http://www.peko-step.com/tool/hsvrgb.html>

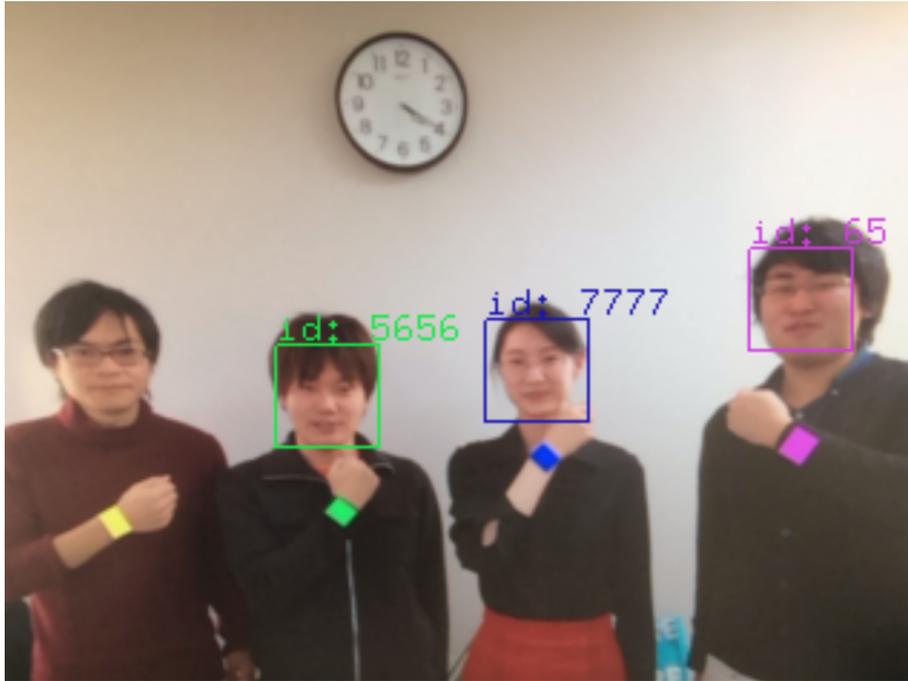


図 5.6: HSV への変換前

最大値 $MAX = V$

最小値 $MIN = MAX - ((S / 255) * MAX)$

RGB 値を求める

本システムでは H の範囲が 0~180 であるため以下の式で RGB 値を求めることができる。

- H が 0~30 の場合

$R = MAX$

$G = (H / 30) * (MAX - MIN) + MIN$

$B = MIN$

- H が 30~60 の場合

$R = ((60 - H) / 30) * (MAX - MIN) + MIN$

$G = MAX$

$B = MIN$

- H が 60~90 の場合

$R = MIN$

$G = MAX$

$B = ((H - 60) / 30) * (MAX - MIN) + MIN$

- H が 90~120 の場合

$R = MIN$

$G = ((120 - H) / 30) * (MAX - MIN) + MIN$

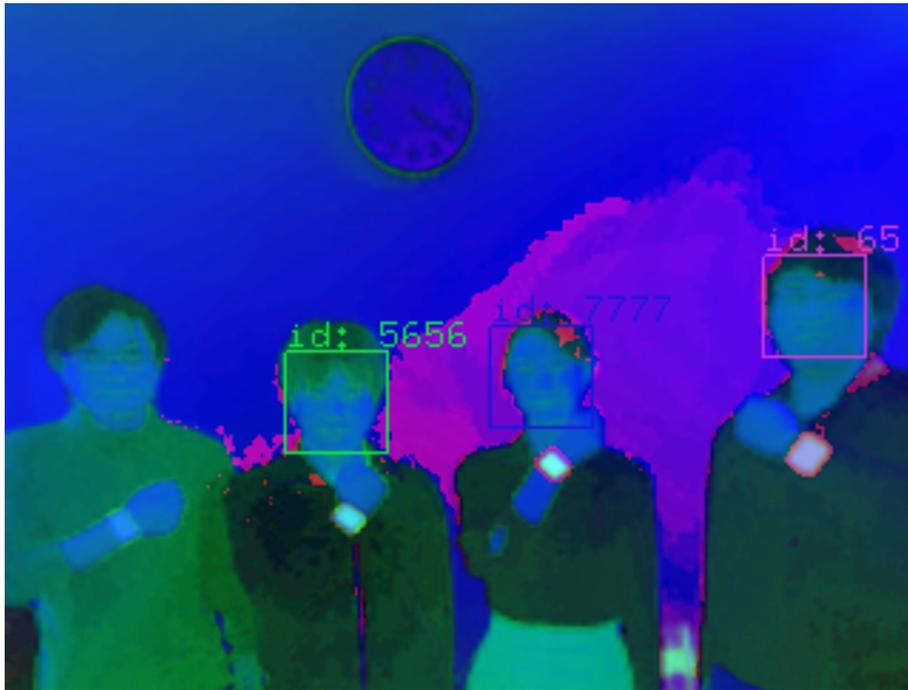


図 5.7: HSV への変換後

- $B = \text{MAX}$
- H が $120 \sim 150$ の場合
 - $R = ((H - 120) / 30) * (\text{MAX} - \text{MIN}) + \text{MIN}$
 - $G = \text{MIN}$
 - $B = \text{MAX}$
- H が $150 \sim 180$ の場合
 - $R = \text{MAX}$
 - $G = \text{MIN}$
 - $B = ((180 - H) / 30) * (\text{MAX} - \text{MIN}) + \text{MIN}$

5.3.3 通信部

カメラ端末の通信ライブラリには iOS 用の通信ライブラリである AFNetworking というオープンソースのライブラリを使用している。各通信は HTTP 通信を使用し、JSON 型で情報の伝達を行っている。

5.4 クライアント - 腕時計型端末

腕時計型端末の処理の流れは上述の 4.2 節で述べた通りである。その際の画面の遷移を図 5.8 に示す。図 5.8(a) ではアプリケーション起動後の画面を示している。開始をタップすることで現在の位置情報がサーバへ送信される。図 5.8(b) はサーバから返されたマーカを表示した画面である。図 5.8(a) から図 5.8(b) までの時間は数秒である。これをカメラにかざし、カメラから認識されるとサーバ経由で腕時計型端末にフィードバックが返され振動する (図 5.8(c))。他人のカメラを用いて撮影する場合はこれで終わりである。公共のカメラを用いる場合には振動後に撮影ボタンが表示され (図 5.8(d))、撮影をタップすることでサーバを経由してカメラのシャッターが切られ、サーバを経由して腕時計型端末に振動フィードバックが返される (図 5.8(e))。

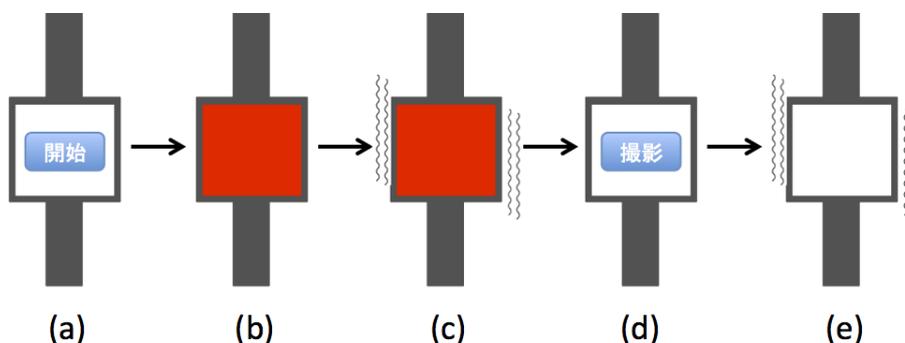


図 5.8: 腕時計型端末の画面遷移

5.4.1 カードの表示

アプリケーション起動時にサーバからカラーマーカを取得し、そのカラーマーカをディスプレイに表示する。今回はカラーマーカを用いているため、画像ではなく、色情報を受け取り、それを画面に表示している。あつかっているカラーマーカの一覧を図 5.9 に示す。カラーマーカで用いる色の選定には開発環境に付属しているカラーパレットに標準で備わっている色を用いた。実際にはブラックを含めた 11 色であるが、ブラックは髪色や服の色と判別が難しいため、今回はブラックを除いた 10 色とした。

5.4.2 通信部

クライアントの通信部分は Android 用通信ライブラリである `okhttp`⁶ というオープンソースのライブラリを用いた。通信は HTTP 通信による GET のみで、位置情報とユーザ ID を送信す

⁶okhttp, <https://github.com/square/okhttp>

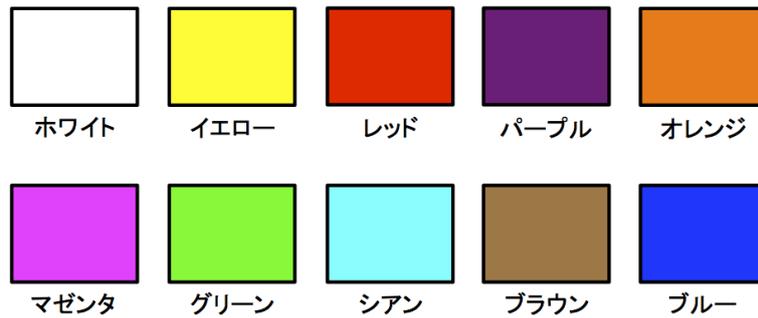


図 5.9: カラーマーカの一覧

るメソッド、カラーマーカを表示するための色を取得するメソッド、シャッターを切るメソッドの3つを作成した。

5.4.3 振動フィードバック

Android 端末で振動を用いるには、android.os パッケージの Vibrator クラスを用いて `Vibrator.vibrate()` を呼び出せばよい。 `Vibrator.vibrate()` には引数が異なる2つのメソッドがあり、引数が1つのメソッドでは振動させる時間をミリ秒で指定する。引数が2つのメソッドでは第一引数で振動パターンを指定し、第二引数で繰り返しを指定することができる。配列の添字が0, 2, 4... がOFFの際の時間（ミリ秒）、添字の1, 3, 5... がONの際の時間（ミリ秒）である。本システムでは簡単のため、引数が1つのメソッドを利用したが、図 5.8(c) と図 5.8(e) で振動パターンをわけた方がユーザにとってわかりやすい設計であると考えられるため、引数が2つのメソッドに変更することを検討している。

5.5 サーバ

サーバは Ruby on Rails を用いて実装している。Ruby on Rails を用いてフレームワークの制約に則ることで通信部の設計が不要となる。元々の設計ではフレームワークとして CakePHP を使用する予定であったが、後述するファイルアップロード用ライブラリを用いることで容易にファイルのアップロードが可能となるため、Ruby on Rails に変更した。データベースには SQLite3 を用いている。サーバで行っている主な処理は腕時計型端末とカメラの仲介である。両者を繋ぐために各腕時計型端末とカメラから位置情報を取得して絞り込みを行い、腕時計型端末に表示される色がどのユーザかをカメラに伝える。カメラはサーバから渡された情報を元に顔と色をマッチングする。この時、カメラに映らなかったマークは位置情報は近かったものの別のグループであるとみなされる。

5.5.1 位置情報による絞り込み

腕時計型端末とカメラ端末の位置情報がそれぞれ送られて来るため、その位置情報に基づいて近くの腕時計型端末とカメラ端末を絞り込む。絞り込みには緯度と経度を用いて行う。Android 端末にて位置情報を取得するためには 2 通りの方法がある⁷。1つ目が GPS を用いる方法であり、2つ目が Network を用いて無線 LAN から位置情報を取得する方法である。本システムでは屋外で使用することもあるため、GPS を用いて取得している。GPS を用いて位置情報を取得するためには LocationManager クラスを用いる。LocationManager に設定することが可能な最高精度の位置情報を取得するための設定を行っているため、およそ半径 10m の精度で位置情報を取得することができる。そのため、本システムでは半径約 10m の距離で前後 10 秒以内に位置情報が送られてきた組を同じグループとみなし、マークを生成し、腕時計型端末とカメラ端末にそれぞれ送信する。同期を取るために位置情報送信後 2 秒毎に腕時計型端末とカメラ端末それぞれからサーバにリクエストを投げ、その時点でグループが決定していた場合、サーバはマーク情報を返す。グループが未決定の場合は null を返す。位置情報による絞り込みの様子を図 5.10 に示す。図 5.10 では、ユーザの位置から半径 10m 以内に 2 つのカメラが存在している場合を例とした。この場合、ユーザ A と B が位置情報を送信した時点でカメラ 1 とカメラ 2 が起動する。カメラ 3 は半径 10m の外にあるため起動しない。

⁷でじういき - Android:位置情報の精度を決める, <http://onlineconsultant.jp/pukiwiki/?Android%20位置情報の精度を決める>

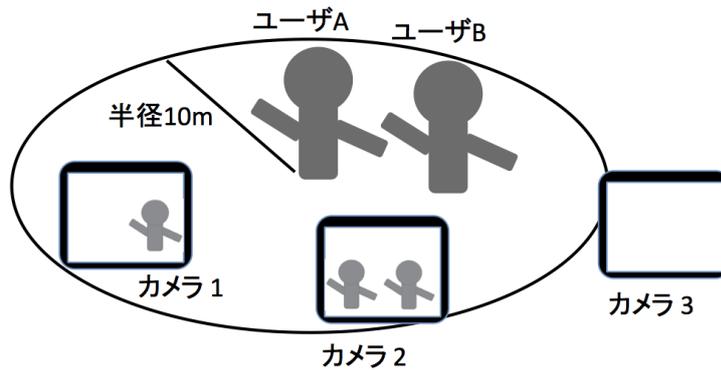


図 5.10: 位置情報による絞り込みの様子

5.5.2 カラーマーカの生成

上述のようにカラーマーカは 10 色使用している。各ユーザが使用している腕時計型端末に表示するカラーマーカを指定する処理について述べる。サーバではリクエストの到着順にカラーマーカを乱数によって生成する。生成したカラーマーカは一時ファイルとしてテキストファイルに保持する。以降、リクエストが来る度にそのテキストファイルから未使用のカラーマーカを取得し、乱数によって指定する色を決定する。しかし、本来この実装は好ましくない。なぜなら、ユーザの着用している衣服の色によってはカメラで識別することが困難な場合が想定されるためである。そのため、改良時にはカメラ端末からユーザ 1 人 1 人の服の色を抽出し、カラーマーカとして割り当てた色との判別が難しいと判断される場合にはそのユーザに割り当てる色を変更するといった処理を追加する。

5.5.3 写真のアップロード・閲覧

写真のアップロードには `paperclip`⁸ という Ruby on Rails 向けオープンソース・ソフトウェアのファイルアップロード用ライブラリを使用した。ファイルはファイルを置くためのフォルダに一次的に保存し、そのパスをデータベースへ格納することで参照を可能にしている。

⁸paperclip, <https://github.com/thoughtbot/paperclip>

第6章 関連研究・関連サービス

本章では本研究と関連している研究及び関連サービスについて述べ、本研究の位置付けについて述べる。

6.1 写真にメタデータを付加して記録

ライフログの観点から写真を記録する場合、写真に付随する情報を載せることは振り返りに多いに役立ち有用といえる。付随する情報というのは撮影された位置情報や撮影された時間、その場所の天候、気温、湿度、気分等があげられる。これらは既に製品として活用されている。例えば、iOS のカメラアプリでは写真を撮影すると自動で位置情報が埋め込められており、写真アプリケーションで閲覧する際には地図上の撮影地にピンが立てられ、撮影した場所が一目でわかるようになっている。その他にも撮影地 viewer¹ というアプリケーションでも撮影地が一覧できる。このように、写真にメタデータを付加して記録することは振り返りの観点からとても有用である。

奥浦ら [3] は Bluetooth の接続履歴から撮影時に近くにいた人を推測し、撮影した写真にメタデータとして付加することで写真の閲覧を支援するシステム Contextual photo browser を開発した。Contextual photo browser は、目的を撮り溜めた写真の閲覧支援としているが、撮影時に取得した情報を写真にメタデータとして付加する点で本研究と関連している。本研究では、撮影時にユーザを識別するため、奥浦らの手法よりも写真に付加するメタデータの精度が高いといえる。しかし、Bluetooth を用いて端末を絞りこむという発想は本研究においても応用が可能なため今後取り入れることを検討したい。また、本研究においても閲覧手法を検討する必要があるが、現時点では閲覧は考慮せず、ライフログとして写真を蓄積するシステムを構築する。

写真にメタデータを付加することは撮影時でなくともかまわない。渡辺 [4] らは写真を複数人で閲覧する際に自然と会話が生まれることに着目し、スライドショーを見ているユーザの様子や音声を記録してデジタル写真にリッチなアノテーションを施すシステム PhotoLoop を開発した。渡辺らによると閲覧時の自然な会話とは主にその写真の感想であることが多いという。つまり、写真に写っているこれは何か、なぜ撮影したのか等の疑問や共感といった会話が行われるため、それをメタデータとして写真に付加している。本研究ではパーソナルクラウドを用いるため、複数人での閲覧は想定していない。しかし、本研究では主に複数人グ

¹撮影地 viewer, <https://itunes.apple.com/jp/app/cuo-ying-deviwer/id888171472?mt=8>

ループでの写真撮影を想定しており、閲覧中にライフログを蓄積することは好ましいことであるため、参考にしたい。

6.2 動的に作成されたマーカの認識

カメラを識別機器として、動的に発信されるマーカを読み取る研究も昔から行われている。2002年には、撮影シーンと 8bit 255 通りの ID を同時に取得可能なイメージセンサ, ID Cam[5] という研究が行われている。ID Cam では撮影時に物体の ID をメタデータとして記録する。物体の ID 発信にはマイコンを用いて LED を点滅制御することで実現しており、それをカメラで読み取る。カメラで動的な信号を読み取る技術は実用化も行われており、ピカピカメラ [6] というスマートフォン向けのアプリケーションが CASIO, Inc からリリースされている。ピカピカメラでは 2 台以上のスマートフォンを用いて 1 台をカメラ、1 台以上を発信機として、発信機側の端末画面にてマーカ画像を点滅させることでカメラ側端末に ID を認識させ、マーカ上にスタンプを重畳表示するアプリケーションである。本研究では、位置情報と時間という制約を加えてマーカを動的に作成することで上述のシステムよりも少ない種類のマーカで認識することが可能である点で優れているといえる。

6.3 カメラの遠隔操作

近年セルフィー棒² (自撮り棒) という商品が流行している。自撮りが流行して久しいが、自撮りは自身を入れた上で背景も写るように撮影することが難しい。これを解決するために、セルフィー棒は棒の先端にスマートフォンを設置し、棒の取っ手にあるスイッチから先端に設置されているスマートフォンのカメラのスイッチが切れるという画期的な商品である。セルフィー棒を用いることで、用いない場合よりも距離が稼げるため、視野の広い撮影が可能となる。ただし、セルフィー棒は棒を把持している手でシャッターを切っているため、遠隔操作とは言い切れない。Chu ら [7] は離れた場所からハンドジェスチャを用いてカメラを操作することで、自身の姿を撮影するシステムを開発した。ユーザへのフィードバックのために正面にディスプレイを置き、プレビューを行う。ユーザが離れた場所から操作して自身の姿を撮影する点で関連している。本研究では公共カメラを使用するため、カメラと各ユーザのマッチングを考慮する必要があるものの、カメラに写る複数人の各ユーザを識別することで写っているすべてのユーザがカメラの操作が可能となる。

²東南アジアで話題沸騰! 自撮りのためのセルフィー棒 (Selfie Stick) とは?, <http://www.sekai-lab.com/times/?p=1431>

6.4 撮影した写真の共有

東芝社が販売している FlashAir³ という無線 LAN 機能を備えた SDHC メモリカードがある。これはデジタルカメラで撮影した写真をこの SDHC メモリカードに保存することで複数のスマートフォンやタブレット端末等のブラウザから SDHC カードに保存されている全ての写真を閲覧することができる。さらに気に入った写真を自身のスマートフォンに保存することが可能となる。これは SDHC メモリカードがアクセスポイントとして機能し、それに接続している端末が SDHC メモリカードにローカル接続することで実現している。本研究では撮影端末に保存された写真全てをユーザに公開することはプライバシーの観点から好ましくないと考え、撮影された写真に写っているユーザにのみその写真を共有することができる仕組みを考案・実装した。

6.5 写真に写っている人の顔の識別

写真に写っている人物を識別する技術は既に製品化の段階に入っている。例えば、Apple の iPhoto⁴ や Google Picasa⁵ では写真を取り込むことで人物を識別し、同一人物と推測される顔をタグ付けしてまとめることが可能である。Google Picasa は API が公開されており、顔の識別にこの API を用いてクラスタリングする研究が行われている [8]。顔の識別技術は年々進歩しているものの、撮影後に識別するため撮影環境によって誤差が生じてしまう。

本研究では、マーカを用いることで撮影最中に識別可能なため、トラッキングを行いユーザの移動差分を取得することが可能となるなど応用が効くことや特徴量を用いる顔の識別よりは精度の面で優位と言える。また、撮影の最中に写っている人の情報とマッチングし、トラッキングするため、動画撮影においても人の識別が可能となる。動画からフレーム毎に顔の識別をすることはサーバ負荷が大きいと困難であるが、本システムを用いることにより、動画の時間とユーザの位置を対応付けることが可能となる。さらに、撮影時点で撮影者に何らかの付加情報を示すことが可能である。

6.6 ビデオカメラを用いた個人識別型位置検出

本研究で想定している公共カメラが普及した未来というのはユビキタス環境の一部である。ユビキタス社会において、ビデオカメラを用いて個人を識別したいというニーズは昔から存在している。既存技術であれば RFID タグ⁶ やビーコン⁷ のような電磁波を発する機器とビデオカメラを組み合わせることで位置検出を行うことが可能である。河合 [9] らは幼児教育分野への応用を考え、電磁波は幼児への影響が懸念されるため好ましくないとビデオカメラ

³FlashAir, <http://www.toshiba.co.jp/p-media/flashair/>

⁴iPhoto, <https://www.apple.com/jp/mac/iphoto/>

⁵Picasa, <http://picasa.google.co.jp/>

⁶株式会社デンソーウェーブ - RFID とは, <https://www.denso-wave.com/ja/adcd/fundamental/rfid/>

⁷Wikipedia - ビーコン, <http://ja.wikipedia.org/wiki/ビーコン>

とモーションセンサを組み合わせて位置検出を行うシステムを開発した。これは幼児の衣服にモーションセンサを装着し、加速度や歩数、歩幅を取得することによって幼児の動きを検出する。ビデオカメラでは入力画像から差分を取り、幼児の動きと差分を照合するという手法である。この手法はセンサデータと動作差分の照合精度が課題となっており、同様の手法で精度向上を目指した研究が丸橋 [10] らや岩本 [11] らをはじめ盛んに行われている。

本研究では、現在の実装においては個人の識別にカラーマーカを用いているのみであるが、カラーマーカのみでは識別が難しいのではないかと考え、4.4.2 ユーザ識別精度で述べたように腕時計型端末から加速度を取得してリアルタイムに照合することで精度向上を図れないかと検討している。

6.7 カメラが自動で撮影するシステム

写真を自動で撮影する製品としてライフログのすすめ [12] にて紹介されているセンスカム⁸の類似製品⁸やSONY社のパーティショット⁹という製品がある。センスカムは首からカメラを吊り下げて日常を過ごしているだけでおおよそ30秒間隔で写真を撮り続け記録することが可能なカメラである。これにより、ユーザは写真とともにその日を鮮明に振り返ることができる。Party-shotは設置しておくだけで自動で人を検出して撮影するため、カメラマンが存在しない。そのため、カメラを意識しない自然な写真が撮影でき、カメラを撮っていたために写真に写れない人というものが存在せず、全員が写真に写ることが可能である。さらに、Party-shotの後継機Cyber-shotでは、スマイルシャッター¹⁰という笑顔を検出して自動でシャッターを切る機能が追加されている。この機能によってただ全員が写真に写るのではなく、被撮影者が笑顔の瞬間を逃さず写真に残すことが可能となった。

本研究では他人のカメラを用いた場合は人がシャッターを切り、公共カメラを用いた場合においても複数人の意思決定によってシャッターが切られるため、カメラが自動でシャッターを切るということには行っていない。しかし、ストレージを考慮する必要がないのであれば、写真を後から選定すれば良いという仮定の下、上述のシステムのように時間によって撮影することや笑顔を検出して撮影する、全員が目を開いている際に撮影するといった自動撮影を取り入れることは有用であると考えられるため、今後検討する予定である。

6.8 写真のブラウジング・探索手法

写真撮影が手軽になったことにより、写真の枚数が膨大になっている。そこで写真のブラウジングの研究も盛んに行われている。綾塚 [13] は撮影された時間が短い写真群は写真同士の結びつきが強いが動画ほど連続してもいないという写真群を擬連続的メディアと提唱し、写真群を編集するためのアプリケーションAlbum Weaverを開発した。これによりグループ分け

⁸NARRATIVE CLIP 2, <http://getnarrative.com/>

⁹Party-shot, <http://www.sony.jp/cyber-shot/party-shot/>

¹⁰SONY - スマイルシャッター, http://www.sony.jp/cyber-shot/products/DSC-T70/feature_1.html

が容易となるため目的の写真を探すことも容易になる。Album Weaver はグループ分けを手動で行うのに対し、Easy-Album[14] では写真を画像解析し、被写体を認識し自動でグループ化を行う。このようにグループ化を行うことで目当ての写真がある場合は探索時間が短くなるが、必ずしも目的の写真が明確とは限らない。捧 [15] らはユーザの曖昧な記憶からでも目的の写真を探せるようにライフログに着目し、時間・位置・人物情報という3つの情報を横断的に検索することができるシステムを開発した。

上記システムでは目的の写真が曖昧ではあっても目的の写真というものが存在していた。しかし、Kirk[16] らは写真のブラウジングにおいてユーザが求めていることは写真の検索ではなく選定であると述べている。つまり、同じような種類の写真からお気に入りの1枚を選ぶことをユーザは求めているが、写真のブラウジング・探索手法の研究では検索の効率化ばかりに注力されているというのである。写真のブラウジングでは効率を高めれば良いということではなくユーザが満足する枠組みを構築する必要がある。

そこで、膨大な写真を楽しく閲覧するためのシステムというものも提案されている。美崎 [17] は記憶する住宅という、住宅の至るところに写真を投影することにより写真の常時ブラウジングを行うシステムを開発した。このシステムを用いることにより、日頃閲覧する機会の少ない写真を日常生活を通して閲覧することが可能になる。北村 [18] らは写真を生物のように常に動かしながら特定のレイアウトへ収束させるといった視覚的にユーザを楽しませる柔軟なシステムを開発した。PhotoLab[19] はメタデータに基づいて3次元空間に画像を表示し、ユーザの閲覧したい画像が派生するのを支援するシステムである。このように明確な検索対象がない中で楽しく閲覧しながらもユーザの欲求を満たすシステムにも注目が集まっている。

第7章 まとめと今後の課題

7.1 まとめ

自身の写っている写真を友人等のグループ内でシェアすることが困難な点に着目し、誰のカメラかによらず撮影時に被撮影者を識別することで撮影した写真に写っている人のパーソナルクラウドに写真を自動で蓄積するライフログシステムを提案し、開発した。本システムを用いることで撮影後に被撮影者一人ひとりに写真を振り分ける煩わしさが解消される。そしてSNS上で共有される場合においてプライバシーの問題が危惧されるが、本システムを用いることで予期せぬ人に写真を見られるといった心配が解消される。また、撮影者がいないことを考慮して公共カメラという未来環境を想定し、連携手法を提案した。これにより、写真のシェアが自動化され、自身の写っている写真を自身で管理できるようになる。さらに蓄積された写真には撮影者、位置情報、一緒に写っている人といったメタ情報が取得可能となるため、ブラウジング手法に応用可能なことを示した。

7.2 今後の課題

今後の予定として、開発したシステムの評価を行う。評価の方法としては、ユーザ識別精度の計測と実際に被験者に使用してもらいアンケート調査を行う予定である。ユーザ識別精度では屋外と屋内での比較、一度に撮影可能な最大人数の検討、適正距離の導出等を行う。アンケート調査では撮影した写真を既存のサービスを用いて共有する場合と本システムを用いて共有する場合の使用感について定性評価を行う。

謝辞

本論文を執筆するにあたり，指導教員である田中二郎先生をはじめ，志築文太郎先生，高橋伸先生，三末和男先生には，ゼミ等を通して丁寧なご指導や貴重なご意見をいただきました。心より御礼申し上げます。また，インタラクティブプログラミング研究室の皆様には，日頃の研究室での生活を通して多くのアドバイスをいただきました。深く感謝申し上げます。最後に，ここまで私を支えてくださいました友人や家族，お世話になった全ての方々に心より感謝いたします。

参考文献

- [1] 山中太一郎. “ウェアラブル端末のインパクトと市場化への課題”. 野村総合研究所, 2014.
- [2] 宝出幸久. “国内 Storage in the Cloud 市場 2013 年の実績と 2014 年～2018 年の予測”. IDC Japan, 2014.
- [3] 奥浦圭一郎, 牛越達也, 河野恭之. “Contextual photo browser:写真参与者情報を利用した写真管理システム”. 情報処理学会研究報告. HCI, ヒューマンコンピュータインタラクション研究会報告, Vol. 2011 No. 1, pp. 1-7, 2011.
- [4] 渡邊恵太, 塚田浩二, 安村通晃. “PhotoLoop: 写真閲覧時の自然な語らいを活かしたスライドショーの拡張”. ヒューマンインタフェース学会論文誌 11(1), pp. 69-76, 2009.
- [5] 松下伸行, 日原大輔, 後輝行, 吉村真一, 暦本 純一. “ID Cam: シーンと ID を同時に取得可能なイメージセンサ”. インタラクション 2002, 8 pages, 2002.
- [6] CASIO, Inc. “ピカピカメラ”. <http://www.casio-isc.com/ja/> (2015 年 1 月 9 日)
- [7] S. Chu and J. Tanaka. “Interacting with a Self-portrait Camera Using Motion-based Hand Gestures”. *Proceedings of the 11th Asia-Pacific Conference on Computer-Human Interaction*, pp. 93-101, 2013.
- [8] A. Gomi, T. Itoh. “A Personal Photograph Browser for Life Log Analysis based on Location, Time, and Person”. *ACM Symposium on Applied Computing, Multimedia Visualization Track*, pp. 1250-1257, 2011.
- [9] 河合純, 永田章二, 清水宏章, 新谷公朗, 金田重郎. “モーションセンサとビデオカメラを用いた個人識別型位置検出手法”. 情報処理学会研究報告, 2004-UBI-5(1), pp. 1-8, 2004.
- [10] 丸橋尚佳, 寺田努, 塚本昌彦. “カメラと加速度センサを用いた複数移動物体の同定手法”. 情報処理学会研究報告, 2010-UBI-27(10), pp. 1-8, 2010.
- [11] 岩本健嗣, 小林亜令, 太田慎司, 西山智. “環境カメラと携帯端末を用いた位置特定手法”. 情報処理学会研究報告, 2006-DPS-129(14), pp. 71-76, 2006.
- [12] G. Bell, J. Gemmill. “*Total Recall: How the E-Memory Revolution will Change Everything*” Dutton Adult, 304 pages, 2009.

- [13] 綾塚祐二. “Album Weaver: 密度の高い写真群からアルバムを紡ぐ”. インタラクティブシステムとソフトウェアに関するワークショップ (WISS2007) 予稿集, 6 pages, 2007.
- [14] J. Cui, F. Wen, R. Xiao, Y. Tian, and X. Tang. “EasyAlbum: an interactive photo annotation system based on face clustering and re-ranking”. *CHI 2007 Proceedings*, pp. 367-376, 2007.
- [15] 捧隆二, 佃洸撰, 中村聡史, 田中克己. “時間・空間・人物情報に基づくインタラクションによるライフログ画像の探索手法の提案”. *DEIM Forum 2012*, 8 pages, 2012.
- [16] D. Kirk, A. Sellen, C. Rother, and K. Wood. “Under-standing Photowork”. *CHI 2006 Proceedings*, pp. 761-770, 2006.
- [17] 美崎薫. “記憶する住宅-55万枚のデジタルスキャン画像の常時 スライドショー・ブラウジングによる過去記憶の甦りの実際”. *インタラクション 2004*, pp. 129-136, 2004.
- [18] 北村喜文, 高本恵介, 高嶋和毅, 伊藤雄一, 横山ひとみ, G.Liu, S.Subramanian. “インタラクティブで柔軟なデジタル写真群動的表示法”. *2013 Information Processing Society of Japan*, pp. 40-47, 2013.
- [19] 堀辺宏美, 伊藤貴之. “PhotoLab: ユーザの思考を支援する画像閲覧インタフェースの開発”. *情報処理学会研究報告, 2008-CG-131(6)*, pp. 31-36, 2008.