

# スマートフォンカメラを用いた群集人数推定システム

小島 颯平<sup>1</sup> 内山 彰<sup>1</sup> 廣森 聡仁<sup>1</sup> 山口 弘純<sup>1</sup> 東野 輝夫<sup>1</sup>

概要：本稿では、我々が開発している、スマートフォンカメラによる群集の俯瞰画像からの群衆人数推定を行うシステムを紹介する。同システムでは、群衆が移動していることを仮定し、短い時間間隔で撮影された2つの画像の差分を取ることで群衆が存在する動体領域を抽出する。1人あたりの占有面積は、撮影高度や撮影傾度、画像内での位置および移動ベクトルにより異なるため、3次元物体移動シミュレーションの可視化データの分析結果に基づき被写体サイズ推定モデルを構築し、高精度な群衆人数推定を実現している。性能評価のため、Unityで群衆の動きを再現し得られた画像と、実環境における性能評価としてJR大阪駅前前で撮影した動画から一定間隔で抽出した画像に対してそれぞれ提案手法を適用したところ、Unityによるシミュレーションでは絶対誤差が10.2%、実環境では12.0%となり、高精度な群衆人数検出を実現できている。

## 1. はじめに

近年、都市街区などにおける人々の存在や行動を把握し、混雑予想や大規模イベント推定に活用する事例がみられるようになってきている。イベント会場や駅前、あるいは道路などの特定のスポット空間における群衆の存在を把握することができれば、交通機関の乱れによる混雑状況や屋外イベントにおける集客状況、帰宅難民による道路占有状況などが把握でき、迅速な代替輸送計画立案や災害時の避難者誘導、施設活性化など様々な応用が期待される。

群衆検出には映像を用いた方式 [1] やGPS情報を用いて群衆密度推定を行う方法 [2] が提案されているが、様々な地点で動的に形成される群衆を可視領域が限定された固定カメラのみで捉えることは非現実的であったり空間的粒度が大きいためスポット空間における群衆は開くには不十分である。

これに対し我々は、多くの人々が所有するスマートフォンによりビルの上階など比較的高い場所から撮影された俯瞰画像を用いた群衆人数推定方式を開発している [3]。近年では参加型センシングにより騒音や大気汚染などの発生状況を広域で把握する研究が多くなされている。本研究はこのような参加型センシングと組み合わせることを想定しており、各スポットでの群衆密度が推定できるようになれば、広範囲にわたる群衆密度分布の把握が実現できる。

画像からの群衆人数推定方式として、従来から知られて

いる人検出の手法 [4] を適用しその検出人数を数え上げる方法が考えられる。しかし、群衆内の人は身体の境界が明確に現れにくいいため、検出に失敗しやすくなるという問題がある。また、これらの方式の多くは移動物体（動体）が写っている領域（動体領域）を抽出し、動体領域のみに対して人検出を行うことで、検出精度の向上を図っている。

このような課題に対し、提案システムでは、群衆が存在すると推定される領域の面積に基づく群衆人数推定法を採用している。同システムでは、群衆が移動していることを仮定し、経験的に得られた0.7秒の時間間隔で撮影された2つの画像の差分を取ることで背景を除去し、動体領域を抽出する。ここで、画像内の各人物が画像領域に占める面積が分かれば、動体領域を占める人数の推定が可能となるが、実際には、近距離の被写体は画像内で大きく、遠距離の被写体は小さくなるため、同一人物であっても画像内の位置によって一人あたりの占有面積は異なる。そこで提案システムでは、被写体サイズ推定モデルを構築し、カメラの高度と傾度、画像内の被写体位置と移動方向に応じた1人あたりの占有面積を推定することで、高精度な群衆人数推定を実現させる。モデルの構築には、3次元空間シミュレータ Unity [5] により生成された網羅的な被写体のサイズを用いる。提案システムでは、スマートフォンの気圧センサから得られるカメラの高度と加速度センサから得られる傾度に対応するモデルを用いることで、動体領域の検出と組み合わせた群衆人数の推定が可能となる。

性能評価のため、Unityで群衆の動きを再現し得られた画像と、実環境における性能評価としてJR大阪駅前前で撮影した動画から一定間隔で抽出した画像に対してそれぞれ

<sup>1</sup> 大阪大学大学院情報科学研究科  
Graduate School of Information Science and Technology, Osaka University

提案システムを適用したところ，Unity によるシミュレーションでは絶対誤差が 10.2%，実環境では 12.0%となり，高精度な群衆人数検出を実現した．

## 2. 群衆人数推定手法

### 2.1 概要

文献 [3] の手法の基本アイデアは，群衆を 1 人 1 人数え上げるのではなく，群衆の大きさ，すなわち画像内において群衆が写っている面積（占有面積）から人数を推定することである．群衆内においては人物の境界が曖昧になり特徴量による認識が難しいため，このように群衆を 1 つのまとまりとしてとらえ，およその人数を推定するアプローチが有効である．提案手法では背景画像が無い環境を想定しているため，厳密には短い時間間隔で撮影された 2 枚の画像に対して差分を取った領域を群衆が存在する領域（動体領域）として検出し，その面積を群衆の占有面積と見なす．このとき，被写体の位置および撮影場所の高度とカメラの傾度によって画像中の見かけ上の大きさが変化するため，同時に占有面積も変化する．さらに，動体領域は被写体の移動ベクトルによっても占有面積が変化する．

提案手法では，動体領域面積の大きさに応じて被写体サイズ推定モデルに基づく人数推定を適用する．被写体サイズ推定モデルに基づく人数推定は，1 人あたりの想定される占有面積から動体領域内の人数を推定する．被写体サイズ推定モデルに基づく人数推定では，適切な 1 人あたりの占有面積を決定するため，平均的な体格の人物ならびに平均移動速度を仮定し，事前にあらゆる高度，傾度，人物の位置，移動ベクトルに対する占有面積を実際に計算し，モデル化する．モデルには身体のみだけでなく，歩行時の足や腕の動きの大きさも含まれる．提案手法では，3次元空間シミュレータ Unity [5] を用いてこれらのパターンを再現し，被写体サイズ推定モデルを構築する．被写体サイズ推定モデルの入力に必要な被写体の移動ベクトルは，2 枚の画像から得られるオプティカルフローによって推定する．

以降ではモデルに必要なカメラの画角，撮影時の画像の縦横のピクセル数，撮影時のカメラの高度および地面に対する傾きは，既知であると仮定する．実際には，Android 端末におけるカメラの画角および画像の縦横のピクセル数は API により取得可能である．高度や撮影傾度はそれぞれスマートフォンに搭載された気圧センサおよび加速度センサから推定する．

### 2.2 被写体サイズ推定モデル

被写体サイズ推定モデルでは，カメラの高度と傾度ならびに被写体の移動ベクトルと画像内の位置から動体領域の面積を推定する．モデルの構築にあたっては，人は一定速度で移動し，短時間では直進しかしないものとする．これ



図 1 0 度のときの動体領域

らのあらゆるパターンをシミュレーションで再現し，その時の面積を実測することでモデルを構築するが，文献 [3] の幾何的な特性により 0 度から 90 度の範囲のみを対象にモデルを構築すれば良い．

すなわち移動ベクトルの方向は 360 度存在するが，そのうち 0 度から 90 度（XY 平面における第 1 象限）のみの範囲を対象にモデルを構築すれば，全ての移動ベクトルに対して被写体サイズ推定が実現できる．第 1 象限の  $[0, 90]$  度の範囲であらゆる移動方向についてモデルを構築することが望ましいが，本研究では簡単のため作成するモデルを 0 度，30 度，60 度，90 度の 4 つとした．

モデルの作成には平均的な体格の人物（標準被写体）が立っている足元を基準点としたときの足元の座標と標準被写体が平均速度で歩いたときに，2.3 節で述べる方法に従って抽出された動体領域の面積を用いる．得られた標準被写体の位置および移動ベクトルと面積に対して線形回帰を適用することで被写体サイズ推定モデルを構築する．

被写体サイズ推定モデルを利用する際には，画像の垂直方向を  $y$  軸とした時に移動ベクトルが  $y$  軸となす角度のうち小さい角  $\theta$  を算出し，被写体サイズ推定モデルに与える．本研究では，簡単のため用意している 4 種類の移動方向のうち最も近い角度を与える．撮影高度および撮影傾度を  $h_c, \theta_c$ ，また移動方向を  $\theta$ ，人の位置を  $y$  とする．また撮影高度および撮影傾度によって 4 つのモデルを選択する関数を  $m(\theta_c, h_c)$  とする．これらによって得られるモデルを  $f(y, \theta, m(\theta_c, h_c))$  とすると次のように表される． $a(h_c, \theta_c, \theta), b(h_c, \theta_c, \theta)$  は係数である．

$$f(y, \theta, m(\theta_c, h_c)) = a(h_c, \theta_c, \theta)y + b(h_c, \theta_c, \theta) \quad (1)$$

例として，以下に高度 7.5m，傾度 25 度の被写体サイズ推定モデルを示す．図 1 が Unity 上で 0 度の方向に移動する標準被写体を撮影した画像と抽出された動体領域を重畳した画像で，図 2 がこれらの画像から構築された被写体サイズ推定モデルである．なお画像の位置の原点は左上を原点とする．



図 3 動体領域検出の例

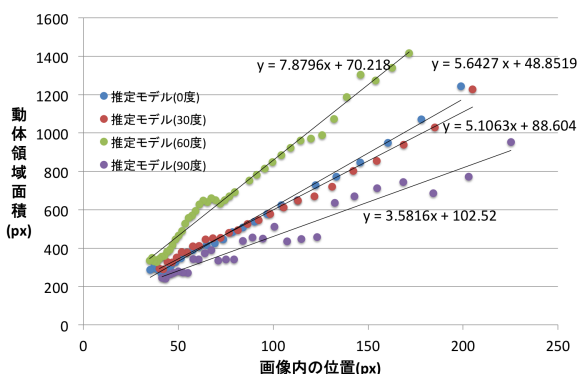


図 2 被写体サイズ推定モデルの例 ( $h_c = 7.5m, \theta_c = 25$  度)

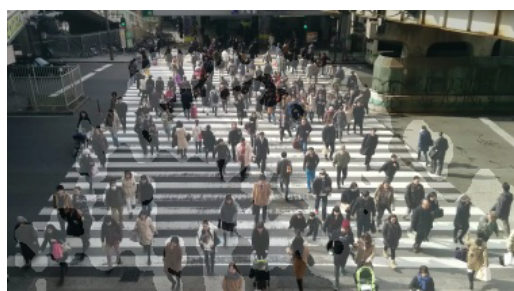


図 4 抽出された動体領域

### 2.3 移動物体の領域検出

動体領域の抽出には図 3(a) に示されるような画像を用いる。まず入力画像に対して、5 分の 1 の解像度に縮小を行う。これはスマートフォン上でこのアルゴリズムが実行されることを想定し、計算負荷を軽減させるためである。文献 [3] にある手法を用いて 2 枚の画像差分を得た結果が図 3(b) であり、これに判別分析法 [6] を適用した結果が図 3(c) である。

また入力画像に対して適切に移動領域が検出できたかを確認するため、図 3(c) と入力画像 1 を重畳した画像が図 4 である。2 枚の入力画像から正しく動体領域を抽出できていることが確認できる。

### 2.4 被写体サイズ推定モデルを用いた群衆人数推定

動体領域と背景領域を表す二値化画像と、2.2 節で導出した被写体サイズ推定モデルを用いて群衆人数推定を行う。

被写体サイズ推定モデルに必要な被写体の移動ベクトルの方向はオプティカルフローにより推定する。ベクトルの大きさは対象環境における平均的な移動速度が事前に与えられるものとする。これにより、抽出された動体領域に対して被写体サイズ推定モデルを適用することができる。群衆人数の推定は図 5 に示す青い四角形で囲われた動体領域ごとに行う。この 1 つ 1 つの独立した動体領域を群衆ブロックと呼ぶ。二値化画像から OpenCV の findContours 関数を用いる事で各群衆ブロックを形成する輪郭の点の座標集合を取得することができる。

各群衆ブロック  $b$  のどの部分に人が存在するか分からないので、群衆ブロック  $b$  における平均的な被写体の 1 人あたりの動体領域面積として、 $b$  の重心の位置  $(g_x, g_y)$  を基準とし被写体サイズ推定モデルによって推定された面積を用いて  $b$  内に存在する群衆人数を推定する。このため群衆ブロック  $b$  内に存在する、ORB 特徴量による特徴点抽出と、Lucas-Kanade 法 [7] による各特徴点の軌跡推定を用いることによって得られたオプティカルフローを 0 度から 90 度の範囲の同一面積となるベクトルに変換し、それらの平均をとる。

平均化されたオプティカルフローのベクトルを  $\vec{v}_b$  とすると、 $\vec{v}_b$  は群衆ブロック内の平均的な移動ベクトルを表していると考えられるため、これに対応する被写体サイズ推定モデルを用いて群衆ブロック  $b$  内の人数を推定する。すなわち、 $b$  内の人数  $N_b$  は以下の式で表される。

$$N_b = \frac{S_b}{f_{\Theta(\vec{v}_b)}(g_y)} \quad (2)$$

ここで  $S_b$  は群衆ブロック  $b$  の面積、 $\Theta(\vec{v}_b)$  は移動ベクトル  $\vec{v}_b$  の移動方向とする。

こうして得られた重心位置における人の面積によって群衆ブロックの面積を除することで 1 つの群衆ブロックの人数を推定することができる。最終的に、それらの総和をとることで画像内の人数を推定する。

事前実験として、Unity を用いて群衆の動きを再現し、そのカメラから見られる映像を経験的に得られた 0.7 秒の間隔で画像を撮影し擬似的に写真の撮影を行い、2.4 節の手法を適用したところ図 6 の結果が得られた。これはモデ





図 5 群衆ブロック検出例

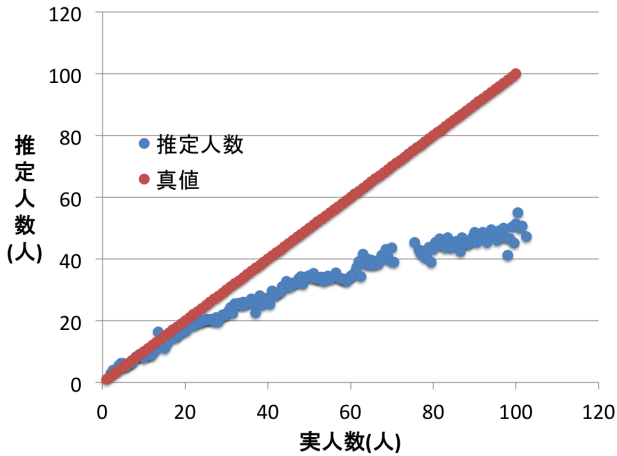


図 6 実人数に対する推定人数

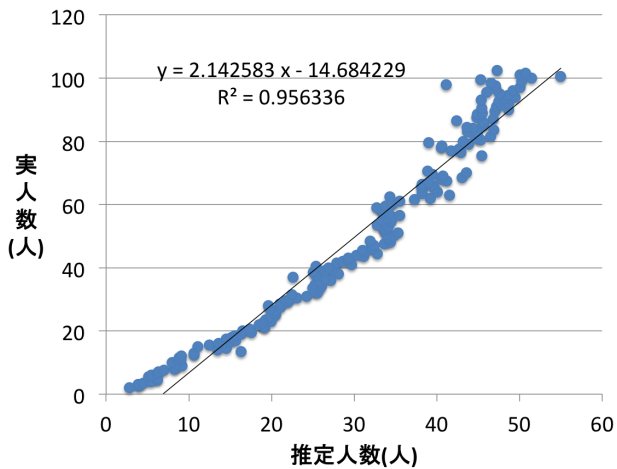


図 7 推定人数に対する実人数

ルの構築時の人の位置は足の位置となっているが、群衆ブロック内の平均的な人の基準位置を動体領域の重心としているため、適切な位置が選択されていないことによる誤差である。この想定された誤差を補正するために、図 6 の横軸を推定人数、縦軸を実人数とした図が図 7 である。図 7 に示すように  $R^2$  値が 0.95 を越えており、高い線形性が見られる。これを線形回帰を適用することで入力を推定人数とし、実人数への補正が可能となる。

### 3. 群衆人数推定システム

本章では 2 章 を用いた群衆人数推定システムの全体の

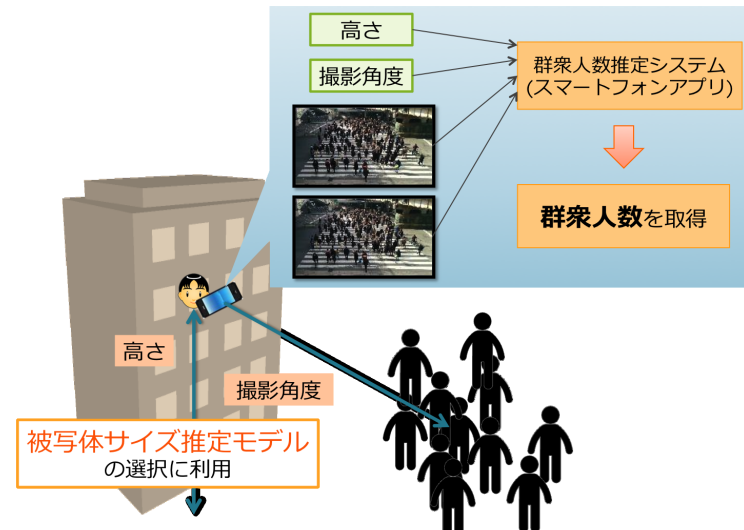


図 8 提案システムの想定環境



図 9 提案システムの概要

概要を示す。まず想定環境を図 8 に示す。撮影者は群衆を俯瞰できる高所から群衆を撮影し、非常に短い時間間隔で撮影された 2 枚の画像を得る。撮影高度はスマートフォンの気圧センサから推定を行い、撮影傾度は加速度センサから推定を行う。撮影高度および撮影傾度と 2 枚の画像を入力として群衆人数推定システムアプリケーションに入力される。2 章のシステムの概要が図 9 である。このシステム内で 2 枚の入力画像から動体領域面積、人の移動方向が計算され、撮影高度および撮影傾度から適切な被写体サイズ推定モデルが選択され群衆人数を得ることができる。

## 4. 性能評価

### 4.1 実験環境

提案手法の性能評価を行うため、Unity を用いて群衆の動きを再現し、そのカメラから見られる映像を経験的に得られた 0.7 秒の間隔で画像を撮影し擬似的に写真の撮影を行った。また実環境における評価として、JR 大阪駅前の交差点において歩行者信号が青の期間を対象に Android 端末 Nexus 5 を用いて撮影した動画から 0.7 秒間隔で画像を抽

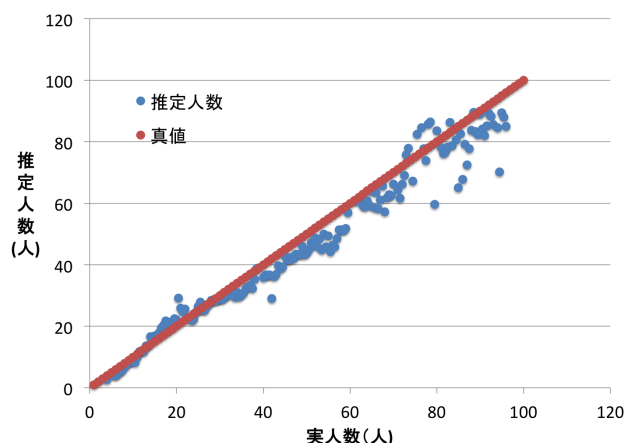


図 10 実人数に対する検出人数 (シミュレーション)

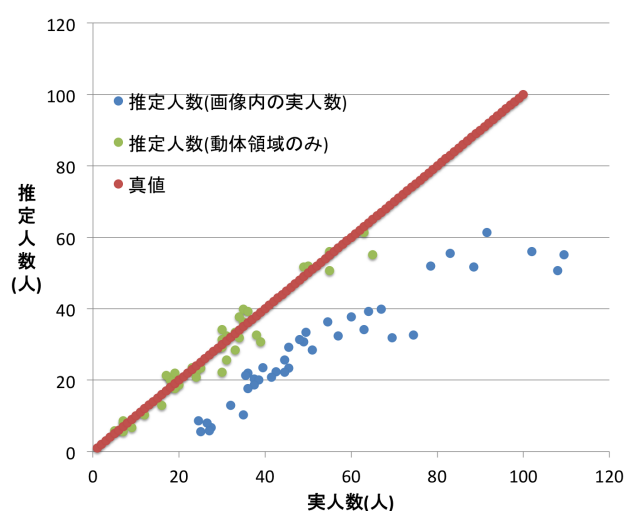


図 11 実人数に対する検出人数 (実環境)

出することで入力画像を得た。各入力画像について目視により人の頭部の数を数えることで真値を得た。Java および画像処理ライブラリ OpenCV2.4.10 を用い、Mac OS10.9.4 上で動作するよう提案手法の実装を行った。

#### 4.2 群衆人数の変化による影響

Unity によるシミュレーションの人数を変化させ群衆人数の変化による性能変化の結果を図 10 に示す。

シミュレーション環境に対して提案手法を適用した場合の絶対誤差は 10.2% となり、高精度な群衆人数検出が実現できた。

続いて実環境における性能変化の結果を図 11 に示す。図 11 のとおり、画像内の実人数に対する性能は低くなっている。これは図 12 に示されるように、本来人が存在し動体領域として抽出されるべき部分が人の動きが小さかったため、うまく抽出できていなかったことによるものである。図 11 における推定人数の動体領域のみというのは、動体領域内に存在する人数を真値として、性能評価をした結果である。この場合の推定人数の絶対誤差は 12.0% とな

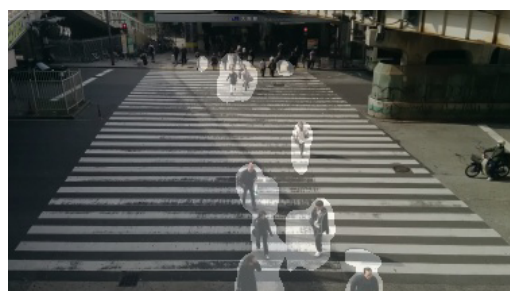


図 12 実環境における動体領域抽出例

り、提案手法により高精度な人数推定が実現できていることが分かった。

## 5. おわりに

本稿では、我々が開発している、スマートフォン俯瞰画像からの群衆人数推定システムについて紹介した。スマートフォンとクラウドサーバーがシームレスに連携動作するよう、同システムを実装中である。

### 参考文献

- [1] 宮崎真次, 宮野博義, 池田浩雄. 群衆行動解析技術を用いた混雑推定システム (社会の安全・安心を支えるパブリックソリューション特集)-(安全・安心な暮らし). NEC 技報, Vol. 67, No. 1, pp. 82-85, 2014.
- [2] Jonathan Reades, Francesco Calabrese, Andres Sevtsuk, and Carlo Ratti. Cellular census: Explorations in urban data collection. *IEEE Pervasive Computing*, Vol. 6, No. 3, pp. 30-38, 2007.
- [3] 小島颯平, 内山彰, 廣森聡仁, 山口弘純, 東野輝夫. 俯瞰画像における被写体サイズ推定モデルに基づく群衆人数推定手法の提案. 情報処理学会, マルチメディア, 分散, 協調とモバイル (DICOMO2015) シンポジウム論文集, 2015.
- [4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (CVPR '05)*, Vol. 1, pp. 886-893, 2005.
- [5] Unity-Game Engine. <http://japan.unity3d.com/>.
- [6] 大津展之. 濃度分布からの閾値決定法. 電子情報通信学会情報・システム部門全国大会講演論文集, 1977.
- [7] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. In *IJCAI*, Vol. 81, pp. 674-679, 1981.