ウェアラブルカメラを用いた道案内映像の自動作成

岡本 昌也 * [1,a)]　柳井 啓司[1,b)]

概要：
近年，ウェアラブルカメラと呼ばれる頭や胸などに着けて撮影するカメラの普及に伴って，一人称視点の映像が多く撮られるようになってきている。本研究では，一人称視点映像の移動映像を入力として，映像中の道順が分かるようシーンの重要度に応じて動的に再生スピードを変化させた要約映像を自動生成することを目的とする。要約映像を生成する為に，横断歩道検出と自己動作分類という 2 つの処理を行う。横断歩道検出は，映像中に出現する横断歩道を検出することで，映像中の交差点や分岐点を推定する。自己行動分類は，映像撮影者の行動を “前進”，“停止” 及び “右折”，“左折” の 4 つに分類する。要約映像の生成は，横断歩道検出と自己行動分類の結果を統合し，動的に再生速度を制御することによって行う。実験では，横断歩道検出の精度実験及び自己行動分類の精度実験を行い，その有効性を検証した。映像 3 本に対して提案手法の要約映像を生成して比較評価実験を行い，提案手法が単純な要約手法より優れている事を示した。

キーワード：ライフログ映像，一人称視点，動画要約

1. はじめに

1.1 背景

近年ウェアラブルカメラと呼ばれる，身に付けて常に撮影を行う為のカメラが普及してきている。ウェアラブルカメラで撮影された映像は，撮影者の視界とほぼ同等で一人称視点の映像と呼ばれ，一人称視点のライフログ映像が多く撮影される様になってきた。

一方で，道案内のシステムは様々なものが考えられているが，それらは，地図や文章などによるものが多い。そこで，一人称視点の映像から道案内用の映像を生成することが出来れば，より直感的かつ容易に道案内を行うことが出来ると考える。

1.2 目的

本研究では，特に歩行中にウェアラブルカメラで撮影した移動映像を対象とし，移動経路が分かるように要約することによって，道順案内の映像を自動生成することを目的とする。なお，本研究での映像要約では，実際には要約映像を生成するのではなく，再生時にビデオプレーヤーの再生速度をシーン認識の結果に応じて自動的にコントロールすることによって要約を実現する。そのため，要約映像閲覧時にユーザが希望する要約映像の長さを指定することが可能となっている。

1.3 対象とする映像

本研究で対象とする移動映像をより厳密に定義する。対象とする移動映像は次の条件を満たす。

- ある地点から別の地点まで移動する行程を撮影した映像
- 移動行程中をウェアラブルカメラで撮影したもの
- 撮りっ放しの映像で，途中切れたりしない

映像の最初は出発地点の映像であり，映像の最後は目的地点の映像である。このような映像を研究の対象とする。また，映像は移動開始地点から目的地まで連続的に撮影されたものであり，途中での撮影中断はなく，全行程を撮影しているものとする。ウェアラブルカメラは頭部もしくは胸部に前を向いて装着されることを想定する。移動手段は徒歩を想定し，自転車やバス，電車は対象としない。

1.4 求める要約の要件

まず，本研究の求める要約の要件を定義する。要約された移動映像を用いて，人に出発地から目的地までの道順を説明する事が本研究の目的である。よって，求める要約の要件とは，次の 3 つである。

- 出発地点と目的地点が分かる
- どのような道順であったか分かる
- 周りがどのような風景であったか分かる

本研究では，以上の観点から要約された映像を評価することとする。

*現在，東京大学大学院 情報理工学系研究科 創造情報学専攻
1 電気通信大学 情報工学科
a) okamoto-m@mm.inf.uec.ac.jp
b) yanai@cs.uec.ac.jp

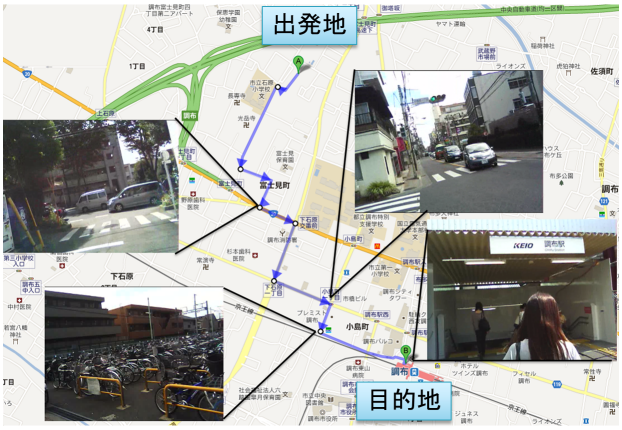


図 1　理想的な要約の例

例を示す。図 1 の地図上に青で示されたような，住宅地から出発して駅までの移動経路を考える。移動方法は徒歩である。曲がり角の部分や道路を横断する部分，出発地と目的地付近は上記の要件を満たす要約映像を生成する上で重要であるので，相対的に遅い速度で再生され，他の移動部分は重要ではないので，早送りで高速再生されるのが提案手法の要約結果となる。

要約結果の映像では，映像のスキップは行わず，重要でない部分は高速に再生を行う。これは，道案内映像は高速再生していてもまわりの風景はある程度把握できるため，スキップして場所が急に変わるよりも，高速再生の方が適していると考えられる為である。

2. 関連研究

2.1 一人称映像に関して

位置情報 (GPS 情報) を使用せずに，ライフログ 映像を要約する研究として Lee らの一人称視点のビデオを要約する手法の提案の論文 [1] がある。

この研究は，数時間におよぶライフログ映像の要約を映像情報からのみで行う手法の提案の研究である。対象とする映像は撮影者の視線とほぼ同等であると仮定して，一人称視点のビデオ特有の特徴量を用いている点が特徴である。

手法は，色情報によるイベント分割及び，物体認識で ライフログ 映像中の重要物体，重要人物を推定し，それに基づいて要約を行う。本研究とは違い，ライフログ映像全般を対象としている。この為，カメラの動き情報などは使用せず，物体検出や色情報などで要約を行っている。出力結果がストーリーボードと呼ばれる数枚から十数枚の画像である点でも本研究とは異なる。

要約の研究として，評価方法やデータ収集の方法などを参考にした。

2.2 画像情報以外の情報による要約

GPS 情報を使用してライフログ 映像を要約する研究として，Datchakorn らの体験の記録と ライフログ 映像に対するインデキシングの論文 [2] がある。

この研究は，GPS 情報やジャイロスコープ，加速度センサなどの客観的な指標となるコンテキストと，メールやドキュメントなどの主観的な指標となるコンテキストを記録するシステムの提案と，それらを検索できるような手法の提案が主な内容である。記録されたコンテキストデータを様々な視点からサンプリングし，撮影した映像からキーフレームを抜き出すことによって要約も可能にしている。

本研究との違いは，動画以外に GPS 情報などの他の情報を使用して要約を行う点である。要約の結果としては，複数枚のキーフレームである画像を出力すると共にそのシーンの動画を見れるようにするなど，インデキシングの側面もある点が本研究との違いである。

2.3 一人称視点映像における自己動作識別について

一人称視点映像での自己動作識別の研究には吉川らの ST-patch 特徴を用いた一人称視点映像に置ける自己動作識別の論文 [3] がある。

この研究は，人の視界を撮影したシーン映像と，眼球の動きをとらえた眼球映像を同時に撮影することが可能な Inside-Out カメラを頭部に装着し，屋内における歩行時の状況を撮影したシーン映像から前進，右左折，階段の上り下りなどの動作を識別する研究である。

アピアランスとモーション特徴を統合した ST-patch 特徴を使用し，マルチクラス識別の手法として Joint Boosting を使用している。シーンは屋内に限定されており，識別する動作は 6 種類である。

本研究の自己動作認識 (4.2 節) の部分に関係している。一人称視点における自己動作認識の研究で，自己動作識別手法や動作の種類について参考にした。

3. 処理の流れ

処理の大まかな流れは図 2 の様になる。

まず，一人称視点の移動映像を入力として，要約映像を生成する為に，大きく 2 つの処理に分かれる。1 つ目は映像中に出現する横断歩道を検出する横断歩道検出である。2 つ目は映像撮影者の行動を “直進”，“停止”，“右折”，“左折” の 4 つに分類する自己行動分類である。さらに，この 2 つの処理の結果を統合し，映像中のシーンごとに重要度を算出する。この重要度を基に再生シナリオを作成する。最後に再生シナリオに従って HTML5 ビデオプレイヤーを制御することで要約映像をユーザに提示する。

映像中に横断歩道が出現するということは，道の分岐点周辺である可能性が高く重要である。さらに，映像中で直進している部分よりも右折や左折を行っている部分は重要であるという仮定に基づいて重要度を算出する。

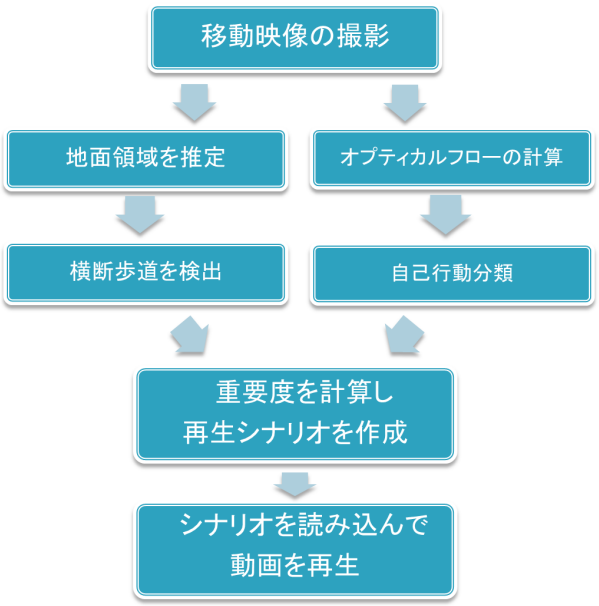


図 2 処理の流れ

3.1 横断歩道検出

要約したい映像 1 秒あたり 3 枚の画像を取り出し，それらの画像から横断歩道を検出する。前処理として，関係のない部分を除去するため，地面領域を推定する。地面領域の推定方法は，Hoiem らの手法 [4] の手法を利用する。認識手法は SVM を使用し，予め複数の映像から学習を行って使用する。

3.2 自己行動分類

一人称視点における自己動作分類を行う。つまり，移動映像を撮影している人自身が映像の中で何をしているか，を推定する。

具体的には，まず，要約したい映像からのオプティカルフローを計算する。さらに一定時間ごとに計算したオプティカルフローを合計し，それを利用して要約したい映像の撮影者の行動を推定する。

分類する行動は “前進”，“停止” 及び “右折”，“左折” の 4 つである。

認識手法は SVM 使用し，one-vs-all 方式で 4 つの認識機を作成して結果を統合して使用する。各認識機は複数の映像から学習を行い使用する。

3.3 重要度推定

横断歩道検出と自己行動分類の結果から，一定の時間間隔ごとに重要度を推定する。本研究では，映像 8 秒間を 1 単位として，映像の開始 8 秒ごとに映像を分割して重要度を算出する。また，分割された 8 秒間の映像をシーンと呼ぶことにする。シーンは短い方がより細かく動画を操作できるが，再生時に動的に再生速度を操作するオーバーヘッドが増えるため 8 秒とした。

重要度が高い部分が最終的な要約映像に残り，重要度が低い部分は要約では早送りされる。

3.4 再生シナリオの生成とプレイヤー制御

推定した重要度から再生速度の算出と，ビデオプレイヤーを制御する再生シナリオを生成する。再生シナリオとは，どの部分をどのくらいの速度で再生するか，という情報である。

最後に，生成された再生シナリオに基づいて HTML5 ビデオプレイヤーを操作して，要約映像をユーザに提示する。

4. 提案手法の詳細

4.1 横断歩道検出

映像中に出現する横断歩道を検出する手法を説明する。

実験で使用した映像は 24fps であるが，この全ての画像に対して横断歩道検出の処理を行う事は計算コストが掛かる。さらに，画像も短時間では大きく変化しないため，より少ない枚数で行っても問題にならないと考える。

このため，映像から 1 秒に 3 枚画像を抽出し，それらに対して以下の処理を行うことにする。

4.1.1 地面領域の推定

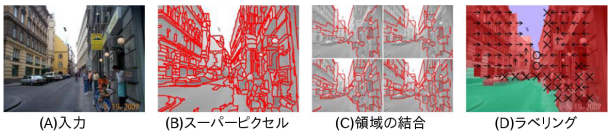


図 3 Geometric Context のアルゴリズム ([4] から引用)

Hoiem らの Geometric Context の研究 [4] の手法を使用して，地面領域を推定する。

この研究は，1 枚の画像から 3 次元的な情報を復元し，領域分割及び空や地面，壁などのラベリングを行う研究である (図 3)。研究の手法について本研究で使用した地面領域の推定の部分を中心に詳しく説明を行う。

先ず初めに，画像をスーパーピクセルに分割する。スーパーピクセルとは類似した特徴をもつ画素の集まりであり，RGB の輝度を利用した過分割手法を用いて得られる (図 3(B))。

次に，スーパーピクセルどうしの結合を行う。これにはスーパーピクセルどうしの結合のしやすさの確率を使用する。この確率は位置情報や色，テクスチャなどの特徴を使用してブースト決定木に学習させておき，計算される。確率のしきい値を設定する又は，マルコフ確率場による決定など様々な方法がある (図 3(C))。本研究では、確率のしきい値による方法を使用している。

最後に，結合された領域それぞれについて，クラス “sky”，“support”，“vertical” のどれに属するかを決定する。これは，それぞれのクラスに対応した one-vs-all 方式のブース

ト決定木を用意することで実現される。領域の特徴量はスーパーピクセルの場合と同様のものを使用し，結合された領域ごとに新たに計算を行う。

各クラス “sky”，“support”，“vertical” は，それぞれ空，地面，垂直な面に対応している。図 3(D) が最終的な出力結果である。図中青，緑，赤がそれぞれ “sky”，“support”，“vertical” に対応している。

本研究では，主要クラスの “support” のブースト決定木の出力結果を使用する。これにより，各ピクセルごとに地面領域の確率が得られる。

この確率がしきい値以上の領域を画像中の画像の領域とみなす。元論文では，しきい値を 0.5 としているが，本研究では地面領域の精度よりも地面領域が漏れなく取得される事を重視し，しきい値は 0.4 とする。

4.1.2 膨張処理

[A] 入力画像　[B] 地面領域　[C] 処理後

図 4 地面領域推定の例

推定した地面領域は，精度の問題から幾つかの領域に分かれてしまったり，地面領域中に非地面領域がある中抜けのような現象がよく発生する。

実際の実行例を次の図 4 に示す。[A] が入力画像である。[B] に緑で示されている領域が 4.1.1 節で求められた地面領域である。これにモルフォルジー演算の膨張を行うと [C] のようになる。中抜けの部分がなくなっている事が確認できる。

この処理を行った後の領域が最終的に推定された地面領域である。

4.1.3 画像特徴抽出と BoF 化

各画像について対応する表現子を作成する。具体的の画像の手法の説明を行う。

先ず，画像全体からキーポイントを検出した後，地面領域外のキーポイントを削除する。そして，残った各点に対応する Scale-Invariand Feature Transform(SIFT) 特徴を計算する。これが画像に対する画像特徴である。

さらに，予め各ビデオについて 5 秒に 1 枚取り出した画像群から，上と同様の方法で取り出した画像特徴を用いてコードブックを作成しておく。

最後に，このコードブックを使用して，各画像に対応する SIFT 特徴を bag-of-features(BoF) ベクトルに変換する。この BoF ベクトルを 1 つの画像の表現子とする。

4.1.4 SVM による 2 値分類

画像に横断歩道が写っているかどうかの認識は SVM を利用する。

予め SVM のを正例（横断歩道が写っている）約 80 枚，負例（横断歩道が写ってない）約 160 枚で学習しておく。学習した SVM を利用して 2 値分類を行う。

本研究では，プログラムとして LIBSVM[*1] を用いた。また，カーネルは RBF カーネルを使用する。

4.1.5 結果の統合

画像に横断歩道があるかどうかは，先の SVM の出力結果を使用する。しかし，必要なのは映像中いつ横断歩道が出現するか，である。

映像中から 1 秒に 3 枚の頻度で画像を取り出し，それぞれに認識の結果を統合して最終的な重要度の推定を行う。詳しい説明は 4.3 節で行う。

4.2 自己行動分類

映像を撮影している人（移動者）の行動を要約の為分類する。移動映像をさらに短い時間の動画に分割して自己行動分類を行う。映像を 4 秒ごとに区切りその時間における自己行動の分類を行う。ここでは，その手法について説明を行う。

4.2.1 分類する行動

本研究では，移動者の行動を次の 4 つに分類する。

- 前進
- 停止
- 右折
- 左折

移動映像であるので，後退などは含まれていないものとする。これらの行動に分類出来ない行動，例えば，右を見る動作などはこれら 4 行動中最も動作が近しいものに分類する。

4 秒ごとに映像をこれらの行動に分類する。4 秒中に複数の行動が含まれるような場合は，そのうち最も時間が長いものに分類されるのが理想とする。

4.2.2 オプティカルフロー計算

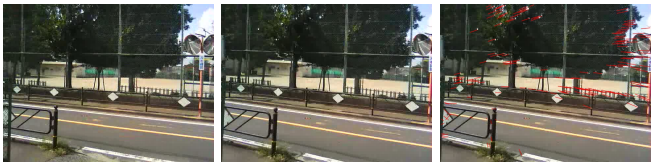
動作を分類するために映像からオプティカルフローを計算する。1 秒間に 12 枚の頻度で画像を取り出し，それぞれ前後の画像からオプティカルフローを取る。

オプティカルフロー計算する手法を説明する。まず，オプティカルフローを計算するのに適した特徴的な点を検出する。これは，1 番目の入力画像に対して行われる。画像中で大きな固有値を持つコーナー（はっきりとした特徴点）を見つける。

次に，先の処理で検出された疎な特徴点に関して，オプティカルフローを計算する。計算には，Lucas-Kanade アルゴリズム [5] に基づいたオプティカルフローを利用する。

オプティカルフロー計算の例を図 5 に示す。[A] が 1 番

[*1] http://www.csie.ntu.edu.tw/∼cjlin/libsvm/



[A] 入力画像 1　[B] 入力画像 2　[C] オプティカルフロー

図 5　オプティカルフロー計算の例

目の入力画像であり，[B] が 2 番目の入力画像である。[C] の画像中赤色の線が実際に計算されたオプティカルフローである。

4.2.3　ベクトル化

本論文では，自己動作認識の 1 動作単位は 4 秒とし，開始時間は 2 秒ごととした。4.2.2 節で計算したオプティカルフローを 4 秒分で合計し，4 秒間の映像の表現子を作成する。具体的な手法を説明する。

オプティカルフローは画像上の二次元ベクトルの集合であるので，それらを 1 つのベクトルで表現する必要がある。そこで，画像を縦横 4 × 4 に分割し，オプティカルフローを出発点（1 番目の画像の特徴点の座標）によって大きく 16 の領域に分ける。さらに，各領域内でベクトルの方向 20 ° ごとに 18 個に分け，計 288 個のビンに分ける。同じビンに入ったオプティカルフローの大きさの平均をとり，288 次元のベクトルとする。

次に，4 秒間の画像からオプティカルフローを各々計算し，それらの平均を取る。さらに 288 次元の要素の合計が 1 となるように正規化を行う。これを映像 4 秒間の表現子とする。

4.2.4　SVM による行動分類

横断歩道検出の認識同様に，自己行動分類でも認識手法に SVM を使用する。予め学習した SVM を作成し，それらを使用して認識を行う。本研究では，横断歩道検出と同様 LIBSVM を使用した。また，カーネルは RBF カーネルを使用する。

one-vs-all で SVM を 4 つ作成し，それぞれの結果を統合して分類を行う。出力結果は，2 値分類の結果でなく，LIBSVM による確率の出力値も使用する。

4.3　重要度推定

これ以前は要約映像に対して予め行う必要のある処理であるが，これ以後の重要度推定からビデオプレイヤーの制御までは JavaScript 上で実行される。8 秒ごとに自己行動分類の LIBSVM による確率の出力値 4 つと横断歩道検出数を JavaScript に埋め込むことで実装する。LIBSVM による確率の出力値は 4 秒ごとであるが，8 秒ごとに平均をとり埋め込む。

要約映像を見る時点で処理を行う為，各種パラメータをその場で変更できる。つまり，要約映像（再生シナリオ）をニーズに合わせてその場で作り直すことが可能である。

要約シナリオを作成する為に重要度を算出する。重要度を算出する単位は映像の 8 秒間とする。つまり，映像 8 秒ごとに重要度を算出し，再生シナリオも 8 秒を単位として生成される。

重要度とは 0 から 1.0 の実数で，値が高いほどその映像の部分が重要であることを示す。この値は，4.4 節で示す様に速度の算出にも使われる。

以下で重要度算出方法の説明を行う。

4.3.1　自己行動分類による重要度推定

自己行動分類の 4 つの識別器の LIBSVM による確率の出力値から重要度を算出する。具体的には，8 秒間の映像に対応するの 4 つの LIBSVM による確率の出力値を重みを付けて合計する。

i 番目シーンの直進の LIBSVM の確率を $v_f[i]$，同様に停止を $v_s[i]$，右折を $v_r[i]$，左折を $v_l[i]$ とそれぞれすると，i 番目の重要度 $p[i]$ は

$$p[i] = c_f v_f[i] + c_s v_s[i] + c_r v_r[i] + c_l v_l[i]; \qquad (1)$$

で算出される。

計算で使用される定数 c_f, c_s, c_r, c_l は，実際の認識結果から次の様に定めた。

- c_f = -2
- c_s = 1
- c_r = 2
- c_l = 2

映像の撮影者が右折や左折を行う部分というのは，移動の道順を知るという意味で重要であると考えられる。従って，c_r 及び c_l は正の値である。逆に，直進中は重要でない事が多いので，係数 c_f は負の値である。次に，停止は重要である場合とそうでない場合が考えられる。例を示せば，混雑等の理由によって道の途中で止まったシーンは重要ではないが，交差点などで信号待ちをしているシーンはある程度重要である。つまり，停止の動作は他の動作に比べて重要度，非重要の分類力が低いと言える。従って，係数 c_s は絶対値を 1 とし，それ以外は絶対値を 2 としている。

さらに，正規化を行い重要度を 0 から 1.0 に直す。具体的には，移動映像中に出現する最大の重要度を p_{max}，最小値を p_{min} とすると，正規化後の重要度は

$$p'[i] = \frac{p[i] - p_{min}}{p_{max} - p_{min}} \qquad (2)$$

で計算される。

4.3.2　横断歩道検出による重要度推定

横断歩道が現れる映像の部分は，十字路などの道の分岐点である可能性が高い。つまり，横断歩道が検出された映像の部分は，道順が分かるという要件について重要であるといえる。

横断歩道検出は誤検出などもあるので，誤検出によるノイズ除去の為の閾値を設定した。8 秒間 24 枚の画像に閾値以上横断歩道が検出された場合，真にその 8 秒間に横断歩道が出現したと見なし，その部分の重要度を 1 を上限に 0.5 上げる。つまり，横断歩道が検出された i 番目の重要度は

$$p''[i] = min(p'[i] + 0.5, 1.0) \quad (3)$$

で計算される。

本研究では，実際の検出結果から閾値は 7 とする。

4.3.3 出発地と到着地の重要度

移動映像の出発地と到着地は，道案内映像においては最も重要である。

出発地周辺は要約される映像の最初の部分が，到着地周辺は最後の部分がそれぞれ対応している。この要件を満たすため，最初と最後の 2 シーン分（約 16 秒）の部分は盲目的に重要度を 1.0 に設定し，早送りされないようにする。

4.4 再生速度計算

重要度に基づいて映像の再生速度を決定する。i 番目の重要度を $p''[i]$ とすると，速度 $s[i]$ は

$$s[i] = \frac{1}{p''[i](1 - (1/(s_{max} - 1))) + (1/(s_{max} - 1))} + 1 \quad (4)$$

s_{max} は再生シナリオで許される最大の再生速度である。再生速度が速くなりすぎると要約動画を再生する負荷が増大するので，定数として s_{max} を与えて再生速度を制御する。s_{max} は動画再生時にユーザが任意に変更可能である。5.4 節の評価実験の際は s_{max} は 7 とした。

4.5 速度の平滑化

再生速度が短い時間で大きく変わる場合，映像が見づらくなることがある。これを軽減する為に，再生速度に対して平滑化を行う。手法は重み付けした平均をとる単純なものである。

具体的には，i 番の速度 $s[i]$ は

$$s'[i] = 0.1(s[i-1] + s[i+1]) + 0.8s[i] \quad (5)$$

これは最初と最後の 2 シーン分には適用しない。

5. 実験

5.1 データセットの収集

データセットを収集するため，映像の撮影には図 6 に示す，ウェアラブルカメラ Looxcie2 を図 7 の様に耳につけて撮影を行った。

収集したデータは，次の表 1 になる。

5.2 横断歩道検出の精度実験

まず最初に本論文で提案した横断歩道検出についての評価を行う。

図 6 Looxcie2

図 7 装着図

表 1 収集したデータセット

移動手段	本数	平均時間
徒歩	9	9:12

5.2.1 実験方法

本論文で提案した手法で実際に横断歩道検出器を作成して精度評価を行う。

4 つの映像から取り出した約 250 枚の画像で学習を行い，テスト用の画像は，5 つ映像から抽出した 200 枚（ポジティブ画像 100 枚，ネガティブ画像 100 枚）を使用する。テスト用の画像を取り出した映像は学習用の映像には含まれていない。

また，画像中の地面領域を推定する方法の有効性を検証するため，地面領域推定を行わずに，画像全体から SIFT 特徴を取り出す手法と比較を行う。この手法はコードブック及び BoF 作成時にも地面領域に対するフィルタリングを行っていない。

5.2.2 実験結果

結果は次の表 2 のようになった。

表 2 横断歩道検出の精度実験結果

手法	再現率	適合率	F 値
地面領域推定あり	0.37	0.787	0.503
地面領域推定なし	0.26	0.839	0.397

5.3 自己行動分類の精度実験

次に，本論文で提案した自己行動分類についての評価を行う。

5.3.1 実験方法

本論文で提案した手法で識別器を作成し，実際の映像から抽出したテスト用のデータを使用して評価を行う。ここでは，4 つの分類器に対して二値分類の実験をそれぞれ独立して行う。提案手法では，それぞれの分類器の確率を使用するため，それぞれの分類器の精度の評価を行う。

4 つの識別器に使用する学習データは，4 つの映像から抽出する。映像中に含まれる動作数は各動作ごとに異なり，収集した映像には直進が最も多く，80%以上が直進である。4 つの映像から学習用のデータを抽出し，学習を行う。停止，右折，左折に関しては，ポジティブの数が少な

い為，ネガティブ数はポジティブ数の 2 倍程度になるように均等な間隔で取り出す。直進だけはネガティブの方が少ない為，ポジティブをネガティブの 2 倍程度に抑えられるよう同様に均等な間隔で取り出した。実際に使用した学習データの数は表 3 の様になった。

表 3 学習のデータセット

行動	ポジティブ数	ネガティブ数	学習データ数
直進	512	216	728
停止	74	146	220
右折	70	147	217
左折	68	131	119

5.3.2 実験結果

テスト用のデータは 3 つの映像から抽出し，ポジティブ数とネガティブ数を同数としてテストを行う。データ数の半数がポジティブでもう半数がネガティブである。学習データ同様，ポジティブ数とネガティブ数のうち少ない方を全て使ってテストを行う。多い方は一定間隔で抽出し使用した。

実際のテストデータ数と結果は次の表 4 の様になった。

表 4 自己行動分類の結果

行動	データ数	再現率	適合率	F 値
直進	244	0.943	0.697	0.801
停止	72	0.694	0.893	0.781
右折	84	0.738	0.969	0.838
左折	88	0.795	0.972	0.875

4 つの分類における F 値の平均は 0.824 である。

5.4 要約映像のユーザ評価実験

最後に，本論文で提案した要約手法についての評価を行う。

5.4.1 実験方法

本論文で提案した手法で要約映像を作成し比較実験を行う。

要約手法は次の 3 つである。

(1) 提案手法
(2) 自己動作分類のみ
(3) 均等に早送り

2 は提案手法から横断歩道検出の部分を無効にしたもの，3 は比較用に加えてある。これらの映像を 1.4 節の基準に従って評価を行い，最も良かった要約に投票してもらう。

3 本の映像に対して実験を行った。同じ条件で比較を行うために，各映像の長さを均一にする。このとき，2 の手法の長さを基本とする。3 は映像の長さから平均の速度を求め，シナリオを与えて再生する。1 については，最終的に計算される速度全てに，映像の長さが等しくなるように定数を掛けて調節を行った。

使用した映像の詳細は次の表 5 にまとめた様になる。

表 5 評価実験のデータセット

動画	映像の長さ	要約後の長さ	平均速度
徒歩 A	7:47	1:45	4.5
徒歩 B	9:17	2:20	3.9
徒歩 C	11:26	2:40	4.3

また，映像中，撮影者の歩くときの頭の揺れが映像を見にくくさせている。映像を早送りする場合はその影響が顕著である。なので，映像に対してスタビライズを行い，それを再生用の映像としている。

5.4.2 実験結果

比較実験の結果，11 人から 25 の投票を得た。結果をまとめると次の表 6 の様になった。

表 6 評価実験の結果

動画	均等に早送り	自己行動分類のみ	提案手法
徒歩 1	0	4	6
徒歩 2	0	4	3
徒歩 3	0	1	7
合計	0	9	16

比較用の均等に早送りには，いずれの映像でも一票も入っていない。

6. 考察

6.1 横断歩道検出の精度実験

横断歩道検出の結果を見ると，地面領域の推定及び特徴点の選択処理を行った方が F 値が 0.106 ポイント上昇しており，横断歩道検出を行う際の前処理として地面領域検出を行う有効性が実証できた。

また，地面領域の推定が失敗すると，横断歩道検出も失敗する傾向にある。その場合，地面領域が小さくなり，十分に特徴点が取れずに失敗すると言う傾向があることが分かった。

地面領域の推定ありの場合のみ成功する検出の例は図 8 である。図中推定された地面領域を緑色にしている。地面領域の推定がないと画像上半分からの特徴点がノイズになり，地面領域の推定がある場合でのみ検出に成功した。地面領域の推定ありの場合のみ失敗する検出の例は図 9 である。地面領域の推定の失敗によって，十分に特徴が出ず失敗した例である。画像全体に横断歩道が映っているので地面領域推定なしでも検出に成功している。

6.2 自己動作分類の精度実験

自己行動分類の結果を見ると 4 つの手法での F 値の平均は 0.824 であり精度よく分類出来ていると思われる。しかし，根本的な問題として右折と直進の境目は何処かという難しい問題もあり，単純には評価できない。

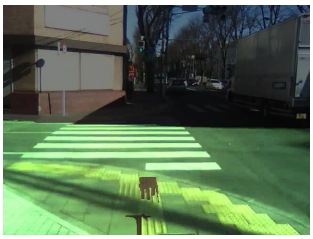



図 8 地面領域推定ありの成功例　　**図 9** 失敗例

代表的な失敗の例は次の図 10 の様なシーンである。横断歩道前で信号待ちをしているシーンで，停止が正しい。しかし，目の前を車が通過するため，オプティカルフローが車に出てしまい，右折と分類される。この様に自身の動きと関係なくオプティカルフローが出てしまう人や車などが多く出現する部分は分類の精度が落ちる事が確認できた。

図 10 自己行動分類の代表的な失敗

6.3 要約映像のユーザ評価実験

ユーザ評価実験の結果，合計では提案手法が最も優れた要約手法である事が分かった。しかし，3 本の動画中 1 本では自己動作分類のみの要約の方が良いという結果であり，自己動作分類のみでも十分要約出来ていると見るべきだろう。

また，実験者からは提案手法の要約と自己動作分類の違いが分からない，という意見もあった。これは，道路を渡る場面など横断歩道が登場して且つ重要な場面はテスト映像で出現するが，その殆どで信号待ちをしている。すると停止に近い状態と推定されて重要度が高くなり，結果的に横断歩道検出が無くても重要度が高くなる。この場合，提案手法と自己動作分類のみの違いを見出す事が難しく，横断歩道の誤検出という問題がある提案手法に不利に働いたと考えられる。また，小さな横断歩道を直進で通過するなどの全体的に見れば，重要でない部分も重要度を引き上げてしまうので見づらいと感じる人が多かったのではないかと考えられる。

7. おわりに

7.1 まとめ

本研究は，一人称視点の移動映像に対して，映像情報のみから要約映像を自動生成する手法を提案した。結果として，単純な要約手法より提案手法の方が優れていることを示した。また，横断歩道検出においては，F 値は 0.503 であり，地面領域を推定して特徴点を削除することを行わなかった場合より 0.106 ポイント向上した。自己動作分類の 4 クラス動作分類の F 値の平均は 0.824 であった。提案手法の要約手法の有用性が実証できた。特に，自己行動分類におけるオプティカルフローを用いた標準的な方法を確立できた。

7.2 今後の課題

今後の課題としては，移動方法を徒歩に限定せず，自転車や自動車など他の移動手段でも有効であるかの詳細な検討と，大規模な実験が上げられる。現在は，大学周辺の主に住宅地で映像を撮影しているが，他の地域以外や住宅地以外の映像データセットを充実させていく必要があるだろう。

横断歩道以外の重要物体を登録もしくは，ユーザデータベースから推定して要約に使用するなど，屋外だけでなく屋内でも使用できる要約手法にしていく予定である。また，現在は手動で設定している要約に使用する各定数も学習によって最適なものを自動で選択する様な手法を検討していく予定である。

参考文献

[1] Y. J. Lee, J. Ghosh, and K. Grauman. Discovering important people and objects for egocentric video summarization. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2012.

[2] Tancharoen. D, Yamasaki. T, and Aizawa. K. Practical experience recording and indexing of life log video. In *Proc. of ACM SIGMM Workshop on Continuous Archival and Retrieval of Personal Experiences*, 2005.

[3] 古川拓弥, 藤吉弘亘. First person vision のための st-patch 特徴を用いた自己動作識別. 信学技報, 第 110 巻 of *PRMU2010-11*, pp. 53–58, 5 月 2010.

[4] Hoiem. D, Efros. A, and Hebert. M. Recovering surface layout from an image. *International journal of Computer Vision*, 2006.

[5] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. International Joint Conference on Artificial Intelligence*, pp. 674–679, 1981.