映像検索手法を利用した車載ビデオ映像の位置同定手法

福元和真 *1　川崎洋 *1　小野晋太郎 *2　子安大士 *3　前川仁 *3　池内克史 *2
鹿児島大学 工学部情報生体システム工学科　〒 890-8580 鹿児島県鹿児島市郡元 1-21-24*1
東京大学 生産技術研究所　〒 153-8505 東京都目黒区駒場 4-6-1*2
埼玉大学 理工学研究科　〒 338-8570 埼玉県さいたま市桜区下大久保 255*3

近年，車載カメラの普及に伴い動画サイトへの車載動画の投稿が普及し始めている．これらの映像を利用し，日常的に自由走行しながら撮影された様々なカメラ映像をつなぎ合わせることで，より広範囲にわたる地域で高い頻度での更新が実現できるが，位置情報の無い映像をプローブカーなどで得られた情報つき映像と対応付ける必要がある．そこで本手法では，まず時系列画像をから建物の相対的な高さ情報を保持する Temporal Height Image (THI) を作成し，それに対して Affine SIFT によりロバストな特徴を取り出す．得られた特徴を Bag of Feature で表現し，効率のよいマッチングを実現する．実際に市街を撮影したデータおよび動画投稿サイトの映像を使用し検証した．

On-Vehicle Video Localization Technique based on Video Search with Geometric Information

Kazuma Fukumoto*1　Hiroshi Kawasaki*1　Shintaro Ono*2　Hiroshi Koyasu*3　Hitoshi Maekawa*3
Katsushi Ikeuchi*2
Kagoshima University*1 Tokyo University*2 Saitama University*3

Abstract Recently, with the spread of on-vehicle video, it became common to share the on-vehicle video on website. These videos can be used for frequently and widely updating of virtual city space such as Google Street View, if the location of the videos are available. To estimate the location, it is required to match the uploaded on-vehicle video to the video with location taken by a probe car. In this paper, we propose the efficient matching method using Temporal Height Image (THI), Affine SIFT and Bag of Feature. THI retains information of relative building height from temporal image sequence, and extract robust features by Affine SIFT. We realize efficient matching by expressing their features using Bag of features. We conducted experiments to show the efficiency of the proposed method by real image sequences of the city.

Keyword: *Temporal Height Image, Bag of Feature, Affine SIFT*

1. 背景

近年，Google Earth や Google ストリートビューのように街をコンピュータ上で表現する研究が数多く行われており，景観シミュレーションや防災の分野とも相まってこれからも発展していくことが期待される．しかし，これらの手法における撮影は専用のプローブカーを使用したものであり，一般的とは言い難い．そのため，頻繁な更新は都市部に限られている．

街を撮影する手段として車載カメラがある．車載カメラは，駐車時や後退時の運転支援，事故記録などの観点から注目を集めその搭載が進んでいる．その普及に伴い，一般ドライバーによる動画投稿サイトへの車載動画の投稿が増加している．これらの映像を利用し，日常的に自由走行しながら撮影された様々なカメラ映像をつなぎ合わせることで高い頻度での更新が実現できる．しかし，そのようなデータすべてに GPS などの情報が付与されているとは限らず，どこで撮影されたデータかを特定する手段は未だ確立されていない．

そこで，本研究では，都市空間の効率的かつ頻繁な更新を行うための複数の車載映像同士のマッチング手法とする．提案手法では，まず時系列画像から建物の相対的な高さ情報の画像である時系列高さ画像 (Temporal Height Image, THI) を作成し，Affine SIFT によりロバストな特徴抽出をする．得られた特徴を Bag of Feature (BoF) 表現を行うことで撮影環境に対してロバストな映像同士の対応付けを行う．これにより，プローブカーで撮影した位置情報つきの映像に，一般車載カメラで撮影された位置情報なしの映像を対応付けることが可能になる．

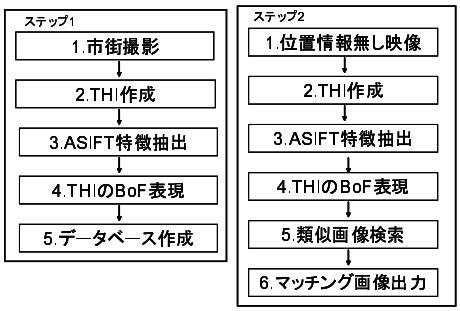


Fig.1 提案手法

2. 関連研究

車載映像を用いた研究は数多く行われている．Google ストリートビュー [1] では，全方位カメラを車または人，自転車に搭載し街をパノラマ撮影し，つなぎ合わせることで街中を歩いているような映像を提供している．また，Google Earth や Microsoft の Virtual Earth では，上空より撮影した航空写真を組み合わせることで，街を三次元で表現する研究を行っている．しかし，これらの手法では，特殊な機材または撮影手段を利用しているため頻繁な更新は都市部に限られている．

車載カメラで撮影した他の研究では，車載カメラを用いた自車の自己位置推定の研究がある [10]．この手法では，位置情報や距離データを付与した距離データマップを作成しておき，自車の走行時の距離データ系列と距離データマップを DP マッチングさせることで自車の位置推定を行っている．しかし，自車の距離データをレーザーレーダで取得するため，一般車による利用は困難という問題点がある．

三浦らは，ロボットから得た画像を Support Vector Machine (SVM) により建物などの領域に分割し，領域のマッチングにより季節や時間帯にロバストな画像マッチングを実現し，それによるロボットの位置推定を提案している [5]．しかし，彼らの手法は一枚の画像によるマッチングである．より高いロバスト性を得るためには，時系列でのマッチングが必要である．

時系列画像を利用した例として小野らは，複数の車載カメラ映像を用いた広範囲の 3 次元都市構築手法を提案している [9]．都市の特徴抽出方法として，エピポーラ平面画像 （Epipolar Plane Image，EPI） と前述の THI を用いる事で，撮影条件の差異を考慮した特徴抽出を行い，得られた画像列を連続 DP マッチングにより対応付けている．

時空間特徴量としては上記のほかに，Space-Time Patch (ST-Patch) や CHLAC，T-junction[4] 等が知られている．本手法では THI と背景と前景が交差する点に発生する T-junction を用いることでロバストな映像マッチングを目指す．

3. 手法概要

本手法の具体的な手順を図 1 に示す．

本論文では，すでに述べたようにプローブカーによる位置情報つきの映像シーケンスと異なる環境で撮られた一般車載カメラによる位置情報のない映像シーケンスの自車位置推定手法について述べる．

異なる撮影環境で撮影された映像を用いる場合，撮影車毎の速度の違いや，撮影するカメラの設置方向の違いのため生じる，建物の見えの違いがある．さらに，市街を走行して撮影を行う場合，道路の凹凸によりカメラが揺れることも考えられる．これらは，ステップ 1-3 の Affine SIFT を用いることで近似することが可能である．

また，ステップ 1-4 の BoF では，局所領域における特徴量をベクトル量子化して扱うため，画像を局所的な特徴の集合として扱うことが可能となる．つまり，特徴量の出現位置を考慮しないため，車載映像の撮影開始，終わりの位置に因らない画像表現が可能となる．

ステップ 2 では，位置情報を持たない画像シーケンスを入力画像として与え，BoF 表現する．これをデータベースの Visual words から検索することで，類似画像シーケンスを見つける．

4. 時系列高さ画像を用いた特徴表現

4.1 時系列高さ画像

- THI 作成

本手法では，まず，得られた全方位画像を一般的なカメラ画像である透視投影画像に変換する．透視投影画像 1 フレームからマッチングにより対応画像を見つけようとしたとき，天気や撮影時間といった撮影環境が異なる場合，建物の色情報が異なってしまう．そのため得られる特徴が異なり，対応画像を見つけることは困難である．

しかし，建物をシルエット画像で表す THI を用いることで上記の撮影環境の変化は克服できる．THI の作成方法を図 2 に示す．THI は，建物とみなされたエッジを画像底辺からの距離に比例した輝度値で表し，時系列順に並べることで得られる．

THI は建物の輪郭情報に依存した画像表現のため，季節や撮影時間による日照条件などの建物の見えの違いの影響を受けない．また，THI 作成において，街灯のような建物の前景にあり，細くて十数フレームしか現れないオブジェクトは，THI を作成した際に細い線となって現れる．これが建物と交わる点を T-junction と呼び，重要な特徴量となる．

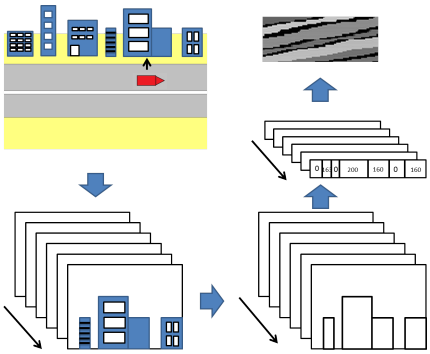


Fig.2 THI の作成方法例

- THI のリファイン

街の高さ情報取得の妨げとして電線がある．本手法では，王らの市街地のシーンに適したメディアンフィルタを用いることで電線除去を行った [7]．これは，メディアンフィルタを $1 \times n$ の縦一次元とし，明るめの画素値で置き換える処理であり効率よく除去できる．

メディアンフィルタによる処理では，電線のように細いノイズは除去できるが雲のように大きな物体の除去は出来ない．そこで，メディアンフィルタにより処理された透視投影画像に対して，本手法ではグラフカット [8) 3)] を用いることで，空を一様にラベリングし，雲による影響を抑えた．しかし，グラフカットにより街灯のような T-junction となりうる物体が失われる問題が生じる．そこでグラフカット後に，さらにクロージング処理を施すことで切れた部分の復元を行った．

これらを図 3 に示す．入力画像 (a) に対し，(b) のようなエッジ画像をグラフカットとクロージング処理を使ったものを (c) と (d) に示す．また，処理前と処理後の THI を図 4 に示す．

4.2 Affine SIFT と BoF による時空間特徴表現

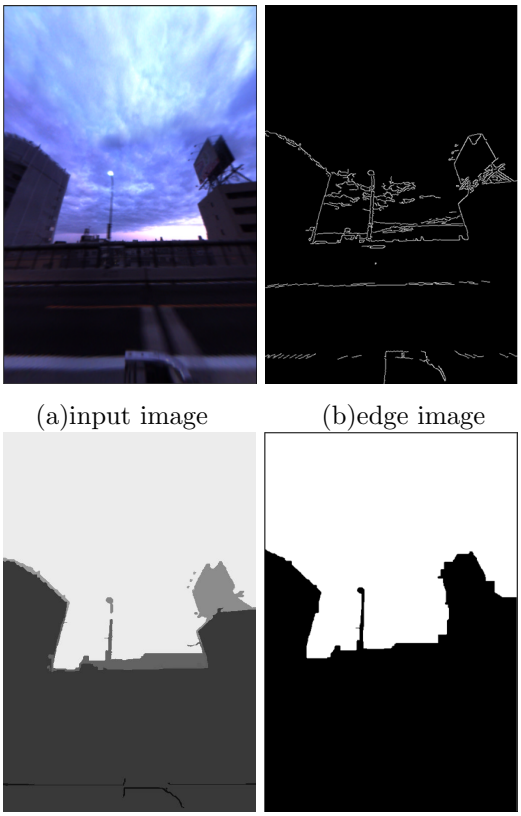
- Affine SIFT による特徴抽出について

THI を作成する場合，撮影車の速度は THI 上での傾きに影響し，対象物体までの距離は輝度とスケールと傾きに影響する．

これらの影響は，照明変動とアフィン変換で近似できると考えられる．本手法は，これらの変動を考慮した特長量である Affine SIFT[6] を用いる．Affine SIFT はアフィン変換を用いた特徴抽出方法であり，市街撮影のように撮影車の速度の違いやカメラの撮影角度の違いによる THI の変形に対しロバストな特徴抽出を行うことが出来，高い再現性が実現できる．

図 5 に，Affine SIFT により特徴抽出した 3 枚の THI および対応する実シーンを示す．各 THI 上の点は特徴点の位置を表す．抽出された特徴点を k-means アルゴリズムによりクラスタリングした．各画像で同じ色の点は同じクラスタに含まれる特徴点を表す．図 5 (d) は図 5 (a) のシーンの 1 フレームである．

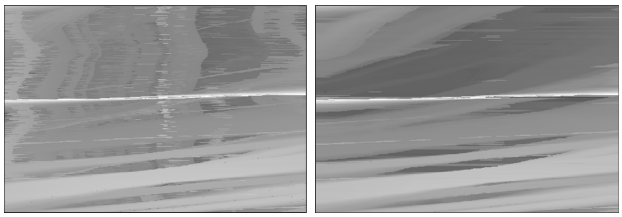
Affine SIFT による特徴点は，色の変化が大きい部分



(a)input image (b)edge image

(c)graph cuts (d)closing

Fig.3 雲除去処理による変化



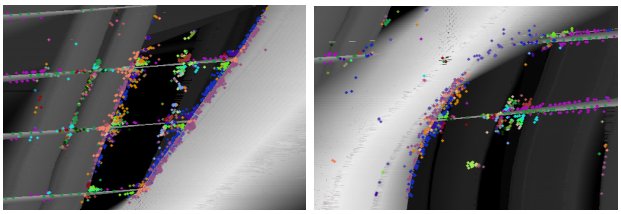
(a)before (b)after

Fig.4 雲除去の有無による THI の比較例

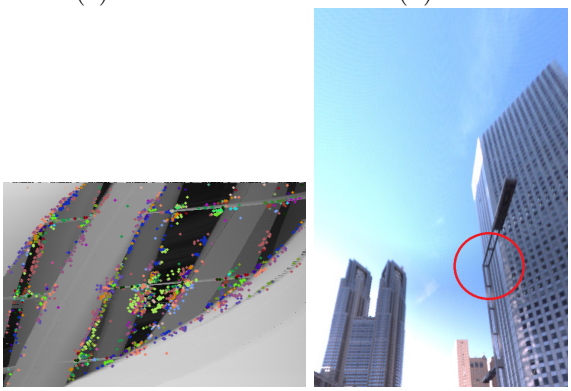
や細い線の周辺に多く出現することが分かる．これを元のシーンで見た場合，色の変化の大きい部分は建物と空や隣の建物との境界である．また，細い線は街灯などの出現により数十フレームだけ発生する．これが建物と交わる部分が T-junction である．

大きな建物の側面に発生する特徴点は同じ色であることから，同じクラスタに分類されていることが分かる．また，T-junction 周辺の特徴もそれぞれの画像で同じ色の特徴に分類されている．つまり，大きな建物だけでなく，T-junction も重要な構成要素となっている．さらに，それぞれの THI はお互い似た要素で構成されており，これを利用する事で，後述する BoF 表現が可能となっている．

- BoF による時空間特徴抽出表現



(a)THI1　　(b)THI2



(c)THI3　　(d) シーケンス内画像

Fig.5 Affine SIFT 特徴量のクラスタリング例

本手法では，対応画像検索に web 画像検索で定評がある Bag of Feature (BoF) [2] を用いる．BoF は，すべての画像を visual words という代表的な特徴量によるヒストグラムで表すものである．本手法では，データベースにある大量の画像群の中から対応する THI を見つけ，それを対応する撮影位置とすることを目標としている．

本手法は，異なる撮影環境で撮られた映像のマッチングを行うので，visual words は，撮影環境に対してロバストなものでなければならない．本手法では，visual words として前述した Affine SIFT を用いて得られた，撮影環境の違いに対してロバストな時空間特徴量を用いることで対応する．BoF は，局所領域における特徴量をベクトル量子化して扱うため，画像を局所的な特徴の集合として扱っている．そのため，特徴の出現位置によらない対応付けが可能となる．

BoF は，まず得られた局所特徴量をクラスタリングする．クラスタリングの手法は多々あるが，本手法では，階層的 k-means アルゴリズムを用いた．すべての特徴量がクラスタリングされた後，得られたクラスタの中から重心を選び，それをクラスタの代表特徴とする．これを visual word と呼ぶ．クラスタ数は，すなわち visual words の数となり，この数が BoF 表現における識別性の重要なパラメータとなる．しかし，visual words 数が多い場合，十分なメモリと各画像を visual words 表現する計算時間の問題がある．

次に，各 THI を得られた visual words の出現率で表す．まず，各画像から特徴量を抽出し，それがどの visual word に属するのかを計算する．この時，計算の高速化を図るため近似最近棒探索の手法で知られる

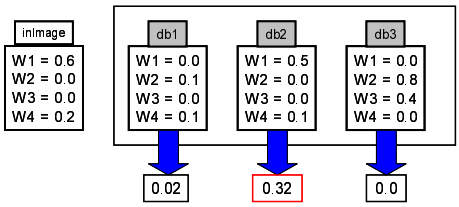


Fig.6 対応画像検索の方法例

(a) 撮影車 A　　(b) 撮影車 B

Fig.7 撮影車

Approximate Nearest Neighbor (ANN) を用い，近似解を求めた．また，このとき，visual words の重み付けとして TF-IDF 重みを用いた．画像 j における visual words　v_i の出現頻度を tf_{ij}，v_i が出現した画像数を df_j，全画像数を N としたとき，以下の式で表せる．

$$w_{ij} = \frac{tf_{ij}}{\sum_i tf_{ij}} \log \frac{N}{df_{ij}} \quad (1)$$

4.3　類似シーン検出への応用

対応区間を見つけたい映像から THI を作成し，Affine SIFT による特徴抽出を行う．入力する THI は，データベース作成に用いた visual words によるヒストグラムで表現する．クエリ画像をデータベースと同じ visual words で表現することでデータベースと同様の画像表現を行える．データベースとの類似度の計算には転置インデックスを用いる．visual words による画像表現を行う場合，画像数によっては大量のメモリを要するため，計算の高速化が求められる．本手法では，各 THI の BoF 表現がほとんどスパースであることを利用してこれを計算する．得られた類似度が最も高かった THI を対応付けられたシーンとして出力する．

5.　実験

5.1　実験概要

実際の映像により対応付けの性能評価を行った．図 7 のような撮影車により東京大学駒場キャンパス周辺，西新宿周辺，首都高速道路 3, 4 号線（片側 2〜3 車線）を制限速度内で流れに沿って走行しながら撮影した．

撮影そのものには，基となるデータベースを作成するため情報量の多い全方位カメラを用いた．この映像の一部を一般的なカメラモデルである透視投影に変換してから THI の作成以降の処理を行った．

5.2　実験 1：同一経路における性能評価

まず，性能評価のため，同一経路における THI をクエリ画像として与え実験を行った．実験では，複数の

経路で撮影された映像に対し，300〜400 フレームの区間 100ヶ所を選んだ．全特徴点として 95720 点が得られ，手法の精度評価のため visual words 数を 90000 点にした．

各区間において，リファインしたもの，電線除去のみ，雲除去のみ行ったものの 3 パターンのデータベースを構築した．また，それぞれのデータベースにおいて，時系列画像を 3 分の 1 フレームずつ間引きして THI を作成することで，高速走行により撮られた THI を想定することが出来，実験ではこれを高速走行とした．クエリ画像には，データベースを構成する 100 枚すべての画像を用いた．結果として表 1 を得た．

	リファイン	雲除去	電線除去
通常走行	1.00	0.94	0.92
高速走行	1.00	0.94	0.92

Table 1 リファインの有無による認識率

リファインした画像はどちらの走行においてもすべて対応付けることが出来た．雲除去と電線除去は，9 割以上の画像で対応付けを行えた．雲除去のみの場合の認識率が高い理由として，太い電線が複数本狭い間隔で配置されている場合，完全に除去しきれないためリファインしたデータベース画像も電線の影響を受けたと考えられる．

また，リファインした画像において visual words を 90000 から 10000 まで 10000 ずつ減らして認識率の変化を確認したところ，すべての場合において全画像で対応づけることができた．

5.3 実験 2：複数経路における性能評価

実験 2 では，実験 1 のそれぞれのデータベースにおいて，データベースに用いた映像とは異なる機会に，反対車線から撮影されたものから各 300〜400 フレームの区間 9ヶ所を選び，それぞれ THI を作成しクエリ画像とした．対応画像が複数ある場合は，そのうちのどれか 1 つがマッチングすれば正解とした．結果を表 2 に示す．

	リファイン	雲除去	電線除去
通常走行	0.89	0.56	0.89
高速走行	0.76	0.56	0.76

Table 2 リファインの有無による認識率

リファインしたデータベースにおいては，9 枚中 8 枚で対応付けを行うことが出来た．対応付けに失敗した原因としては，走行方向の違いによカメラの撮影方向に違いがあったことや T-junction の現れ方の違いなどが考えられる．

次に，visual words 数を実験 1 と同様に 90000 から 10000 まで変化させて実験を行った．クエリ画像には

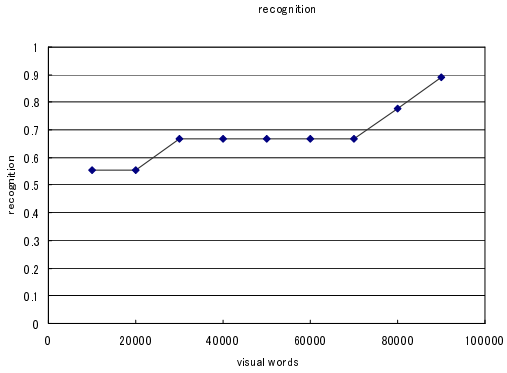


Fig.8 複数経路における visual words の推移に伴う認識率

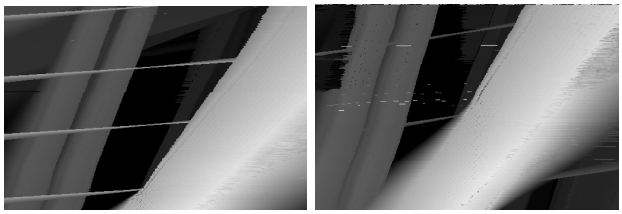
上記の 9 枚を用いた．結果を図 8 に示す．

visual words が増加するにつれて認識率も上がったが，全画像を対応付けることは出来なかった．また，実験 1 と比べた場合の認識率に比べて，大きく下がったことが分かる．

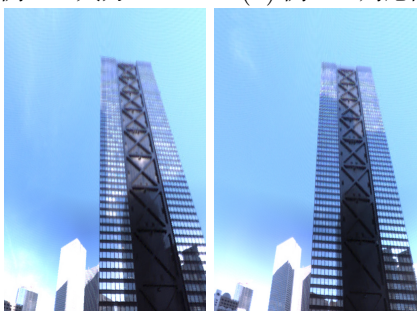
visual words が多い場合，認識率は高いが，計算コストが大きいため，更なる高速化を考える必要がある．

また，今回は，データベース数，入力画像数，共に少なかったので大量の画像を用いた検証が必要と考えられる．

図 9，図 10 に対応付けられた THI とシーケンス内の透視投影画像を示す．(a) を入力画像として与えた場合，(b) を対応付けられた画像として得ることが出来た．また，それぞれのシーケンス内の 1 フレームの画像を (c)，(d) とする．



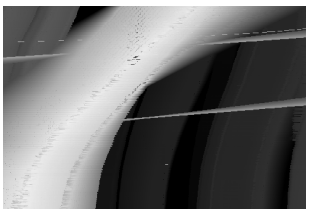
(a) 例 1 ：入力　(b) 例 1 ：対応画像



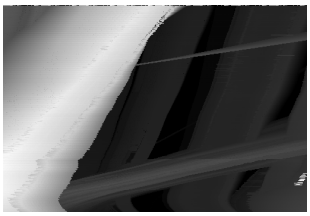
(c) シーケンス内画像 (d) シーケンス内画像

Fig.9 マッチング例

対応付けることが出来なかった画像を図 11 に示す．



(a) 例 2 ：入力

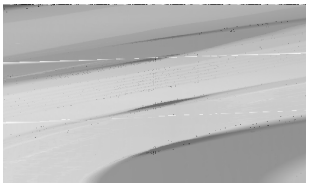


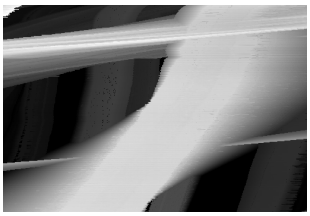
(b) 例 2 ：対応画像

(c) シーケンス内画像 (d) シーケンス内画像

Fig.10 マッチング例

入力画像として (a) を与えた時，対応付けられた画像として (b) が出力された．しかし，対応付けられた画像として出力されなければならないのは (c) である．





(a) 入力画像 (b) 出力画像

(c) 正解画像

Fig.11 失敗例

5.4 実験 3：YouTube 動画

実験 3 では，YouTube より同じ経路を走行した車載映像 3 本をダウンロードした．ダウンロードした映像は表 3 のとおりである．それぞれの映像でデータベースを作成し，そのデータベースに対して残りの映像から作成した THI を入力として与えた．認識結果を表 4 に示す．

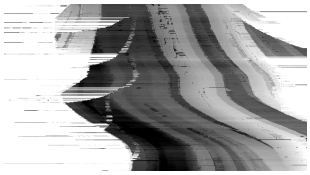
同じシーンとして認識された THI とそのシーケンス内の画像の例を図 12 に示す．

場所	YouTube1	YouTube2	YouTube3
フレームレート	30fps	18fps	30fps
フレーム数	4540 枚	2690 枚	3910 枚
解像度	640 × 360		
THI 数	12 枚	9 枚	11 枚

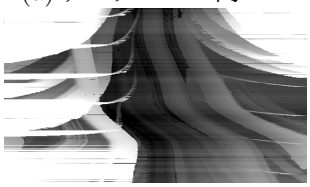
Table 3 YouTube 映像

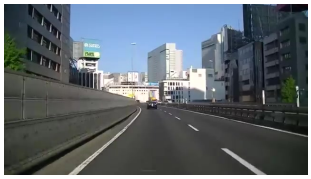
		YouTube1	YouTube2	YouTube3
入力	YouTube1	-	89%	72 %
	YouTube2	67%	-	91%
	YouTube3	67%	89%	-

Table 4 YouTube 映像の対応付け結果



(a) データベース内 THI (b) シーケンス内画像





(c) クエリ画像 (d) シーケンス内画像

Fig.12 対応付けられた THI とそのシーン

6. まとめ

本論文では，カメラ以外のセンサを用いずに THI と BoF を用いた複数のビデオデータより映像の対応付けを行う手法を提案した．精度の高い対応付けを行うためには，THI 作成のノイズとなる雲と電線の除去が必要性であり，これらを除去することにより精度が改善されることも確認した．

今後の課題として，データベース画像，入力画像を大量に用いた検証の必要が考えられる．また，対向車線で撮影された映像同士の対応付けを行う必要がある．

1) Dragomir Anguelov, Carole Dulong, Daniel Filip, Christian Frueh, Stephane Lafon, Richard Lyon, Abhijit Ogale, Luc Vincent, and Josh Weaver. Google street view: Capturing the world at street level. *Computer*, Vol. 43, , 2010.
2) Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cedric Bray. Visual categorization with bags of keypoints. In

In Workshop on Statistical Learning in Computer Vision, ECCV, pp. 1–22, 2004.

3) Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *Int. J. Comput. Vision*, Vol. 59, No. 2, pp. 167–181, September 2004.
4) David Kriegman, B. Vijayakumar, and Jean Ponce. Constraints for recognizing and locating curved 3d objects from monocular image features. In *European Conference on Computer Vision (ECCV)*, pp. 829–833, 1992.
5) Jun Miura and Koshiro Yamamoto. Robust view matching-based markov localization in outdoor environments. In *IROS*, pp. 2970–2976. IEEE, 2008.
6) Jean-Michel Morel and Guoshen Yu. Asift: A new framework for fully affine invariant image comparison. *SIAM J. Img. Sci.*, Vol. 2, No. 2, pp. 438–469, April 2009.
7) Jinge Wang, Shintaro Ono, and Katsushi Ikeuchi. Matching on-vehicle image and building model by temporal height image and proposal of texture mapping. *Technical Report of IEICE. IE, Image engineering*, Vol. 108, No. 425, pp. 103–108, 2009-01-28.
8) Jianxiong Xiao and Long Quan. Multiple view semantic segmentation for street view images. In *ICCV*, pp. 686–693. IEEE, 2009.
9) 小野晋太郎, 松久亮太, 池内克史, 川崎洋. 車載カメラ映像の時空間マッチングを利用した自車位置推定. 第 9 回 ITS シンポジウム 2010, 2010.
10) 村瀬洋. 車載カメラ映像の認識. 第 4 回音声ドキュメント処理 WS, Vol. 6, , 2010.