SIFT 特徴量を用いた交通道路標識認識

Road Sign Recognition using SIFT feature

高木雅成† 藤吉弘亘‡
Masanari Takagi†, Hironobu Fujiyoshi‡
†株式会社デンソー ‡中部大学
†DENSO CORPORATION, ‡Chubu University
E-mail: MASANARI_TAKAGI@denso.co.jp

Abstract

交通道路標識（以下，道路標識と表記）は交通安全に対して大きな役割を担っている．本稿は，ドライバが標識を見落とすことを防ぐため，車載カメラによって撮影された前方実環境画像から道路標識を検出し，その種類を認識する手法を提案する．

提案手法は，SIFT（Scale Invariant Feature Transform）[1]特徴量を用いて道路標識を認識する．SIFT特徴量を用いることにより，道路標識の拡大・縮小・傾き・オクルージョンに対してロバストな認識をすることに成功した．また，評価データは実環境下における約6時間の走行データ（画像枚数：215,200枚）を用いた．従来提案されている手法では，道路標識が必ず評価画像中に存在するもの，人間がその評価画像を見て道路標識の判断が容易なもの（白とび，黒つぶれは除外）等を評価画像として用いている．一方，本研究では人間の目をもってしても道路標識の判定が困難なシーン等を含んだ様々な走行データを評価画像としており，その結果，実環境下において認識率88.7%，誤検出回数16回という高い認識性能が得られ，本手法の有効性を明らかにすることができた．

1 はじめに

道路標識はドライバに道路の通行方法等を伝達する役割を有しており，道路交通の安全性・円滑性を確保する上で極めて重要な施設である．しかし現在の道路標識システムでは，道路標識の認識はドライバに委ねられており，運転中のドライバの精神状態などが原因で，道路標識の見落としが発生し，交通の安全性・円滑性に影響を及ぼす場合がある．そこで，道路標識を自動認識することにより，走行環境下における車両前方状況などの交通情報をドライバに提供し，交通事故を予防することが考えられる．その結果，ドライバの発見の遅れや判断の誤りを減少できるなら，約 6 割の交通事故を抑えることができるという試算がある[2].

従来，道路標識の検出・認識には，テンプレートマッチングや色情報を用いた手法が提案されている[3]-[11]. 認識対象物である道路標識は，車載カメラで撮影された画像（以下，入力画像と表記）中に様々な大きさで出現し，また傾きやオクルージョンを伴っている場合も存在する．テンプレートマッチングを用いた手法[3][4]では，その入力画像中に存在する様々な形状の道路標識を考慮したテンプレートを多数用意する必要がある．道路標識の場合，拡大・縮小・回転を行い変形させたテンプレート画像を多数用意することになる．さらには，照明変動等の外乱も考慮したテンプレート画像が必要なるため，膨大な手持ちデータとなる．

また，道路標識色を抽出することにより，道路標識の検出を行う手法も数多く提案されている[5]- [11]．しかし，実環境下における道路標識は，太陽光の照明変動により，様々な色に変化する．そのため，あらゆる照明変動を考慮した標識色抽出は困難であると言える．

そこで本稿は，色情報を用いず，SIFT 特徴量を用いた道路標識の拡大・縮小・傾き・オクルージョンに対してロバストな道路標識認識を提案する．

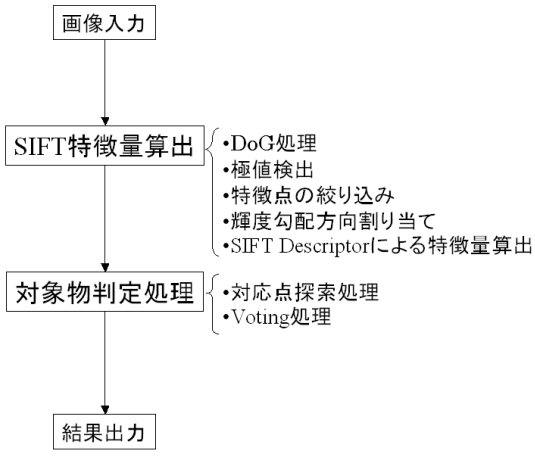


図１ 標識認識全体の処理の流れ

本稿では，まず SIFT を用いた道路標識認識アルゴリズム（図 1 参照）について述べた後，実際に車載カメラで撮影した画像を用いて，オフラインで認識アルゴリズムを評価した結果，および考察を述べる．

2 SIFT 特徴量算出

SIFT は，特徴の抽出に適した点（以下，特徴点と呼ぶ）を検出と，スケール変化・回転・照明変化に不変な特徴量を記述する 2 段階からなる．検出部は極値検出，特徴点の絞り込みからなり，記述部は輝度勾配方向の割り当てと，SIFT Descriptor による特徴量算出からなる．

2.1 極値検出

特徴点の候補となる極値を検出する．

2.1.1 Difference-of-Gaussian 処理

特徴点候補は，スケールの異なるガウス関数 $G(x,y,\sigma)$ と入力画像 $I(x,y)$を畳み込んだ平滑化画像の差分をとった画像から求める．このスケールの異なる平滑化画像の差分をとることを Difference-of-Gaussian（DoG）という．それぞれ以下の式により求められる．

$$L(x,y,\sigma)=G(x,y,\sigma)*I(x,y) \tag{1}$$

$$G(x,y,\sigma)=\frac{1}{2\pi\sigma^2}e^{-(x^2+y^2)/2\sigma^2} \tag{2}$$

DoG の結果の画像を $D(x,y,\sigma)$とすると，DoG 画像は次式で求まる．

$$D(x,y,\sigma)=L(x,y,\sigma_{i+1})-L(x,y,\sigma_i) \tag{3}$$

この処理を複数の異なるスケール間で行い，複数の DoG 画像を得る．

2.1.2 極値検出

得られた DoG 画像から極値の検出を行う．極値の検出は，DoG 画像 3 枚一組で行う．極値の探索を行いたいスケールの画像のある画素と，その周辺と，さらにその注目画素のある画像のスケールと隣接する上下のスケールの画像それぞれ注目画素と隣接する領域の画素を比較し，極値であった場合その画素を検出する．検出された極値は特徴点の候補点となる．

2.2 特徴点の絞り込み

検出された特徴点の候補点から特徴点として有効な点を選定する．特徴点候補位置での 2 次元ヘッセ行列 H を計算し，主曲率を求める．

$$H=\begin{bmatrix}D_{xx} & D_{xy}\\ D_{xy} & D_{yy}\end{bmatrix} \tag{4}$$

ヘッセ行列 H の固有値を a,b とすると，対角成分の和 $Tr(H)$と行列式 $Det(H)$は次のように計算できる．

$$Tr(H)=D_{xx}+D_{yy}=\alpha+\beta \tag{5}$$

$$Det(H)=D_{xx}D_{yy}-(D_{xy})^2=\alpha\beta \tag{6}$$

さらに，$a=\gamma b$ とすると次式のようになる．

$$\frac{Tr(H)^2}{Det(H)}=\frac{(\alpha+\beta)^2}{\alpha\beta}=\frac{(\gamma\beta+\beta)^2}{\gamma\beta^2}=\frac{(\gamma+1)^2}{\gamma} \tag{7}$$

この値は特徴点の固有値ではなく，固有値の比率のみで決まる．従って，固有値を求めずに主曲率を得ることができる．この値を閾値処理することで，有効な点のみを選定することができる．

2.3 輝度勾配方向の割り当て

検出された各特徴点に対して，輝度勾配方向を割り当てる．特徴点が検出された平滑化画像の各画素の勾配 $m(x,y)$とその勾配方向$\theta(x,y)$を，以下の式より求める．

$$m(x,y)=\sqrt{f_x(x,y)^2+f_y(x,y)^2} \tag{8}$$

$$\theta(x,y)=\tan^{-1}\frac{f_y(x,y)}{f_x(x,y)} \tag{9}$$

$$\begin{cases}f_x(x,y)=L(x+1,y)-L(x-1,y)\\ f_y(x,y)=L(x,y+1)-L(x,y-1)\end{cases} \tag{10}$$

求めた勾配 m の大きさと勾配方向θから，重み付方向ヒストグラムを作成する．次に，作成したヒストグラムから最大値となる方向を特徴点の輝度勾配方向として割り当てる．

2.4 SIFT Descriptor による特徴量算出

特徴点周辺領域を 2.3 で割り当てられた方向を基準とした軸に回転させる．この状態で特徴量を算出するため，回転に対する不変性を得ることができる．分割した周辺領域ごとの各ピクセルの輝度勾配ヒストグラムを作成する．周辺領域を少領域に分割し，それぞれについて輝度勾配ヒストグラムを作成する．周辺領域を 4×4 の 16 に分割し，それぞれに 8 方向の輝度勾配ヒストグラムを作成する．これにより，4×4×8 の 128 次元の特徴量が算出される．このとき使用するガウス窓のスケールは，特徴点が検出された画像のスケールに関連している．

以上により，検出された特徴点はスケール変化や回転にロバストな特徴量を持つ．

3 対象物判定処理

対象物判定処理は，テンプレート画像と入力画像の特徴点の特徴量を用いてマッチングをする対応点探索処理と，対応付けされた特徴点を用いた Voting 処理の 2 段階からなる．

3.1 対応点探索処理

あらかじめ抽出しておいたテンプレート画像の特徴点と，入力画像中から抽出された特徴点のマッチングを行

う．以下に，対応点探索の流れを示す．

テンプレート画像から M 個の特徴点が抽出され，与えられた入力画像から N 個の特徴点が抽出されたとすると，テンプレート画像と入力画像の特徴点は以下のようになる．

$$S_{temp}{}^{M} = \left(s_1{}^{M}, s_2{}^{M}, \cdots, s_{128}{}^{M}\right) \tag{11}$$

$$W_{in}{}^{N} = \left(w_1{}^{N}, w_2{}^{N}, \cdots, w_{128}{}^{N}\right) \tag{12}$$

テンプレート画像中の特徴点（m 番目）と，入力画像中特徴量類似度が最も小さくなる特徴点 n'を以下の式から求める．

$$n' = \arg\min_{n \in N}\left(\sum_{i}^{128}\sqrt{\left(s_i{}^{m} - w_i{}^{n}\right)^2}\right) \tag{13}$$

テンプレート画像中の特徴点 1 点に対して，入力画像中の全特徴点との類似度を求め（図 2 参照），その類似度を比較し，一番近い類似度の点同士を対応点とする．この処理を全テンプレートの全点に対して行う．

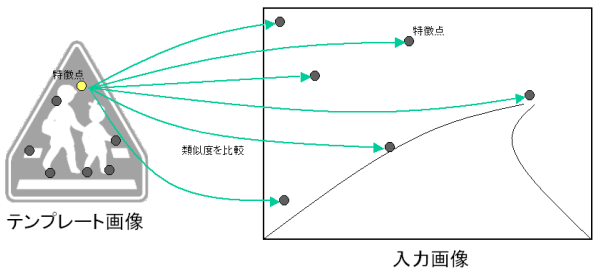


図 2　対応点探索

3.2　Voting 処理

3.1 の対応点探索処理でマッチングしたテンプレート画像と入力画像の特徴点を用いて，テンプレート画像の中心位置が，入力画像中のどこに位置するかを求める．求められた中心候補点を投票していくことにより，最終的な対象物の判定を行う．以下に，Voting 処理の流れを示す．

まず，あらかじめテンプレート画像中に，ある点（以下，基準点と表記）を与える．与える位置の条件は特に存在しないが，本稿では分かりやすくするために，テンプレート画像の中心とする．その与えられた基準点と各特徴点の位置関係を算出し，特徴点ごとに位置ベクトルを与える（図 3 参照）．

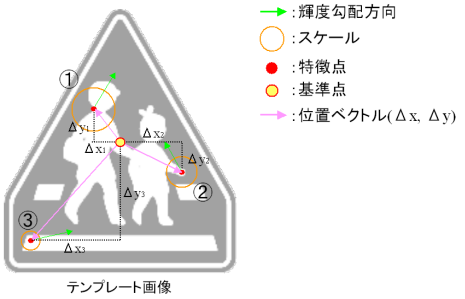


図 3　位置ベクトル算出

次に，テンプレート画像中の特徴点と基準点の位置ベクトルから，その特徴点とマッチングのとれた入力画像中の特徴点における基準点を求める．テンプレート画像と入力画像の特徴点を以下のように与える．

テンプレート画像の特徴点		入力画像の特徴点	
座標	$: (x_{temp}, y_{temp})$	座標	$: (x_{in}, y_{in})$
スケール	$: \sigma_{temp}$	スケール	$: \sigma_{in}$
輝度勾配方向	$: \theta_{temp}$	輝度勾配方向	$: \theta_{in}$
位置ベクトル	$: (\Delta x, \Delta y)$		

それぞれの特徴点の座標，スケール，および位置ベクトルから入力画像中の基準点(X,Y)は次式のようになる．

$$\begin{cases} X = x_{in} + \dfrac{\sigma_{in}}{\sigma_{temp}} \times \sqrt{\Delta x^2 + \Delta y^2} \times \cos\left(\theta + \theta_{temp} - \theta_{in}\right) \\ Y = y_{in} - \dfrac{\sigma_{in}}{\sigma_{temp}} \times \sqrt{\Delta x^2 + \Delta y^2} \times \sin\left(\theta + \theta_{temp} - \theta_{in}\right) \end{cases} \tag{14}$$

$$\text{但し、} \theta = \arctan\left(\frac{\Delta y}{\Delta x}\right) \tag{15}$$

以上の処理をマッチングしたすべての特徴点に対して行い，入力画像中に基準点の投票を行う．つまり，入力画像中に対象物が存在するならば，求められる基準点は同じ位置に存在する（図 4 参照）．

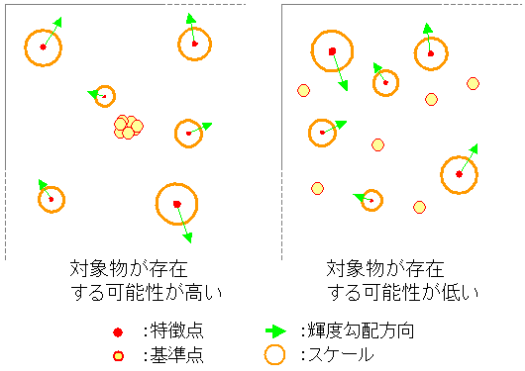


図 4　基準点投票結果

投票された基準点が隣接している場合，それらの基準点をクラスタリングすることにより投票数を求める．投票数にある閾値（以下，投票閾値と表記）を設定し，その閾値を超えた場合入力画像中に対象物が存在すると判定する（図 5 参照）．

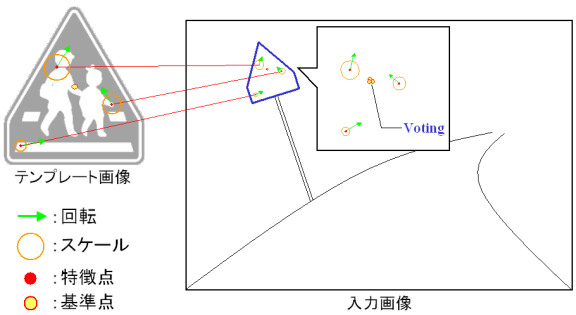


図 5　特徴点のマッチングと Voting 処理

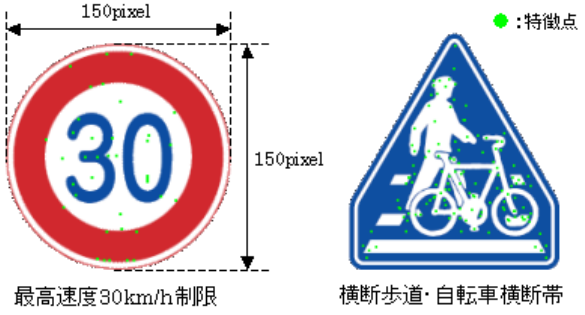
4　評価実験

実環境下における走行データを用いた実験により提案手法の有効性を示す.

4.1　実験方法

認識対象標識として，標識自体の持つ危険度や設置頻度等を考慮し，17 種類とし，車内ルームミラー部に固定したカメラによって車載前方実画像を収集したのち（10fps で収集），オフラインで本手法の評価を行った. 実際にテンプレート画像として用いたイラスト画像を図 6(a)に示し，テンプレート画像からの特徴点抽出例を図 6(b) に示す.

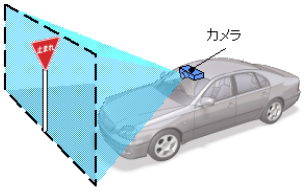
(a)認識対象イラスト画像



(b)特徴点抽出例

図 6　認識対象標識

本手法の評価に用いたカメラは，PIXIM 社製のハイダイナミックレンジカメラである（図 7，および表 1 参照）. ダイナミックレンジが 120dB ということで，画像の白飛びや黒つぶれが発生しにくく，実環境下では非常に有効なカメラである.



(a)データ収集システム

(b)使用カメラ外観

図 7　走行シーン撮影環境

表 1　PIXIM 社製カメラの主なスペック

画素数	640×480 画素
水平画角	77.6°
垂直画角	57.6°
ダイナミックレンジ	120dB

実環境下の走行シーンは，晴れの天候下での市内と郊外における昼間および夕刻時に撮影されたものである. ３日間の実環境下における走行シーンの撮影を行った. 撮影した走行シーンの詳細な情報を以下に示す.

■走行データ収集 1 日目
天候：晴れ
収集時間：約 70 分（15：15～16：25）
収集画像枚数：43,700 枚
撮影された認識対象標識個数：143 個

■走行データ収集 2 日目
天候：晴れ
収集時間：約 130 分（10：00～12：10）
収集画像枚数：76,000 枚
撮影された認識対象標識個数：365 個

■走行データ収集 3 日目
天候：晴れ
収集時間：約 160 分（15：00～17：40）
収集画像枚数：95,500 枚
撮影された認識対象標識個数：172 個

以上，約 6 時間の走行シーン 215,200 枚を入力画像とし，本手法の標識認識に適用し，認識性能の評価を行った. 他の論文[5][6]では，入力画像中に必ず標識が存在する画像や，人間がその画像を見て標識種類の判断が容易な画像（白とび，黒つぶれは除外）を入力画像としているもの，または実環境下における短時間の走行データを入力画像として扱っているものが多い. しかし本稿では，実際の風景を人間の目で見たとしても，標識の判断が困難なシーンを含めた 215,200 枚の実環境画像を入力画像とすることにより，実環境下における本手法の有効性を検証した.

4.2　実験結果

表 2 に本手法の評価結果を示す．全標識の認識率は 88.7%, 誤認識回数は 3 回, 誤検出回数は 16 回となった.

表 2　評価結果

	全標識	一時停止 徐行	横断帯	最高速度 追い越し禁止 転回禁止
認識率	88.7% (603/680)	86.7% (98/113)	85.9% (146/170)	90.4% (359/397)
誤認識	3回	0回	0回	3回
誤検出	16回	6回	1回	9回

また，実際に標識認識されたシーンの入力画像と認識結果画像の例を図 8 に示す.

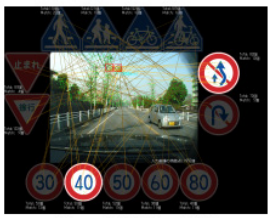
(a) 入力画像①

(b) 認識結果画像①

(c) 入力画像②

(d) 認識結果画像②

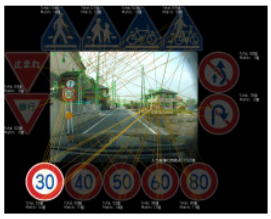
(e) 入力画像③


(f) 認識結果画像③

(g) 入力画像④

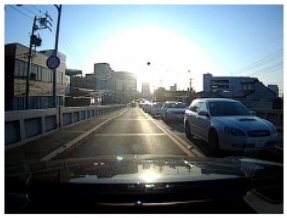
(h) 認識結果画像④

(i) 入力画像⑤


(j) 認識結果画像⑤

(k) 入力画像⑥

(l) 認識結果画像⑥


(m) 入力画像⑦

(n) 認識結果画像⑦

図 8　認識シーン

“認識率”とは，標識設置数単位での認識率のことである．表 3 に認識シーケンスの例を示す．画面上に標識が現れて，画面から消失するまでに一回でも認識に成功したら，認識できたものとする．ただし，誤認識が発生している場合は，認識成功フレーム数と誤認識フレーム数の割合の高いほうの結果を優先する (表 3(f)参照)．また，両割合が同数の場合は，近距離での認識結果に重みをつけて判定をする（表 3(e)，(g)参照）.

認識が可能な標識サイズ（画像上における）は，30～40pixel 以上である.

表 3　認識成功/失敗の例

	フレーム番号										認識判定
	t	t+1	t+2	t+3	t+4	t+5	t+6	t+7	t+8	t+9	
(a)	・	・	○	○	○	○	○	○	・	・	成功
(b)	・	・	－	－	－	－	－	－	・	・	失敗
(c)	・	・	－	－	－	－	○	－	・	・	成功
(d)	・	・	－	－	－	×	－	－	・	・	失敗
(e)	・	・	－	－	－	×	○	－	・	・	成功
(f)	・	・	－	－	×	×	○	－	・	・	失敗
(g)	・	・	－	－	×	×	○	○	・	・	成功

○：認識に成功したフレーム　　－：未認識のフレーム
×：誤認識のフレーム

“誤認識”とは，標識の検出には成功しているものの，シンボル識別（標識内部の識別）に失敗した回数のことである（フレーム単位）．例えば，最高速度 30km/h 制限の標識を最高速度 50km/h 制限と認識した場合を指す.

“誤検出”とは，標識でないものを標識であると判断した回数のことである．1 フレーム中に誤検出回数が 1 回とは限らない（図 9 参照）．つまり，すべてのフレームに対して 2 回ずつ誤検出が発生した場合，誤検出回数は 430,400 回となる.

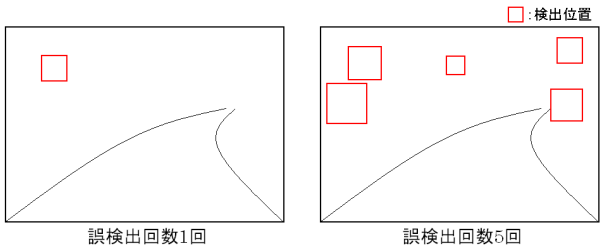


図 9　誤検出シーン例

4.3　考察

4.3.1　　認識率について

認識対象標識 680 個中 603 個の認識に成功し，認識率 88.7%を得られたが，残りの 77 個の標識については認識できていない．そこで，未認識の標識に対して 6 つの原因に分類した．図 12 に各未認識のシーン例を示す．

【原因 1】　解像度不足　3.9%（3/77）

画面で標識が消失するまでに，認識可能なサイズである 30～40pixel に達しなかったものが該当する．
この未認識は，カメラ画角の調整や高解像度のカメラを用いることにより解決できる．しかし，高解像度のカメラを用いると，認識する領域の画素数が増加するため，処理時間が増加する．

【原因 2】　標識の傾き　13.0%（10/77）

本手法では，回転に不変な特徴量を用いているため，画面奥行き方向（z 軸）の回転には全く影響を受けない．しかし，縦・横方向（x 軸，y 軸）の回転にはある程度までしか対応できない．なぜなら，縦・横方向の回転が発生すると画面上の標識が変形してしまう．そのため，特徴点は抽出されるものの，特徴量が大幅に変化してしまい，特徴点の対応付けに失敗し，未認識となる．この未認識は，テンプレート画像を 3 次元変形させた画像をいくつか用いることで解決できる．しかし，テンプレート画像を増やした分処理時間が増加する．

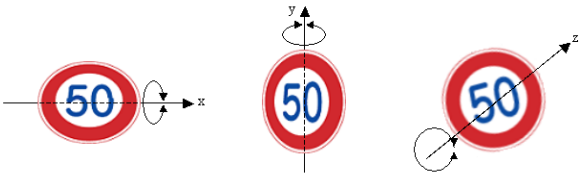


図 10　標識の 3 次元変形

【原因 3】　標識の日焼け　23.4%（18/77）

長い年月設置されている標識は，日焼け，酸化等により標識自体が劣化してしまっているため，標識の模様がかすれてしまっている．そのため，画像上ではコントラストが低下し，特徴点が抽出できず未認識となる．

【原因 4】　白とび　9.1%（7/77）

順光時（太陽を背面にしたとき）に撮影を行うと，標識の傾き具合や太陽の位置などの条件が重なってしまった場合，標識が白く塗りつぶされてしまうことがある．そのため，画面上では標識内部の模様が映っておらず未認識となる．

【原因 5】　黒つぶれ　28.6%（22/77）

逆光時（太陽を正面にしたとき）に撮影を行うと，白飛びとは逆の現象である黒つぶれが発生する．そのため，標識が黒く塗りつぶされてしまい，画面上では標識内部の模様が映っておらず未認識となる．

今回の評価において，ハイダイナミックレンジカメラを撮影に用いた．そのため，かなりの白とび・黒つぶれを回避することができた．しかし，ハイダイナミックレンジとはいっても性能に限界がある．今回発生した白とび・黒つぶれのシーンは，そのシーンの実際の風景を人間の目で見たとしても，標識の判定をすることは困難なシーンである．

【原因 6】　その他　22.0%（17/77）

その他に分類された標識は，設置されている標識の状態や太陽光の影響がなく，人間の目では十分に判断可能なシーンのものである．つまり，本手法の認識アルゴリズムに原因があると考えられる．そのため，その他に分類された 17 個の標識に対してさらに未認識の原因究明を行った．

まず，その他に分類された未認識の標識が撮影されているシーンを抽出し，手作業で標識の切り出し作業を行う．次に，切り出された部分以外を黒く塗り潰した入力画像を作成する．最後に，その入力画像を本手法に適用し未認識の原因究明を行い，3 つの原因に分類した（図 11 参照）．

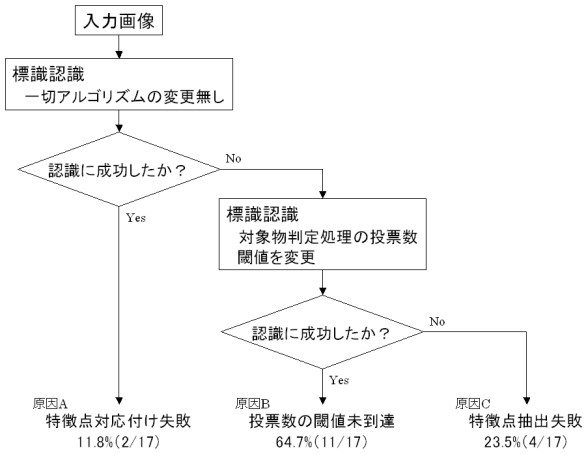


図 11　その他に分類された未認識原因究明フローチャート

“特徴点対応付け失敗”とは，入力画像の特徴点抽出には成功しているが，テンプレート画像の特徴点との対応点探索処理で失敗しているものである．つまり，入力画像中の標識内部以外の特徴点（樹木・建物等か

ら抽出される特徴点）とマッチングしてしまった場合を指す．この未認識は，標識を認識する入力画像の領域を制限することにより対策可能である．しかし，領域を制限することによる別の未認識や標識認識の遅れという問題が生じる．

“投票数の閾値未達成”とは，基準点の投票数が対象物判定処理で用いている投票数閾値を超えていないものである．この未認識は，投票数閾値を下げることにより対策可能である．しかし，閾値を下げることにより，誤検出，つまり標識でないものを標識と判定する回数が大幅に増加する．

“特徴点抽出失敗”とは，入力画像中の標識内部の特徴点がうまく抽出されておらず，対応点探索処理が行えていないものを指す．この未認識は，標識がオーバーヘッド部に設置されているシーンで発生している．オーバーヘッド部に設置された標識は，画面から消失する手前で認識可能なサイズの 30pixel 程度となる．本手法で確実に標識を認識するためには，40pixel 程度の大きさが必要なため，30pixel では認識が困難な場合が存在する．そのため，未認識原因“解像度不足”と同様の対策により解決できる．

図 12　未認識シーン例

680 個の認識対象標識中に発生した未認識の標識を 6 つの原因に分類した．この原因の【原因 1】から【原因 5】は，評価に用いたカメラの性能や実環境下で設置されている標識の状態に起因するところが大きい．そのため，これからの原因で発生した未認識を除き，本手法の認識アルゴリズムに原因があるもののみを考慮した認識率は，97.5%となり，本手法の有効性は非常に高いといえる．

4.3.2　誤認識について

誤認識は，最高速度 30km/h 制限の標識を最高速度 50km/h 制限と認識した場合，標識内部の識別失敗を指す．今回の評価では，3 回の誤検出が発生している．誤検出が発生しているシーンは，画像上の標識サイズが 30pixel 程度のときに発生している．つまり，標識が認識されだす初期の時点で発生している．画面上から標識が消失する付近時点での誤検出は一度も発生していない．誤認識の対策は，時系列処理を加えることにより回避が可能である．つまり，1 フレームでの認識結果を出力するのではなく，マルチフレームでの認識結果を用いて，最終的な標識認識の判定を行う処理を加えることを指す．これにより，認識結果の信頼性は上がるが，複数フレームの認識結果を用いるため，認識結果の出力までに時間を要す．

4.3.3　誤検出について

入力画像 215,200 枚に対して，16 回の誤検出が発生した．誤検出の傾向を掴むために，その 16 回の誤検出が発生したシーンを抽出し，誤検出の原因究明を行った．入力画像上で誤検出が発生している位置には，樹木や建物等が存在しており，その物体から抽出された特徴点とマッチングし，さらに Voting 処理を行った結果，投票数閾値を超えて標識であると判定されている．しかし，テンプレート画像と類似した模様が入力画像中に存在しているわけではないため，誤検出の発生するシーンの傾向を把握することが困難であった．

誤検出の対策は，Voting 処理における投票数閾値の変更等で可能である．しかし，閾値を上げることにより，未認識，つまり標識を見落とす回数が増加する．

4.3.4　処理時間について

今回評価では，CPU Pentium D 3.2GHz のパーソナルコンピュータを用いた．1 フレームあたりの処理時間は，平均 1.5s である．その処理時間の約 8 割を SIFT 特徴量算出における特徴点抽出処理に要している．しかし，本手法は単純な繰り返し演算から成り立っているため，比較的ハードウェア化しやすいと考えられる．そのため，本手法のハードウェア化により，大幅な処理時間短縮が実現できる可能性がある．

5　むすび

ドライバが標識を見落とすことを防ぐため，車内ルームミラー部に設置されたカメラにより，撮影された画像から標識の種類を認識する手法を提案した．車両前方実画像を取り込み，入力画像とする．その入力画像から抽出された特徴点と，あらかじめ抽出しておいたテンプレート画像における特徴点の SIFT 特徴量の類似度を比較し，対応点探索を行う．そのマッチングした特徴点を用いて，基準点の投票を行う Voting 処理を行い，最終的な標識の判定を行う．

本手法は，スケール変化・回転・照明変化に不変な特徴量を用いている．そのため，大きさを変化させた複数毎のテンプレート画像が必要ない．また，標識が奥行き方向に回転し，傾いた場合にも問題なく認識が可能である．さらに，特徴点による識別を行っているため，標識が樹木等で隠れてしまい，オクルージョンが発生している場合にも有効である．

今回は標識自体の持つ危険度や設置頻度等を考慮し，認識対象標識を 17 種類とした．新しく認識対象標識を追加したとしても，これまでの認識性能に影響を与えることはない．その点で， 学習ベースとなる SVM やニューラルネットワーク等を用いた認識手法より拡張しやすい手法と言える．

実環境下で約 6 時間（215,200 枚）の様々な走行シーンを収集し，オフラインで本手法の評価を行い，非常に高い認識性能を得ることができ，本稿の提案手法が実環境下において有効であることを明らかにすることができた．また，未認識・誤認識・誤検出の原因究明を行うことにより，それぞれの対応策を考案した．今後は，本手法をどのような用途に用いるか，ドライバへの情報提供，車両制御といったアプリケーションを考慮した上で，対策を行う必要がある．

今後の課題は，より遠方からの標識認識，本手法の認識アルゴリズムの高速化，および本手法の標識認識以外への適用が挙げられる．

謝辞

本研究を進めるにあたり，実環境下での走行シーン撮影，および標識認識デモ作成にご尽力していただいた当社開発部の皆様には深く感謝を申し上げます．

参考文献

[1] David G. Lowe, “Distinctive image features from scale-invariant keypoints”, Journal of Computer Vision, 60, 2 , pp. 91-110, 2004.

[2] AHRSA 走行支援道路システム開発機構，”事故原因はどこに，” http://www.ahsra.or.jp/whats_ahs/02/index.html

[3] 竜円，長坂，鈴村，”円形状の道路標識の認識”，電子情報通信学会技術研究報告，Vol.99，No.609，pp.17-22，2000

[4] 岩村，根岸，大町，阿曽，”変形を許容する高速なテンプレートマッチングによる情景画像中文字認識”，画像の認識・理解シンポジウム 2006 講演論文集，pp.906-911

[5] Yangxing Liu, Takeshi Ikenaga, Satoshi Goto, “Geometrical, Physical and Text/Symbol Analysis Based Approach of Traffic Sign Detection System” Intelligent Vehicles Symposium 2006, pp.238-243

[6] Claus Bahlmann, Ying Zhu, Visvanathan Ramesh, “A System for Traffic Sign Detection, Tracking, and Recognition Using Color, Shape, and Motion Information”, Intelligent Vehicles Symposium 2005, pp.255-260

[7] Jim Torresen, Jorgen W. Bakke and Lukas Sekanina, “Efficient Recognition of Speed Limit Signs”, IEEE Conference on Intelligent Transportation Systems (ITSC 2004), p. 652-656

[8] S. H. Hsu, C. L. Huang. “Road sign detection and recognition using matching pursuit method” Image and Vision Computing, 19:119-129, 2001

[9] 明珍，景山，西田，”カラー情景画像における円形道路標識の認識に関する検討”， 電子情報通信学会技術研究報告，PRMU 2004-262

[10] 莫，青木，”カラー画像における道路標識の認識”，電子情報通信学会論文誌（D-Ⅱ），Vol.87，No.12，pp.2124-2135，2004

[11] 松浦，山内，高橋，”特定色判別と領域限定を用いた円形道路標識の抽出”，電子情報通信学会論文誌（D-Ⅱ），Vol.85，No.6，pp.1075-1083，2002