

射影変換を用いた単眼カメラ画像からの障害物の高さ情報推定

中島 慶子[†] 田中 志歩[†] 大川 剛直[†] 伊東 敏夫^{††}

[†] 神戸大学大学院システム情報学研究科 〒 657-8501 兵庫県神戸市灘区六甲台町 1-1

^{††} ダイハツ工業 〒 563-0044 大阪府池田市ダイハツ町 1-1

あらまし 近年、コンピュータビジョンの技術が自動車の予防安全に広く取り入れられている。従来の研究として、車両前後方の歩行者や他の車両など、ある程度高さを持った障害物の認識の研究が進められてきた。しかし、駐車場の輪止めや道路の縁石など、車両が乗り越えられないものの高さを検出し、ドライバの視覚支援をすることも重要である。そこで本論文では、コストが低くカメラキャリブレーションが容易な単眼カメラ画像を用いて、障害物の高さ情報を推定する手法を提案する。射影変換の原理と上面視に変換した画像の特徴に着目し、カメラモデルから導出した画像と高さとの関係式と、テンプレートマッチングによる対応点探索を利用して高さ情報の推定を目指す。実験により本手法の有効性を示す。

キーワード ITS 射影変換 単眼カメラ画像 三次元情報復元

1. はじめに

近年、道路交通において、交通事故の多発、交通渋滞、環境負荷など多くの問題を抱えており、高度交通システム-ITS(Intelligent Transport System)[1]の研究が進められている。ITSの中で自動車を知能化して安全・利便性の向上を狙うものが運転支援システムである。運転支援システムは、自動車の目となり、ドライバの視界を補助するために、車両周辺環境を認識し三次元情報を獲得することが必要になる。近年、運転支援システムの開発にコンピュータビジョンの技術が広く取り入れられている。コンピュータビジョンとは人が視覚認識により実世界の情報を取得する機能をコンピュータ上に実現しようとする研究分野である。特にカメラ画像からの3次元復元は、人の視覚支援を始めとして幅広い応用が期待されている。単眼カメラや複眼ステレオカメラなどが多く用いられているが、ステレオカメラはコストが高くカメラキャリブレーションが困難であるという問題がある。一方、単眼カメラの場合はコストが低くカメラキャリブレーションも容易であることから、本研究では単眼カメラを用いた運転支援システムを対象とする。

カメラを利用した運転支援システムの従来の研究として、車両前後方の歩行者や他の車両など、ある程度高さを持った障害物の認識の研究が進められてきた。しかし、駐車場の輪止めや道路の縁石など、車両が乗り越えられないものの高さを検出し、ドライバに代わり車両を制御することも重要である。自動車が乗り越えられるような障害物は問題無いが、高さのある障害物であれば運転の妨害になり、最悪の場合事故につながる可能性がある。このような問題を解決するために、ドライバの支援や車両の制御を行うシステムのニーズが高まっている。

単眼カメラを用いた三次元復元には多数の手法が提案

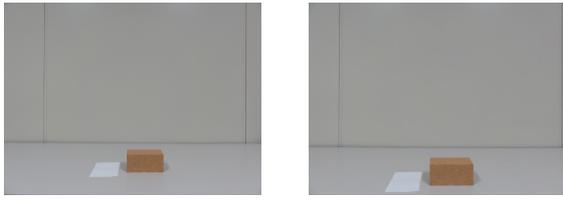
されている。代表的なものに、SfM(Structure from Motion)の代表であるTomasi-Kanade-factorization[3]がある。SfMとは、与えられた複数枚の画像の特徴点を抽出し対応付けを行うことで、そこに映っている物体やシーンの形、同時にカメラキャリブレーション自体を復元する、という手法である。ここで用いる画像はカメラ視点のものを用いるが、特徴点が少ない画像に適用すると正しく復元できない可能性がある。

そこで、カメラ視点の画像を上面視に射影変換することを考える。カメラの光軸方向に直進するカメラで撮影した時系列画像を上面視画像に変換したときに、カメラの光軸に平行な面とそれ以外の面では、形状の変化に違いが見られる。これは、つまり高さの変化が無い領域とそれ以外の領域には形状の変化が見られると言い換えることができ、この形状の変化に着目することで高さ情報が得られることが期待できる。また、復元する障害物はカメラの光軸より下にあるものと仮定する。これにより、上面視画像に射影変換すると必ず障害物を見下ろしたような画像になり、形状の変化の違いを利用することができる。以上より、特に高さ情報を推定する上でこの特徴は有効であると考えられるため、本論文では上面視への射影変換を用いて障害物の高さ情報を推定する手法を提案する。まず、上面視画像への射影変換の特徴と変換の手順を説明し、カメラモデルから導出した画像と高さとの関係式と、テンプレートマッチングによる対応点探索を利用して高さ情報を推定する。

2. 上面視に射影変換した画像の特徴

1章で既存手法としてSfMの手法について述べたが、この手法ではカメラで撮影した複数枚の時系列画像をそのまま用いて三次元情報を復元する。

図1のように、視点 O_1 と O_2 の位置のカメラのカメ



(a) I(t) (b) I(t+1)
図1 カメラ画像

ラ座標をワールド座標と等しい向きに設定し、地面上の四角形 S を撮影する．視点 O_1 は視点 O_2 をカメラの視点方向に直進させたものである．

このような時系列画像では、画像間で障害物の大きさが変化する．第1章に挙げた SfM の手法は特徴点の対応付けによってカメラの運動推定を行うものであるが、このような障害物の大きさが変化している画像では対応付けの精度が下がることが考えられる．また、このように特徴が少ない環境で撮影した画像には特徴点自体が少なく、上手くカメラの運動を推定することができないと考えられる．

そこで、本研究では、カメラ画像を上面視画像に変換したものを利用する．図1のように、視点 O_1 と O_2 の画像平面をそれぞれ X 軸のまわりに 90 度回転させた視点 O'_1 と O'_2 から見たような上面視画像に射影変換する．このとき、上面視画像に射影変換しているため視点 O'_1 と O'_2 のそれぞれの画像平面は地面と平行であることから、視点 O'_1 と O'_2 の高さが等しいとき、視点 O'_1 と O'_2 の画像平面上に投影される四角形 S_1, S_2 は合同となる．このことから、上面視画像の場合、射影変換後の2枚の上面視画像上では地面に描かれているものの形状が変化しないことがわかる．つまり、カメラ光軸と垂直な面には形状の変化が見られない．逆に言えば、それ以外の部分には何らかの形状の変化が現れると考えられ、これを解析することで障害物の高さ情報を得られることが期待できる．

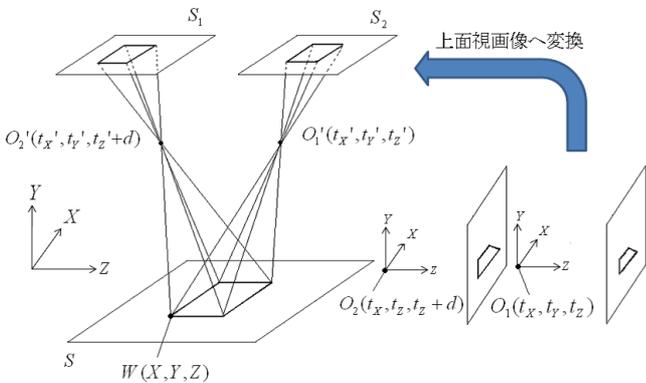


図2 上面視画像の特徴

3. 透視投影モデルを用いた上面視画像への変換

ここでは、カメラ画像を上面視画像に変換する手順について、透視投影モデルを用いて説明する．図1と同様の状況において、障害物を点 $O_1(t_X, t_Y, t_Z)$ のカメラで撮影した画像を、点 $O'_1(t'_X, t'_Y, t'_Z)$ のカメラで撮影したような上面視画像に変換することを考える．なお、 $O_1(t_X, t_Y, t_Z)$ 、 $O'_1(t'_X, t'_Y, t'_Z)$ はワールド座標系から見たカメラの位置である．

上面視画像への射影変換の大まかな手順は以下のとおりである．

(1) 点 $O(t_X, t_Y, t_Z)$ から撮影したカメラ画像の画素座標 $P(u, v)$ からワールド座標 $W(X, Y, Z)$ を求める．

(2) 求めたワールド座標 $W(X, Y, Z)$ を点 $O'(t'_X, t'_Y, t'_Z)$ から撮影したような上面視画像の画素座標 $P'(u', v')$ に対応させる．

まず、ステップ1において、カメラ画像の画素座標 $P(u, v)$ から画像座標 $M(x, y)$ を求める．カメラの焦点距離を f 、画像の中心座標を $M_0(u_0, v_0)$ とすると、式(1)(2)で表される．

$$\begin{cases} x = \frac{u - u_0}{f} & (1) \\ y = -\frac{v - v_0}{f} & (2) \end{cases}$$

次に、画像座標 $M(x, y)$ からカメラ座標 $C(X_C, Y_C, Z_C)$ を式(3)(4)から求める．

$$\begin{cases} X_C = xZ_C & (3) \\ Y_C = yZ_C & (4) \end{cases}$$

最後にカメラ座標 $C(X_C, Y_C, Z_C)$ からワールド座標 $W(X, Y, Z)$ を求める．カメラ座標 $C(X_C, Y_C, Z_C)$ をワールド座標 $W(X, Y, Z)$ を用いて表すと式(5)のようになる．

$$\begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} = \begin{bmatrix} X - t_X \\ Y - t_Y \\ Z - t_Z \end{bmatrix} \quad (5)$$

式(1)~(5)を用いて画素座標 $P(u, v)$ からワールド座標 $W(X, Y, Z)$ を求める．しかし、このままでは未知数が3つに対して式が2つしかなく、未知数が求められない．そこで、 $Y = 0$ という仮定を置く．この仮定により、画素座標 $P(u, v)$ からワールド座標 $W(X, Y, Z)$ を求めることができる．

次に、ステップ2において、ステップ1で求めたワールド座標 $W(X, Y, Z)$ から上面視のカメラ座標 $C'(X'_C, Y'_C, Z'_C)$ を求める． $C'(X'_C, Y'_C, Z'_C)$ の座標軸は、 $C(X_C, Y_C, Z_C)$ の座標軸を X 軸回りに 90 度回転させたものに等しいことから、式(6)のように表される．

$$\begin{bmatrix} X'_C \\ Y'_C \\ Z'_C \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \frac{\pi}{2} & \sin \frac{\pi}{2} \\ 0 & -\sin \frac{\pi}{2} & \cos \frac{\pi}{2} \end{bmatrix} \begin{bmatrix} X - t'_X \\ Y - t'_Y \\ Z - t'_Z \end{bmatrix} \quad (6)$$

次に、カメラ座標 $C'(X'_C, Y'_C, Z'_C)$ から上面視の画像座標 $M'(x', y')$ を式 (7) により求める。

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \frac{1}{Z'_C} \begin{bmatrix} X'_C \\ Y'_C \end{bmatrix} \quad (7)$$

最後に画像座標 $M'(x', y')$ から画素座標 $P'(u', v')$ を求める。カメラの焦点距離を f 、画像の中心座標を $M_0(u_0, v_0)$ として、式 (8) を用いて求めることができる。

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = \begin{bmatrix} f & 0 & u_0 \\ 0 & -f & v_0 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \quad (8)$$

以下に、カメラ画像を上面視画像に射影変換した例を示す。図 1 の視点 O_1 と O_2 から障害物を撮影した画像を図 3 の $I(t)$, $I(t+1)$, それぞれの上面視画像を図 4 の $I'(t)$, $I'(t+1)$ とする。 $I(t)$, $I(t+1)$ は地面に積み木と紙を置いたものである。 $I'(t)$ と $I'(t+1)$ を見比べてみると、積み木の上面と紙の形状に変化は無いが、側面部分の形状が変化していることが分かる。カメラの光軸と平行な面は上面視画像に射影変換をしても形状の変化は無く、それ以外の部分には形状に変化が見られることが確認できる。



(a) $I'(t)$ (b) $I'(t+1)$
図 3 上面視画像

4. 上面視画像における障害物の形状の変化の定式化

上面視画像における障害物の形状の変化を定式化するために、図 1 と同様の状況を考える。点 $O_1(t_X, t_Y, t_Z)$ の位置のカメラから障害物を撮影したとき、障害物の一点 $W(X, Y, Z)$ が投影される射影変換画像上の点 $P_1(u_1, v_1)$ を、透視投影モデルを用いて求める。カメラ座標を $C_1(X_{C_1}, Y_{C_1}, Z_{C_1})$ 、画像座標を $M_1(x_1, y_1)$ 、画像の中心座標を $M_0(u_0, v_0)$ として、式 (9)(10) を用いて式 (11) のように求めることができる。

$$\begin{bmatrix} X_{C_1} \\ Y_{C_1} \\ Z_{C_1} \end{bmatrix} = \begin{bmatrix} X - t_X \\ Y - t_Y \\ Z - t_Z \end{bmatrix} \quad (9)$$

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \frac{1}{Z_{C_1}} \begin{bmatrix} X_{C_1} \\ Y_{C_1} \end{bmatrix} \quad (10)$$

$$\begin{bmatrix} u_1 \\ v_1 \end{bmatrix} = \begin{bmatrix} f & 0 & u_0 \\ 0 & -f & v_0 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} \quad (11)$$

ここから、点 $O'_1(t'_X, t'_Y, t'_Z)$ の位置のカメラで撮影したような画像に射影変換をする。点 $O'_1(t'_X, t'_Y, t'_Z)$ のカメラのカメラ座標は、 $O_1(t_X, t_Y, t_Z)$ のカメラのカメラ座標を X 軸周りに 90° 回転させたものとする。まず、点 $P_1(u_1, v_1)$ に投影された、点 $O'(t'_X, t'_Y, t'_Z)$ から見た三次元空間上の点 $W'(X', Y', Z')$ (空間的には点 $W(X, Y, Z)$ と同じ点だが、分かりやすいように区別する) を透視投影モデルを用いて求める。カメラ座標 $C'(X'_{C_1}, Y'_{C_1}, Z'_{C_1})$ は式 (12)(13) で求めることができる。

$$\begin{cases} X'_{C_1} = x_1 Z'_{C_1} \\ Y'_{C_1} = y_1 Z'_{C_1} \end{cases} \quad (12)$$

$$\begin{cases} X'_{C_1} = x_1 Z'_{C_1} \\ Y'_{C_1} = y_1 Z'_{C_1} \end{cases} \quad (13)$$

また、カメラ座標 $C'_1(X'_{C_1}, Y'_{C_1}, Z'_{C_1})$ をワールド座標 $W(X', Y', Z')$ を用いて表すと式 (14) のようになる。

$$\begin{bmatrix} X'_{C_1} \\ Y'_{C_1} \\ Z'_{C_1} \end{bmatrix} = \begin{bmatrix} X' - t'_X \\ Z' - t'_Z \\ -Y' + t'_Y \end{bmatrix} \quad (14)$$

ここで、 $Y' = 0$ と仮定して式 (12)(13)(14) を用いて、 $W'(X', Y', Z')$ は式 (16) で求めることができる。

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} \frac{X - t_X}{Z - t_Z} t'_Y + t'_X \\ 0 \\ \frac{Y - t_Y}{Z - t_Z} t'_Y + t'_Z \end{bmatrix} \quad (15)$$

最後に、この点 $W(X', Y', Z')$ が投影される画像上の点 $P'_1(u'_1, v'_1)$ は、式 (9)(10) と同様にして式 (16) のように表すことができる。

$$\begin{bmatrix} u'_1 \\ v'_1 \end{bmatrix} = \begin{bmatrix} f & 0 & u_0 \\ 0 & -f & v_0 \end{bmatrix} \begin{bmatrix} x'_1 \\ y'_1 \\ 1 \end{bmatrix} = \begin{bmatrix} f \frac{X - t_X}{Y - t_Y} + u_0 \\ f \frac{Z - t_Z}{Y - t_Y} + v_0 \end{bmatrix} \quad (16)$$

同様に、カメラが Z 軸の向きに距離 d だけ直進した位置 $O_2(t_X, t_Y, t_Z + d)$ で撮影した場合に、同じ点 $W(X, Y, Z)$ が射影変換画像上に投影される点 $P'_2(u'_2, v'_2)$ は、式 (17) のように求めることができる。

$$\begin{bmatrix} u'_2 \\ v'_2 \end{bmatrix} = \begin{bmatrix} f \frac{X - t_X}{Y - t_Y} + u_0 \\ f \frac{Z - (t_Z + d)}{Y - t_Y} + v_0 \end{bmatrix} \quad (17)$$

ここで、求めた $P'_1(u'_1, v'_1)$, $P'_2(u_2, v'_2)$ の変化を見るために、2点の差分をとると、式(18)(19)のようになる。

$$\begin{cases} u'_2 - u'_1 = 0 \\ v'_2 - v'_1 = \frac{fd}{t_Y - Y} \end{cases} \quad (18)$$

$$(19)$$

このことから、カメラが直進する場合に限り、移動前後で射影変換画像の u 軸方向に変化は無いが、 v 軸方向には変化が見られる。そこで、 v 軸方向の変化にのみ着目する。式(19)で表される2画像間の v 軸方向のずれ量を $V = v'_2 - v'_1$ として式(20)のように定義する。なお、障害物の高さがカメラの光軸よりも小さいものを仮定しているため、 $t_Y - Y > 0$ となる。

$$V = \frac{fd}{t_Y - Y} \quad (20)$$

前章で考察した、2枚の射影変換後の画像の物体の形状

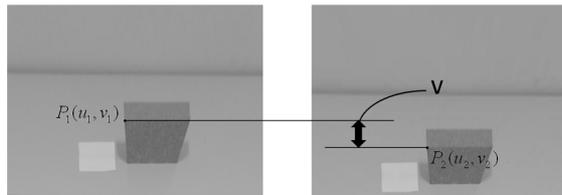


図4 ずれ量 V

の変化は高さに関係しているということが、この式から確認できる。焦点距離 f 、カメラが直進した距離 d 、カメラの高さ t_Y は既知であるため、この v 軸方向のずれ量 V は点 W の高さ Y にのみ依存している。式(20)を変形すると式(21)のように表すことができる。

$$Y = t_Y - \frac{fd}{V} \quad (21)$$

これは逆に言えば、点 W の高さ Y が v 軸方向のずれ量 V にのみ依存しているということであり、画像からずれ量 V を抽出することで、点 W の高さ Y を求めることができる。

5. テンプレートマッチングを用いた対応点探索による差分抽出

高さ Y を復元するために、テンプレートマッチングを用いた対応点探索により v 軸方向のずれ量 V を求める。2枚のカメラ画像を I_1, I_2 とすると、 I_1 の一部をテンプレートとして SAD によって I_2 上の対応点を探索し、2画像間のずれ量を求める。求めたを式(21)の V に代入し、高さ情報 Y を計算する。

ここで、 I_1 と I_2 ではカメラと障害物との距離が変化

しているため、画像中の障害物の場所も変化している。この変化は高さがゼロである部分の変化であり、式(22)において $Y = 0$ とすることによって求められる。よって、この変化を D として式(4.17)で表すことができる。

$$D = \frac{fd}{t_Y} \quad (22)$$

この変化 D を用いて、 I_1 の画素 (x, y) におけるテンプレート T を I_2 上の画素 $(x, y + D)$ を中心とした探索範囲内で動かすことによって、テンプレートマッチングにおいて I_2 全体を走査する必要はなくなり、対応点の探索時間が短縮される。

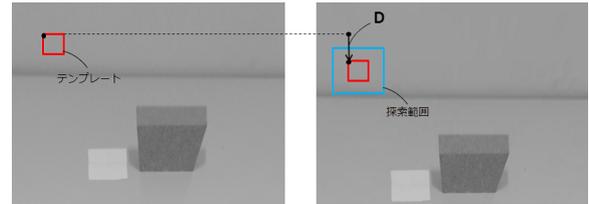


図5 テンプレートマッチング

6. 実験

以上で述べた上面視画像を用いた障害物の高さ情報推定的手法を実画像に適用することで、障害物の高さ情報を推定し、これを評価する。ここで、今回は画像と高さとの関係式(21)において fd を定数として相対値を求める。実験に用いる画像は、路上に直方体と白線が存在するシーン、路上に駐車用ブロックと白線が存在するシーンの2つである。いずれも、カメラの光軸方向に直進するカメラで撮影した2枚の画像を用いる。撮影するカメラの光軸は地面と平行になるように設定する。得られた2枚の画像を上面視画像に射影変換し、提案手法を用いて高さ情報を推定した結果を示す。なお、対応点探索の際のテンプレートサイズと探索範囲によって画像の端の画素に関してはずれ量が求められず高さ情報を得ることができない画素がある。

6.1 路上に直方体と白線が存在するシーン

路上に直方体と白線が存在するシーンに対して、提案手法を適用する。

実験に用いた2枚の画像を図6(a)(b)に示し、それらを上面視画像に変換したものを図7(a)(b)に示す。照度変化の影響を低減するため、濃淡画像のエッジ画像(図8)を用いる。エッジ画像を用いて対応点探索を行い差分を抽出したものを図9に、その差分から高さ情報を推定したものを図10に示す。なお、対応点探索の際のテンプレートサイズは 20×20 ピクセル ($T=20$)、 30×30 ピクセル ($T=30$) の二通りで実験を行った。

図9の対応点探索によるずれ量の結果を見ると、直方体の上面部分をピークに前面部分は地面に向かうにつれ



図 6 実 画 像

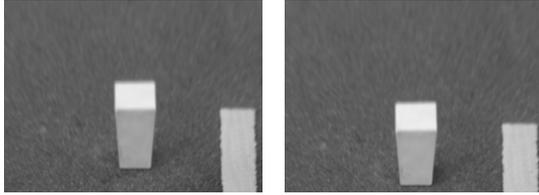


図 7 上面視画像

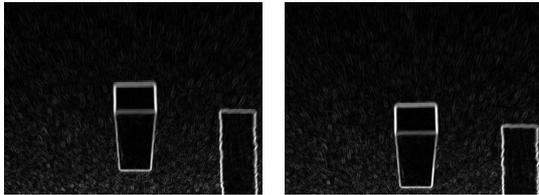


図 8 エッジ画像

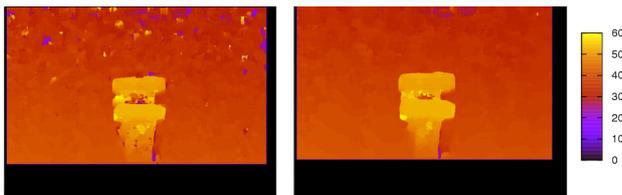


図 9 対応点探索で求めたずれ量 (左:T=20, 右:T=30)

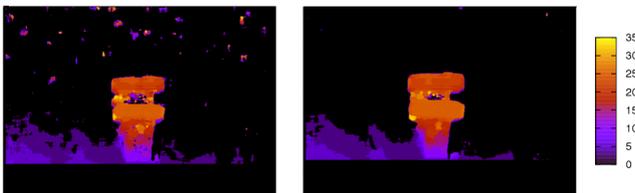


図 10 高さ情報の推定結果 (左:T=20, 右:T=30)

減少していることが分かる．この結果を式 (21) に代入し高さを推定した結果が図 10 であるので，上面部分の最大の高さから地面に向かうにつれ減少していることがわかる．上面部分に関しては，図 9，図 10 共にノイズが出ているが障害物の最大の高さの相対値は推定することが出来ている．ここで，ノイズは対応点探索での誤差が原因であると考えられる．障害物の上面，地面が平面であると仮定したとすると，対応点探索の際，周辺の画素の差分値に対して急激に値が変化するような画素の差分値はノイズであると推測できる．よって，周辺の画素の差分値を用いて補正をすることでずれ量のノイズを減らすことができ，高さを推定した結果においてもノイズを減らすことができると考えられる．また， $T=20$ と $T=30$

ではノイズの量が変化していることから，テンプレートのサイズを適切に設定することでもノイズを軽減できると考えられる．

6.2 路上に駐車用ブロックと白線が存在するシーン

路上に駐車用ブロックと白線が存在するシーンに対して，提案手法を適応する．ここで使用するブロックは，シーン 1 の直方体に比べ形状に曲線が含まれている，図 11 のような形状のブロックである．



図 11 駐車用ブロック

実験に用いた 2 枚の画像を図 12(a)(b) に示し，それらを上面視画像に変換したものを図 13(a)(b) に示す．このエッジ画像（図 14(a)(b)）を用い，差分を抽出した結果が図 15，差分を用いて高さ情報を推定したものを図 16 に示す．



図 12 実 画 像

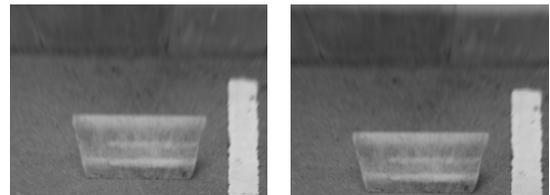


図 13 上面視画像

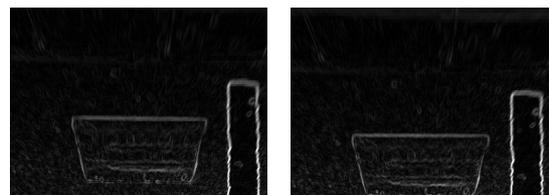


図 14 エッジ画像

図 15 の対応点探索によるずれ量の結果を見ると，前面部分のずれ量は駐車用ブロックの上辺部分から地面に

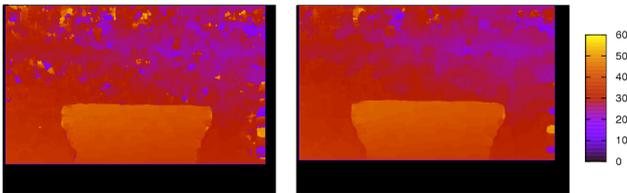


図 15 対応点探索で求めたずれ量 (左:T=20, 右:T=30)

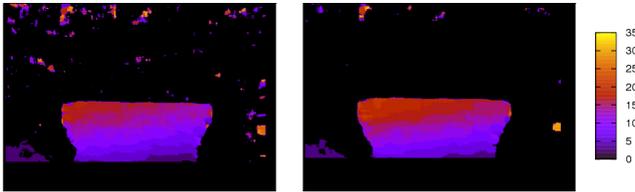


図 16 高さ情報の推定結果 (左:T=20, 右:T=30)

向かうにつれ減少していることが分かる．図 16 では図 15 のずれ量が反映され上辺の最大の高さから地面に向かって減少していることがわかる．直方体の場合と同様，地面の部分にノイズが多く出ているが，地面が平面であると仮定したとすると，対応点探索の際に周辺の画素のずれ量の値に対して急激に値が変化するような画素のずれ量はノイズであると推測できる．よって，周辺の画素の差分値を用いて補正をすることでずれ量のノイズを減らすことができ，高さを推定した結果においてもノイズを減らすことができると考えられる．また，直方体の場合と同様， $T=20$ と $T=30$ ではノイズの量が変化していることから，テンプレートのサイズを適切に設定することでノイズを軽減できると考えられる．

文 献

- [1] 国土交通省道路局，“国土交通省道路局 ITS ホームページ”
- [2] 二宮芳樹，“車載周辺のセンシング・認識技術”，システム/制御/情報, Vol.55, No.1, pp.8-13(2011)
- [3] C. Tomasi, T. Kanade, “Shape and motion from image streams under orthography: a factorization method”, International Journal of Computer Vision, 9(2):137-154(1992)
- [4] 斉藤佳明, 太田直哉, “オプティカルフローの再検出による高精度な形状復元”, 電気電子情報通信学会論文誌, Vol.J81-D-2, No.6, pp.1123-1131(1998)
- [5] S. Agarwal, N. Snavely, I. Simon, Steven M. Seitz, R. Szeliski, “Building Rome in a Day”, 12th IEEE International Conference on Computer Vision, pp.72-79(2009)
- [6] 伊東敏夫, “VB.NET で学ぶ画像処理アルゴリズム”, 広文社 (2002)
- [7] 奥富正敏 他, “デジタル画像処理”, CG-ARTS 協会 (2004)
- [8] B.K.P. Horn, “Robot Vision”, The MIT Press(1986)
- [9] 高木幹雄, 下田陽久, “新編 画像解析ハンドブック”, 東京大学出版会 (2004)
- [10] S. Benoit, F.P. Ferrie, “Towards Direct Recovery of Shape and Motion Parameters from Image Sequences”, 9th IEEE International Conference on Computer Vision, Vol.2, pp.1395-1402(2009)
- [11] 井上誠喜, 八木伸行, 林正樹, 中須英輔, 三谷公二, 奥井

- [12] 誠人, “C 言語で学ぶ実践画像処理”, オーム社 (1999)
立松直倫, 大谷淳, “移動ステレオカメラ画像中からの移動物体の抽出と三次元形状復元およびカメラの姿勢推定を行う Temporal Modified-RANSAC に基づく手法の検討”, 信学技報 *IECE Technical Report CS2010-60, IE2010-99(2010-12)*(2010)