

# 全天周画像道路地図登録および 交差点における接近車両検出

2011年1月

海 鷹

鳥取大学大学院 工学研究科  
博士後期課程 情報エレクトロニクス 専攻

# 目 次

第一章 序論.....	1
1.1 本論文の背景 .....	1
1.2 本論文の目的 .....	2
1.3 本論文の構成 .....	3
第二章 一対の魚眼カメラによる全天周画像の獲得 .....	7
2.1 球面投影モデル.....	7
2.2 魚眼画像と球面画像の対応 .....	10
2.3 全天周画像の作成.....	11
第三章 全天周画像道路地図登録のための同期問題 .....	17
3.1 はじめに.....	17
3.2 関連研究.....	18
3.3 一対の魚眼カメラからなる全天周カメラの同期.....	20
3.3.1 同期手法.....	20
3.2.2 カメラの運動推定 .....	21
3.4 慣性センサーとの同期及び傾きの補正 .....	22
3.4.1 慣性センサーと全天周カメラの相対姿勢の推定.....	23
3.4.2 慣性センサーの同期 .....	24
3.4.3 全天周画像の傾き補正.....	24
3.5 GPS データの同期.....	25
3.6 実験と結果.....	26
3.7 まとめ .....	32
第四章 全天周画像の表現法.....	35
4.1 はじめに.....	35
4.2 関連研究.....	37
4.3 二次元配列球面画像フォーマット .....	39
4.3.1 SCVT 画像の二次元配列.....	39
4.3.2 隣接頂点算出 .....	40
4.3.3 球面座標と SCVT 配列要素の間の写像 .....	43
4.3.4 二次元矩形球面画像フォーマット .....	44
4.4 SIF 画像の透視画像表示 .....	46
4.5 実験.....	47
4.6 まとめ .....	48
第五章 バイナリ運動履歴画像による交差点における側面からの接近車両の検出 .....	51

5.1	はじめに.....	51
5.2	関連研究.....	52
5.3	バイナリ運動履歴画像の生成.....	54
5.4	側面から接近車両の検出.....	55
5.4.1	背景と側面から接近車両の分離.....	55
5.4.2	運動領域分割と運動方向推定.....	57
5.3.3	実験.....	59
5.4	まとめ.....	60
第六章	結論.....	63

## 目 次

図 2.1	ピンホールカメラモデルの透視投影	7
図 2.2	球面投影モデルの球面投影	8
図 2.3	等距離射影方式	9
図 2.4	(a) ミラーを利用した全方位視覚センサー (b) 20 個の広角カメラによる全天周カメラ (c) Google ストリートビューツールカメラ (d) 本研究の全天周カメラ	12
図 2.5	二つの魚眼画像から全天周画像を作成する流れ	13
図 2.6	(a) 作成した全天周画像球面画像 (b) 全天周画像から作成した六面透視画像	14
図 3.1	全天周カメラ, 慣性センサーとハイブリッド GPS 受信機を車に搭載された様子	18
図 3.2	全天周カメラ(下)と慣性センサー(上)の座標関係	23
図 3.3	慣性センサーの角度データ列を 10 から 90 レームの範囲で平滑化した結果	25
図 3.4	GPS センサーの方位角データのノイズを除去した結果	26
図 3.5	魚眼カメラの隣接している二つのフレームのサンプル画像. 上の点はカメラの運動算出に抽出した特徴点になる	27
図 3.6	(a)と(b)はそれぞれ二つの魚眼カメラの画像列から推定した回転角度ヨー(y)データ列. (c)は慣性センサーの角度ヨーのデータ列 (d)は処理した GPS 方位角データ列	27
図 3.7	(a)魚眼カメラのピッチ(x), ヨー(y)と ロール(z)角度の相関係数の計算結果(b) 3つの相関係数の乗積で表す回転全体の相関係数の結果.	28
図 3.8	二つ魚眼カメラを同期して作成した全天周画像. (a)同期前の合成全天周画像 (b)同期後の全天周画像	29
図 3.9	(a)魚眼カメラと慣性センサーのピッチ(x), ヨー(y)と ロール(z)角度の相関係数の計算結果(b) 3つの相関係数の乗積で表す回転全体の相関係数の結果	30
図 3.10	魚眼全天周画像の歪みの補正結果	31
図 3.11	(a)GPS センサーの方位角と全天周画像列のヨー角の相関係数の計算結果 (b) 全天周画像列をリサンプリングした結果	31
図 3.12	Google マップでクリックした位置の全天周画像のサンプル	32
図 4.1	魚眼カメラから作成する全天周画像の余分領域と重なる領域	35
図 4.2	正 20 面体からの測地ドーム分割過程	36
図 4.3	(a) 経緯度を表した球面画像と展開された経緯マップ (b) 球面画像とキューブマップ	37
図 4.4	SCVT 画像を 5 つの平行四辺形として表すデータ構造	38
図 4.5	測地ドーム分割で得られた頂点からなる 2 次元 SCVT 配列	39

図 4.6 コンパクト 2D 矩形配列で表す球面画像フォーマット SIF の構成する手順.....	45
図 4.7 SIF で表現した全天周画像 .....	46
図 4.8 透視画像表示と魚眼画像との関係 .....	47
図 4.9 (a)従来手法で作成した 6 面の透視画像(b)提案手法で SIF 画像から生成された 6 面の透視画像 .....	49
図 5.1 本研究で解決問題の略図.....	52
図 5.2 トヨタ社のフロントモニターの映像.....	52
図 5.3 (a)運動履歴画像(MHI). (b) 時間的運動履歴画像 (tMHI). (c) 提案手法: バイ ナリ運動履歴画像(BMHI). 以上すべての運動履歴画像画像は連続 16 フレー ムの差分画像から構成されている. ....	53
図 5.4 側面から接近車両を検出手法.....	56
図 5.5 本手法の処理流れ.....	56
図 5.6 (a) BMHI 画像 (b) BMHI 画像の領域分割結果.....	57
図 5.7 分割領域中心をビット分けて計算した結果.....	58
図 5.8 運動領域分割と推定方向を比べた結果 .....	59

## 表 目 次

表 4.1 SCVT 配列の接続点位置.....	42
表 4.2 処理時間を比べた結果.....	48
表 5.1 $D(x,y,t)$ , $tMHI$ および $BMHI$ のそれぞれの特徴.....	55
表 5.2 処理時間を比べた結果.....	60

# 第一章 序論

## 1.1 本論文の背景

近年、車両にはカメラ、レーダなどの様々なセンサーが搭載され、車の周りの環境認識や車の状態把握など運転支援システム[1-4]が実用化されている。車載カメラはレーダなどほかのセンサーより情報量が多く、周りの障害物、運動物体、歩行者、車、車線など様々な情報を認識して分析することができる。特に、魚眼カメラはその180度以上という広い画角から、車両の周辺監視をしやすく、広く用いられている撮影装置である。これにより少ない台数の魚眼カメラで周囲の環境情報を獲得することが可能である。

車載カメラ映像から運転を支援するために人々に分かりやすい道路地図の提供が重要課題の一つである。一般的な道路地図では線を使って平面幾何形状で路線情報を表示しているが、人間の主な認識は現地の視覚情報である。例えばカーナビゲーションなどではコンピュータグラフィック (CG) により作成した映像で現地視覚情報を表現している。しかし、現実感の高い仮想空間を生成するのが困難である。したがって高度な地図情報に埋め込まれた人間に分かり易い道路地図を構築する必要がある。例として Google マップ ストリートビューツールは図 2.4 (c) のような全天周カメラから作成した仮想空間画像で道路地図を表示する。しかし、この地図登録作業の詳細が公開されていない。また、ユーザーが自分の車に搭載されているカメラで簡単に地図登録できれば、それぞれ個人のニーズに合わせた道路地図を作製できる。そのため、道路地図登録の自動化が重要である。本論文では、全天周カメラ、慣性センサー、GPS を用いた全天周画像道路地図登録の手法を提案する。

コンピュータグラフィックの代わりに、2つの反対向きの魚眼カメラから合成した全天周画像で現実感の高い仮想空間を実現できる。大規模道路地図を構築するため、カメラ映像、GPS、慣性センサーなど様々なセンサーの同期処理が必要となる。専門の装置で同期したデータを得ることはできるが、実験コストが高いまたは装置の変更が困難であり、それぞれ独立に記憶されると複数種類のセンサーを同期する必要がある。同種類センサーの間は多くの共通情報があり、それを利用して同期できる。例えば複数カメラ間の共通画像領域を通じて同期できる。異種類のセンサーの場合に新しい同期方法が必要となる。

一方、全天周画像を構成する魚眼画像は、撮像カメラの内部パラメータと外部パラメータを用いて球面画像に写像することができる。全天周画像の保存管理の際には撮像された魚眼画像の保存と全天周画像を球面に写像して得る球面画像を有効なデータ構造で表現されて保存する2つの方法がある。撮像魚眼画像の保存には画像以外にカメラパラメータが

必要となり、画像の間にも多くの冗長データが入っているため全天周画像の保存、転送、処理には不便である。球面画像を保存するには有効なデータ構造で球面三次元情報を表現できる球面画像フォーマットが必要である。例えば、球面画像データを階層的な木構造で表す手法が提案されているが、頂点の隣接情報が失われる欠点がある。また、球面画像データを三次元配列データ構造で表す手法も提案されているが、球面画像の画素の探索は3次元のデータ空間で行う必要である。

車載カメラ映像による運転支援におけるもう一つの重要な課題は環境の認識である。交差点における多くの交通事故は、運転者が側面から接近する車両に十分に注意を払わなかったために起こっている。狭い道路から信号がない交差点に入る時、運転者は車をゆっくり移動しながら側面から接近する車両を注意する必要がある。特に周りに建物がある場合に、側面からの接近車両を正確に把握するのが困難である。一方、自車体の前方に魚眼カメラを取り付ければ、その広い視野で交差点の状況を観測できる。その映像から高速かつ高精度な側面から接近する車両を検知できれば、交通事故の減少につながる。動物体の検出には、運動履歴画像がよく用いられている。しかし、従来の運動履歴画像は、前時刻の運動情報が後の時刻の運動情報に上書きされる可能性がある。また、交差点における接近車両を検出する際には、自車両もゆっくり交差点に進入する可能性があるために、接近車両を複雑な背景運動から検出する必要がある。

本論文では、全天周画像道路地図登録及び交差点における接近車両検出に着目して、研究を行っている。

## 1.2 本論文の目的

本論文で車載魚眼カメラを用いて運転支援を目指して、全天周画像道路地図登録及び交差点における接近車両検出に関する手法を提案する。全天周画像道路地図登録を実現するためには以下の課題が挙げられる。

- 独立に撮られた魚眼カメラ画像列、GPS、慣性センサーデータの同期

本論文では車に搭載した、それぞれ独立に記憶される魚眼画像列と慣性センサーおよびGPSデータを運動情報により同期する手法を提案する。複数のセンサーは同じ車に搭載しているため同じ運動情報を持ち、画像からカメラの運動を推定により同期する。

- 全天周画像の表現法

本論文で測地ドーム構造に基づくコンパクトな二次元配列の球面画像のフォーマットを提案する。その球面画像のフォーマットにより冗長データをなくし、二次元配列として、球面画像を扱うことができる。



もう一つの目標は自車体前方の魚眼カメラ映像から交差点での側面からの車両を検出である。これを実現するためには以下の課題が挙げられる。

- 動物体の実時間検出

動物体検出には様々なアルゴリズムが提案されているが、運転補助にはアルゴリズムの早い処理速度と検出精度が要求される。本論文では従来の運動履歴画像に起こる運動領域消失問題を克服できるバイナリ運動履歴画像と呼ばれる動物体検出手法を提案する。提案手法では、運動履歴情報を完全に保存でき、更にビットシフト演算で高速処理を実現できるために、従来法より有効かつ高速に運動を解析できる。

- 複雑な背景運動から側面接近車両の検出

魚眼画像で自車両と接近する車両の運動方向が反対である特徴を利用して側面からの接近車両検出手法を提案する。

### 1.3 本論文の構成

本論文の構成は次の通りである。

第一章 序論

第二章 一对の魚眼カメラによる全天周画像の獲得

第三章 全天周画像道路地図登録のための同期問題

第四章 全天周画像の表現法

第五章 バイナリ運動履歴画像による交差点における側面からの接近車両の検出

第六章 結論

**第一章**は序論であり、本研究の背景と目的について述べる。

**第二章**では本論文で用いる魚眼カメラの球面投影モデルと魚眼画像から全天周球面画像を獲得する手法について述べる。まず、球面投影モデルと一般的な透視投影モデルの比較、魚眼レンズで撮影した画像と球面画像との対応関係について述べる。次に、画像歪みを含む魚眼カメラの内部パラメータ校正手法により反対向きの一对のそれぞれ半球視野をもつ魚眼カメラの画像を接合するというシンプルなアイデアに基づき全天周球面画像を獲得する手法について述べる。

**第三章**では全天周画像道路地図作成のために提案した同期手法について述べる。まずは

関連研究について述べる．次に二つの画像列からカメラの運動情報を推定してその二つの画像列の同期時刻を推定する手法について述べる．次に慣性センサーの回転角速度より全天周画像列との同期について述べる．後に GPS の方位角度により全天周画像列との同期し、リサンプリングして地図への登録について述べる．最後に実験と結果をまとめる．

**第四章**では提案した全天周球面画像のフォーマットについて述べる．はじめに球面画像フォーマットについての関連研究を紹介する．次に提案する矩形二次元配列で表現した球面画像フォーマットの構成及び簡素化隣接点を探すアルゴリズムについて述べる．さらに、提案される球面画像フォーマットにより全天周画像から透視画像を高速に生成するについて述べる．最後は実験結果について述べる．

**第五章**では提案バイナリ運動履歴画像による交差点における側面からの接近車両の検出手法について述べる．まず、従来の運動履歴画像と比べてバイナリ運動履歴画像を説明する．次に提案手法により画像上の運動領域を検出して運動方向を推定するアルゴリズムについて述べる．次に運動領域の運動方向と自車体の運動より生じる背景運動方向を比べて側面からの接近車両の領域を決める手法について述べる．最後は従来の運動履歴画像と比べた実験と結果について述べる．

**第六章**はまとめである．以上の研究を総括し、残された課題について述べる．

## 参考文献

- [1] <http://www.honda.co.jp/tech/auto/mvcs/>
- [2] [http://www2.toyota.co.jp/jp/tech/safety/technologies/active/cornering\\_assist.html](http://www2.toyota.co.jp/jp/tech/safety/technologies/active/cornering_assist.html)
- [3] <http://www2.toyota.co.jp/jp/tech/safety/technologies/active/monitoring.html>
- [4] <http://www2.toyota.co.jp/jp/tech/safety/technologies/active/radar.html>

## 第二章 一対の魚眼カメラによる全天周画像の獲得

全天周画像は車の周囲の情報を死角なく獲得するために有効な手段である。本論文では、一対の 180 度以上の視野を持つ魚眼カメラを用いて前後 2 方向の画像を接合するというシンプルなアイデアに基づき全天周画像を獲得する。本章では球面投影モデルに基づいて魚眼画像から全天周画像[1-3]を獲得する手法について述べる。

### 2.1 球面投影モデル

通常の平面画像は、レンズへの入射光を平面スクリーンに透視投影させて得られるイメージであり、その透視投影は、一般的に図 2.1 に示すようなピンホールカメラモデルにより説明される。カメラ座標系の原点をピンホールに置き、その  $z$  軸が光軸と重なる。シーン内の空間点  $P$  からの光線はピンホールを通り画像平面上の位置  $P'$  に像を結ぶ。ここで画像平面とピンホールとの距離をカメラの焦点距離  $f$  (focal length)、光軸と画像平面との交点  $c$  を光軸点(optical center)または主点(principal point)という。

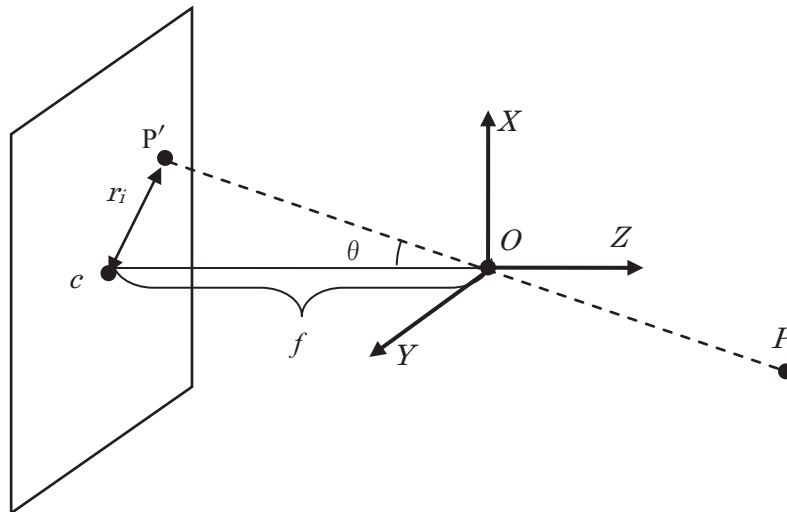


図 2.1 ピンホールカメラモデルの透視投影

このピンホールカメラによる射影は中心射影，または透視投影モデルと呼ばれ，カメラの射影を説明するために一般的に用いられており，従来のカメラ校正手法の多くはこの射影モデルに基づいている．この射影モデルは一般的な狭い視野のカメラを説明するには十分であるが，180度以上の視野を持つ魚眼カメラの視野全体の情報を扱うことができない．

一方，球面画像[4]は，レンズへの入射光を，焦点を中心とする球面状の画像面上に投影させることで得られるイメージであり，全天周球面画像は撮影地点から見えている周囲全ての点を同一球面上に投影することで得られるイメージである．球面画像は投射方向に対して等方性をもち，全ての方向の情報を一様に扱うことができる全天周球面画像は，周囲を取り巻く環境を表現するのに適している．特に環境マップへの利用の観点では，全天周球面画像は撮影地点に対して周囲の環境から放射されてきている光を全方向について記録したものとみなすことができるため，全天周球面画像を獲得することは周囲の光環境を獲得することに繋がる．球面投影モデルに基づく投影によって得られる画像は球面画像と呼ばれる．

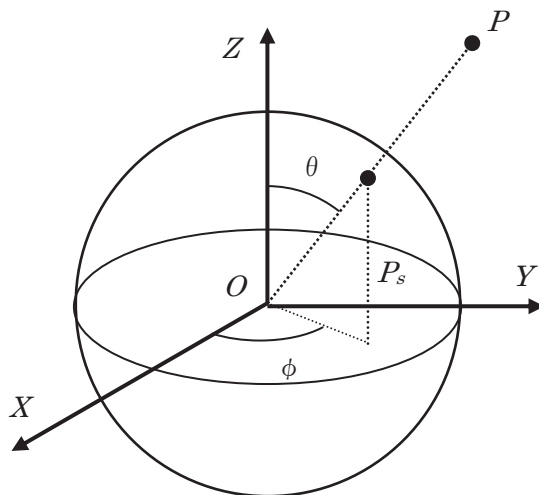


図 2.2 球面投影モデルの球面投影

球面投影は，図 2.2 に示すように，空間に半径  $f$  の球と空間点  $P$  があるとき， $P$  と球の中心を結ぶ直線が球面と交わる点  $P_s$  を，点  $P$  の球面上への投影点とする投影モデルである．原点を球の中心においたカメラ座標系において，三次元空間上の点  $P$  の座標が

$$P = [X_c \quad Y_c \quad Z_c]^T \quad (2.1)$$

であるとき、この点の球面画像への投影点  $P_s$  の座標は、投影光線の方角を表す2つの角度、天頂角  $\theta$  と方位角  $\phi$  を用いて

$$P_s = [f \sin \theta \cos \phi \quad f \sin \theta \sin \phi \quad f \cos \theta]^T \quad (2.2)$$

と書けるので、シーンの点  $P$  とその投影点  $P_s$  との関係は、

$$\begin{aligned} P_s &= [f \sin \theta \cos \phi \quad f \sin \theta \sin \phi \quad f \cos \theta]^T \\ &= \frac{f}{\sqrt{X_c^2 + Y_c^2 + Z_c^2}} [X_c \quad Y_c \quad Z_c]^T \end{aligned} \quad (2.3)$$

と表すことができる。すなわち、 $P_s$  はスケールファクタを除けば  $P$  に等しい。

実際に球面画像を扱う際には、一般に有益な情報となるのは空間点  $P$  の方向を表す球面座標  $(\theta, \phi)$  であり、投影面として想定した球の半径には興味が無い。そこで、投影球面の半径を  $f=1$  とし、単位球への投影である正規化座標

$$P_s = [\sin \theta \cos \phi \quad \sin \theta \sin \phi \quad \cos \theta]^T \quad (2.4)$$

を普通は用いる。このように球面投影は、ある方向から入射してきた光の投影点を、その方向  $\theta$  と  $\phi$  を用いて表す素直なモデルである。

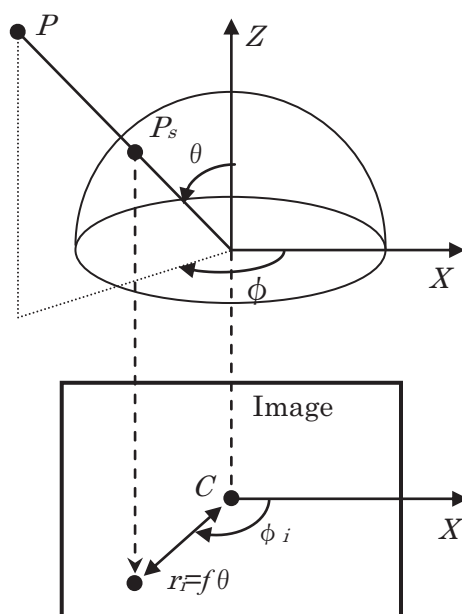


図 2.3 等距離射影方式

## 2.2 魚眼画像と球面画像の対応

魚眼カメラで撮影した画像から球面画像を獲得するためには、撮影魚眼画像上の位置  $P_i$  と球面画像上の位置  $P_s$  との対応を知る必要がある。この対応はレンズの射影式から得られる。魚眼レンズの射影式は4種類[5]あるが、以下に魚眼レンズの最も一般的な射影方式である等距離射影(equidistance projection)の方式による例を示す(図 2.3)。シーンからレンズへの投射線の  $(\theta, \phi)$  と、画像上で位置  $P_i$  の像高  $r_i$ , 基準軸(通常は  $x$  座標軸)からの角度  $\phi_i$  によって記述できる。

$$\begin{bmatrix} r_i \\ \phi_i \end{bmatrix} = \begin{bmatrix} f & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \theta \\ \phi \end{bmatrix} \quad (2.5)$$

ここで  $f$  はレンズの焦点距離である。撮影画像上での点  $(x_i, y_i)$  と球面画像上の位置  $P_s$  の座標の対応の場合、魚眼カメラの光軸点を  $(x_c, y_c)$  すると  $(r_i, \phi_i)$  は

$$\begin{aligned} r_i &= \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} \\ \phi_i &= \arctan(y_i - y_c, x_i - x_c) \end{aligned} \quad (2.6)$$

球面画像座標  $P_s$  は

$$P_s = \begin{bmatrix} \sin \frac{r_i}{f} \cos \phi_i & \sin \frac{r_i}{f} \sin \phi_i & \cos \frac{r_i}{f} \end{bmatrix} \quad (2.7)$$

として得られる。

実際のレンズは製造過程で発生する誤差などの影響のために、その投影は理想的な射影式に正確には従わない。文献 [6]ではレンズ歪みを補正する魚眼カメラの内部パラメータの推定手法を提案している。これは、レンズ歪みを補正するに放射方向歪曲収差、光軸点のずれ、接線方向歪曲収差を考慮されている。

- 放射方向歪曲収差 (radial distortion)

理想的な射影式で投射角  $\theta$  に対して線形や三角関数で表されており、これをテイラー展開し、テイラー展開式の各項の係数  $k_1, k_3, k_5$  を用いて放射方向歪曲収差と呼ばれる放射方向に沿う歪をモデル化している。

$$r \approx k_1 \theta + k_3 \theta^3 + k_5 \theta^5 \quad (2.8)$$

- 光軸点のずれ

実際のレンズの光軸は正確に画像中心を通らず、多少ずれている。撮影画像での単位距離あたりの  $x$  方向,  $y$  方向の画素数の比  $\alpha$  で光軸点のずれを表すことで式 (2.6) を

$$\begin{aligned} r_i &= \sqrt{\alpha^2(x_i - x_c)^2 + (y_i - y_c)^2} \\ \phi_i &= \arctan(y_i - y_c, \alpha(x_i - x_c)) \end{aligned} \quad (2.9)$$

として書き直すことができる。

- 接線方向歪曲収差 (decentering distortion)

接線方向歪曲収差は光軸が画像平面に直交しないことに起因する、放射方向歪曲収差と垂直な方向に作用する歪である。極座標に式

$$\begin{aligned} \Delta T_r(p_i) &\approx 3P_1 r_i^2 \cos(\phi_i) + 3P_2 r_i^2 \sin(\phi_i) \\ \Delta T_\phi(p_i) &\approx -P_1 r_i \sin(\phi_i) + P_2 r_i \cos(\phi_i) \end{aligned} \quad (2.10)$$

で表す。

上記の 3 種類の歪をまとめた魚眼カメラの内部パラメータは  $x_c, y_c, \alpha, k_1, k_3, k_5, P_1, P_2$  であり, 球面座標 ( $\theta, \phi$ ) と, 画像上で位置を以下の式で表す。

$$\begin{aligned} k_1 \theta + k_3 \theta^3 + k_5 \theta^5 &= r_i + \Delta T_r(p_i) \\ \phi &= \phi_i + \Delta T_\phi(p_i) \end{aligned} \quad (2.11)$$

これらの魚眼カメラの内部パラメータより半球以上の視野を持つ魚眼レンズの歪みの補正を行う。

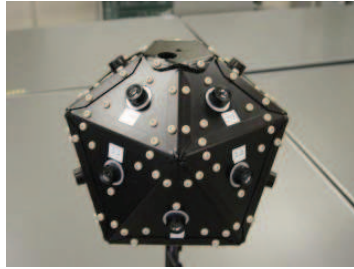
### 2.3 全天周画像の作成

全天周画像は, 図 2.4 で示したようにミラーを利用した全方位視覚センサー[7,8], 互いに視野を補い合う複数のカメラ[9,10]などさまざまな装置から獲得することができる。本論文で反対向き的一对の 180 度以上の視野を持つ魚眼カメラの画像を接合するというシンプルなアイデアに基づき全天周球面画像を獲得する。この手法はカメラの個数が少なく, システム構成コストが低い。





(a)



(b)



(c)



(d)

図 2.4 (a) ミラーを利用した全方位視覚センサー (b) 20 個の広角カメラによる全天周カメラ (c) Google ストリートビューツールカメラ (d) 本研究の全天周カメラ

一つの魚眼カメラから球面投影により魚眼画像を球面上にマッピングして半球画像を得ることができる。反対向きの二つの半球画像を接合すると一つの球面画像になる。車の上に載せた二つの魚眼カメラから全天周カメラ作成する流れを図 2.5 で示している。まずはカメラから二枚の魚眼画像を撮り、2.2 節で説明した手法により二枚の魚眼画像をそれぞれ球面にマッピングして二つの半球画像を得る。一つの半球画像を 180 度回転してからもう一つの半球画像と合わせて球面画像を得ることができる。図 2.6 では全天周画像から作成した立方体の六つの面への透視画像を示す。

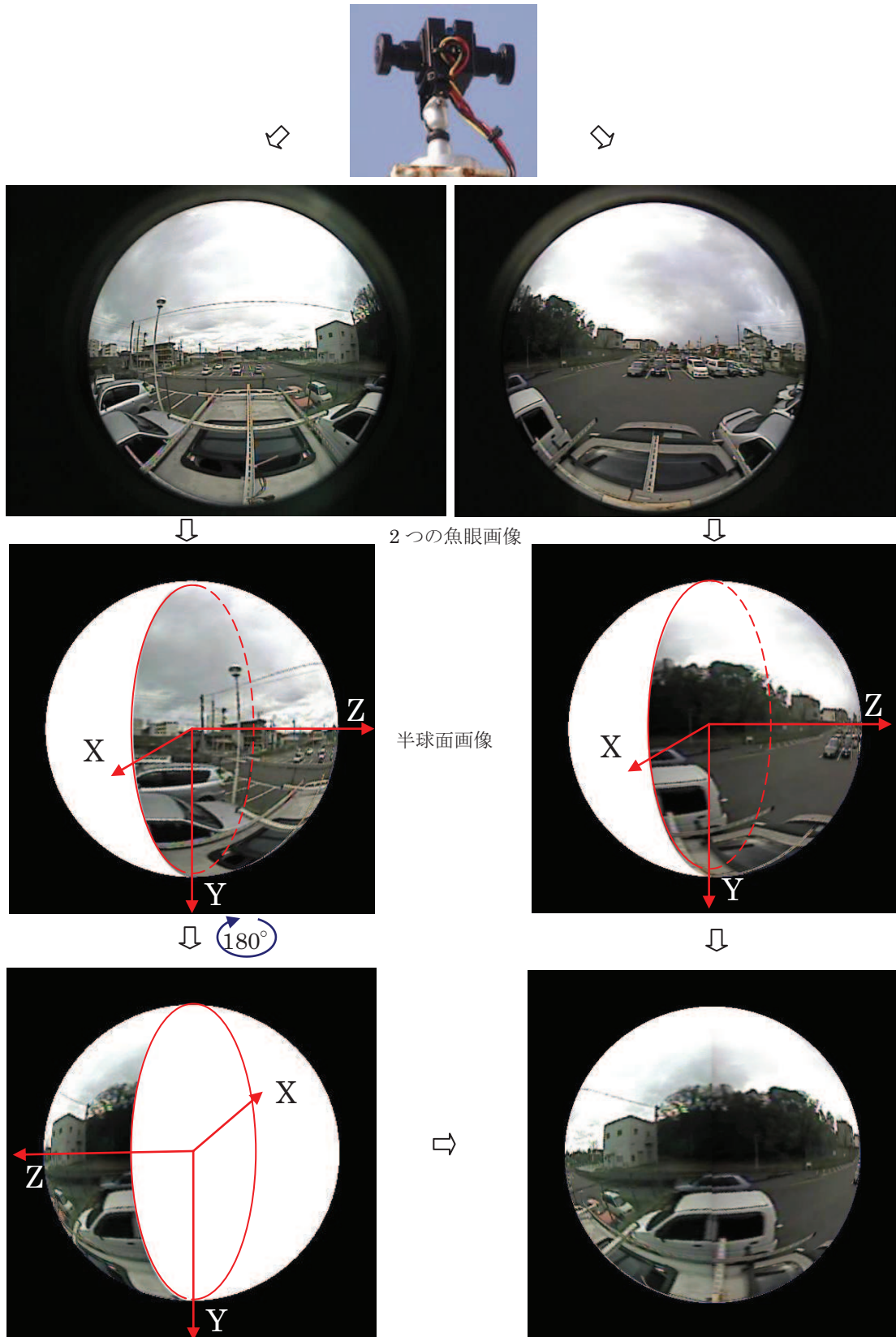
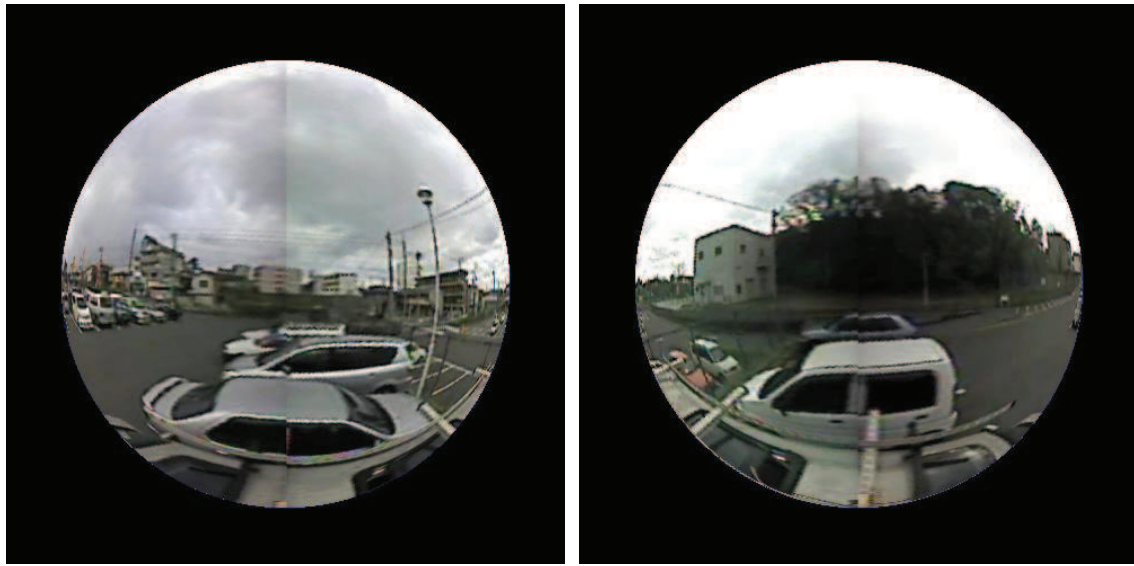


図 2.5 二つの魚眼画像から全天周画像を作成する流れ



(a)



(b)

図 2.6 (a) 作成した全天周画像球面画像 (b) 全天周画像から作成した六面透視画像

## 参考文献

- [1] S. Li, M. Nakano, N. Chiba, "Acquisition of spherical image by using fish-eye conversion lens", IEEE VR Conf., 2004.
- [2] S. Li, "Full-view spherical image camera", International Conference on Pattern Recognition, 2006.
- [3] 中野誠士, 李仕剛, 千葉則茂, “球面画像獲得のための魚眼カメラの校正”, 電子情報通信学会論文誌, Vol. J88-D-II, No.9, pp. 1847-1856, 2005.
- [4] 木室義彦, 長田正, “球面六角形画像ピラミッドを用いた球面投影による全方位画像処理”, 日本ロボット学会誌, Vol.14, No.6, pp.809-819,1996.
- [5] K.Miyamoto, "Fish eye lens", J. of Optical Society of America, Vol.54, No.8, pp.1060-1061, 1964.
- [6] 中野誠士, 李仕剛, 千葉則茂, “球面モデルに基づく縞模様パターンを用いた魚眼カメラの校正”, 電子情報通信学会論文誌, Vol.J90-D, No.1, pp.73-82,2007.
- [7] S.Nayar, "Catadioptric omnidirectional camera", Proc. of Computer Vision and Pattern Recognition, pp.482-488, 1997.
- [8] S.B.Kang, "Catadioptric self-calibration", Proc. of Computer Vision and Pattern Recognition, pp.201-207, 2000.
- [9] R Swaminathan, and S.K.Nayer, "Nonmetric calibration of wide-angle lenses and polycameras", IEEE Trans. PAMI, Vol.22, No.10, 2000
- [10] 遠藤隆明, 谷川智洋, 広田光一, 広瀬通孝, “超多眼カメラによる全天周画像の再構成”, 情報処理学会論文誌, Vol.43m, No.SIG11(CVIM5), 2002.
- [11] Y. Amakai, S. Murayama, Y. Obi, H. Takano, N. Momono and K. Takanashi: J. Phys: Conf. Series 150 (2009) 2004

## 第三章 全天周画像道路地図登録のための同期問題

車載カメラより全天周画像を撮って道路地図に自動登録するには2つの魚眼画像が同時に撮る必要がある。言い換えれば画像列の同期が必要となる。しかしながら二つの魚眼カメラを反対向きに置くため共通の観測領域に基づく同期の手法は利用できない。本章では、共通の運動情報を利用して、それぞれ独立に記憶される魚眼画像列と慣性センサーおよび GPS データを同期する手法を提案する。その同期を通して時間的整合性の取れたデータを利用して、背中合わせの一对の魚眼カメラで撮像される全天周映像の獲得、坂道や傾斜などの路面状況により生じる全天周映像の傾きの補正、及び、移動速度によらない全天周映像の等距離サンプリングを実現することができる。

### 3.1 はじめに

ITS（高度道路交通システム）の研究において、人々に分かりやすい道路地図の提供が重要課題の一つである。一般的な道路地図では線を使って平面幾何形状で路線情報を表示しているが、人間の主な認識は現地の視覚情報である。例えばカーナビゲーションなどではコンピュータグラフィックにより作成した映像で現地視覚情報を表現している。しかし道路周りの情報がほとんど入っていないため正確な道案内や場所の同定を行うには困難である。したがって、現実感の高い仮想空間を生成し、高度な地図情報を埋め込まれた、人間に分かり易い道路地図を構築する必要がある。

大規模道路地図を構築するため、実用的な方法として、GPS (Global Positioning System) 情報を利用して沿路で撮られた画像を平面幾何道路地図に自動登録するのが挙げられる。より臨場感のある仮想空間を生成するために、全天周カメラを用意し、撮った映像の正しい位置が分かれば全天周映像道路地図を構築できる。

本研究では慣性センサーを取り付けた全天周カメラを車の上に固定し、方位情報付 GPS センサーを車内に搭載する（図 3.1）。この全天周カメラは二つの半球面以上の視野を持つ魚眼カメラで構成され、慣性センサーは3軸の角速度と回転角度を検知できる。それらのデータの同期がとれれば、全天周カメラの位置姿勢情報を獲得できる。ゆえに、全天周画像をデジタルマップで正確に登録するため、独立に記録される各センサーからのデータ列

を同期することは極めて重要である．具体的には，以下の同期を行う．

- 全天周画像列を生成するため，二つの魚眼カメラの画像列の同期．
- 全天周画像列と慣性センサーデータの同期．
- 全天周画像列と GPS データの同期．

本提案手法では運動情報を用いて同期を行う．二つの画像列に対し，二つのカメラを同じ座標系に配置していると仮定し，それぞれ球面ステレオ法でカメラ運動の推定を行う．カメラの振動は同一車の運動で生じるため，二つのカメラから推定される運動情報から高い相関値を探し出してその二つのカメラで撮像された画像列の同期位置を決定できる．その運動情報に基づいて慣性センサーおよび GPS との同期もできる．慣性センサーはカメラと同じ座標系に配置され，軸の角速度によりカメラと同期する．GPS は車の運動方向を示している方位角によりカメラとの同期ができる．



図 3.1 全天周カメラ，慣性センサーとハイブリッド GPS 受信機を車に搭載された様子

## 3.2 関連研究

地図に含まれる沿路視覚情報を自動的に生成する方法は盛んに研究されている．沿路で撮ったビデオデータを圧縮するためにパノラマ表現[1]と呼ばれるコンパクト表現方法が提案されている．パノラマ画像からランドマークを抽出し，経路を定性的に記述する認知地図[2]を生成することができる．交差点を含む 2 次元地図は，交差点がノードに，経路がアークに対応したグラフのような表現として構築できる[3]．ただし，そのアークに対応する経路パノラマ画像を関連付ける．さらに，GPS の情報と沿路パノラマ表現を合わせると

幾何学的情報と沿路パノラマ画像視覚情報を含む地図[4-6]を生成することができる。

しかし、パノラマ画像がカメラの運動軌跡を中心とした表現であるため、車が交差点で曲がる時に歪みが生じる。ほかには、距離センサー[7,8]やラインスキャンまたはステレオビジョン処理[9]などから獲得した三次元情報で沿路の三次元モデルを構築する手法も提案されている。しかし、現場を本物の質感ある正確な三次元モデルの道路景色を三次元的に再構築するのは容易ではない。全方位画像の歪み補正手法に関して、消失点やEPI分析[10]などの手法を提案されているが、坂道や傾斜などの路面の緩慢な変化で生じる歪みを補正するのが困難である。

Google マップ ストリートビューツールで示すように、GPS 情報を利用して、全天周画像を直接平面幾何地図に登録するのが有効な方法である。しかし、その画像の登録手法を具体的に示していない。本研究では、全天周画像センサーとGPSと慣性センサーからなるシステムを用いて、カメラの傾きも補正できる全天周画像をデジタル地図に自動登録する手法を提案する。具体的には、運動情報を用いて各センサーからのデータ列を同期して、二つの魚眼カメラから時間的整合性のとれた全天周画像列を獲得し、慣性センサーデータを用いて全天周画像の傾きを補正する。さらに、同期のとれたGPSデータにより全天周画像列を均等にリサンプリングすると、車の移動速度に影響されずデジタルマップ上に均一に登録することができる。

複数のカメラを同期する問題について重複視野を利用して固定カメラの時刻を同期する手法が提案されている[11,12]。文献[11]では、まず動物体を追跡してその軌跡からおおよその同期時刻の範囲を推定し、次に観測される特徴点に対して、ステレオ法のエピポーラ拘束条件に基づいて正確な同期時刻を決定する。文献[12]では、画像での変化を時間的特徴として用い、その時間的特徴の相関に基づいて複数のカメラで撮像された複数の画像列間の対応を行う。それらの先行研究に対して、本システムでは、以下の特徴が挙げられる。

- 本研究で用いられる全天周画像センサーは、背中合わせで取り付けられた二つの約半球視野の魚眼カメラからなっているので、その二つのカメラ間には重複視野がほとんどなく、また、車の移動中に撮像される画像列でカメラの同期を行うので、カメラが静止状態になっていない。そのため、前述の先行研究の手法を適用できない。
- 本研究では、同種類センサーの二つのカメラ間の同期だけではなく、異種類センサーのカメラ、ハイブリッドGPS、慣性センサー間の同期を行う。それらのセンサーが同一の車に搭載されていることに着目して、各センサーから推定される運動情報に基づいてセンサー間の同期を行う手法を提案する。

### 3.3 一対の魚眼カメラからなる全天周カメラの同期

まずは二つの魚眼カメラ画像列の同期手法について説明し、次は画像からカメラの運動を推定する手法について述べる。

#### 3.3.1 同期手法

本研究の全天周カメラは図 3.2（下部）で示され、二つの半球視野の魚眼カメラで構成されている。その二つの魚眼カメラの内部パラメータと外部パラメータは、文献[13]で提案されている手法により校正する。

二つのカメラは互いに前後に反対向きとなり、相対姿勢は行列  $R_{rf}$  により校正されたと仮定する。また、カメラ間の距離は非常に小さいので、二つのカメラの座標系の原点は同じであると仮定している。車が移動するとカメラの運動をそれぞれ推定することができる。画像からカメラの並進運動のスケールを定めないので、本論文では、推定される回転成分の変化から同期の時刻を決定する。推定された二つのカメラの回転をそれぞれ  $R_f$ 、 $R_r$  とすると次の式を得る。

$$R_f = R_{rf} R_r \quad (3.1)$$

魚眼カメラからの画像列は独立に記録するために、この間に  $\Delta t$  時間遅延があるとする。と、(3.1) 式を以下に書き換えることができる。

$$R_{f(t+\Delta t)} = R_{rf} R_{r(t)} \quad (3.2)$$

全天周カメラの同期するためには時間遅延  $\Delta t$  を決定すればよい。

二つのカメラから推定された回転データ列をそれぞれ  $R_{f(t)}$ 、 $R_{r(t-\Delta t)}$  ( $t=1, \dots, n$ ) とすると、この回転行列からピッチ (x)、ヨー (y) とロール (z) 角度を計算し、カメラごとに角度変化を表した三つのデータ列を作成することができる。

たとえば、カメラのピッチ角のデータ列をそれぞれ  $X_f(x_{f(1)}, x_{f(2)}, \dots, x_{f(n)})$ 、 $X_r(x_{r(1)}, x_{r(2)}, \dots, x_{r(n)})$  とすると、この  $X_f$ 、 $X_r$  の間の相関係数は

$$\rho_X = \frac{\text{cov}(X_f, X_r)}{\sigma_{X_f} \sigma_{X_r}} \quad (3.3)$$

である。ただし、 $\text{cov}(X_f, X_r)$  が  $X_f$  と  $X_r$  の共分散で、 $\sigma_{X_f}$  と  $\sigma_{X_r}$  が  $X_f$  と  $X_r$  の標準偏差



である。データ列間の時間遅延  $\Delta t$  を考えると、このピッチ角の  $X_{f(t+\Delta t)}$  と  $X_{r(t)}$  間の相関係数は

$$\rho_X(\Delta t) = \frac{\text{cov}(X_{f(t+\Delta t)}, X_{r(t)})}{\sigma_{X_{f(t+\Delta t)}} \sigma_{X_{r(t)}}} \quad (3.4)$$

になる。同じ方法でヨー角とロール角の相関係数  $\rho_Y(\Delta t)$  と  $\rho_Z(\Delta t)$  を計算できる。

本論文では3つの角度の相関係数の乗積を計算して、回転全体の相関度として定義する。その理由は、任意の回転運動はロール、ピッチ、ヨーの三つの角度で一意的に表せる。二つのカメラが同じ運動を行うとすれば、その二つのカメラから推定されるそれぞれのロール、ピッチ、ヨーの角度も同じである。よって、その3つの角度の相関係数の乗積を計算することにより、最大値1である回転全体の相関度を得る。この回転相関関数  $Corr(t)$  は次の式で得られる。

$$Corr(t) = \rho_X(\Delta t) \rho_Y(\Delta t) \rho_Z(\Delta t) \quad (3.5)$$

最後に、同期時刻はこの回転相関関数の最大値により決定できる。

$$\arg\{\max_{\tilde{t}} \{Corr(\tilde{t})\}\} \quad (3.6)$$

次の節で画像列から魚眼カメラの回転の推定手法について述べる。

### 3.2.2 カメラの運動推定

魚眼カメラは半球視野を持つため、従来のピンホールカメラモデルを直接適用することができない。本研究では、魚眼カメラの運動を推定するため、球面カメラモデル[13]を使って魚眼画像の点を球面画像変換して計算している。

魚眼カメラの映像が 30 fps で録画されるため隣接している二つのフレームの間に車の運動は非常に小さくなる。そのため、我々は微小運動推定モデル<sup>[14]</sup>でカメラの回転を計算している。

空間点  $P$  を異なる二つの場所で観測していると仮定する。点  $P$  の一つの球面画像での投影点を  $m$ 、もう一つの球面画像での投影点を  $m'$  とする。二つの球面画像間の回転行列を  $R$ 、並進ベクトルを  $t$  とすると、ベクトル  $m$ 、 $m'$  と  $t$  は同じ平面上にあるので、次の式を得る。

$$m \cdot [t \times (Rm')] = 0 \quad (3.7)$$

運動は非常に小さいと仮定し、カメラの並進速度を  $v$ 、回転角速度を  $\omega$  とするとカメラの微小運動は次の式で示される。

$$\begin{cases} t = \delta t v \\ R = Id + \delta t [\omega_{\times}] \\ m' = m + \delta t \dot{m} \end{cases} \quad (3.8)$$

式(3.7)(3.8)から次の式が得られる。

$$m^T ([v_{\times}] [\omega_{\times}]) m - (m \times \dot{m}) \cdot v = 0 \quad (3.9)$$

この式は  $v$  と  $\omega$  の非線形方程式である。  $v$  または  $\omega$  が既知であると仮定すると線形方程式になる。このことより、本論文で次の反復方法で  $v$  と  $\omega$  を推定する。

**カメラ微小運動推定アルゴリズム：**

- ① 車が移動すると、以下の計算が始まる
- ②  $\omega_o = 0$  とする
- ③  $\omega_o$  の値で式(3.9)により並進ベクトル  $v$  を計算する
- ④ 計算した  $v$  の結果で式(9)により  $\omega_n$  を計算する
- ⑤  $\omega_o = \omega_n$
- ⑥ 次の式により推定誤差を計算し、その誤差が閾値  $T_e$  より小さい場合  $v$  と  $\omega$  を結果として出力し、そうでなければ③に戻る。

$$\sum_{i=1}^N \left( m_i^T ([v_{\times}] [\omega_{\times}]) m_i - (m_i \times \dot{m}_i) \cdot v \right)^2 \leq T_e \quad (3.10)$$

ただし、  $N$  は特徴点の数である。

同期時刻  $\Delta t$  は二つの魚眼カメラの録画した画像列から上記の手法で計算したカメラの回転角度データ列を使って 3.3.1 で説明した手法で求めることができる。

### 3.4 慣性センサーとの同期及び傾きの補正

全天周画像は坂道や傾斜などの路面状況の影響を受けて傾きが生じる。我々は慣性センサーを用いて傾きを補正する方法を提案する。まず、全天周カメラと慣性センサーの相対

姿勢の推定手法について述べる。

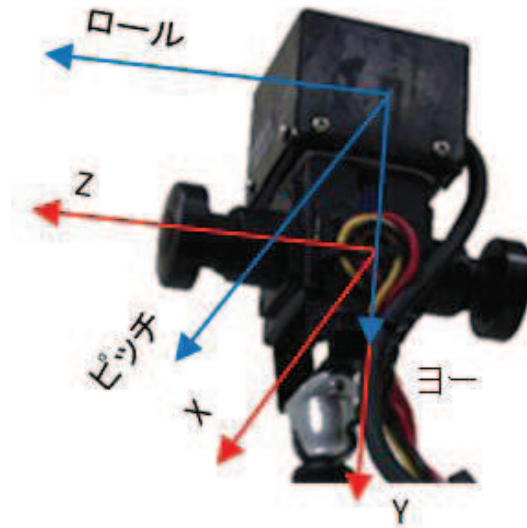


図 3.2 全天周カメラ（下）と慣性センサー（上）の座標関係

### 3.4.1 慣性センサーと全天周カメラの相対姿勢の推定

慣性センサー図 3.2（上部）は 3 軸の角速度と回転角度を検知できるセンサーであり，全天周カメラと一緒に固定されている．両者間の相対姿勢は，事前に以下の手法で測定しておく．

慣性センサーと全天周カメラとの間の座標関係は図 3.2 のように置かれている．カメラの撮る物体との距離に対し慣性センサーと全天周カメラとの距離は非常に小さいので二つの座標系の原点は重なっていると仮定することができる．しかし二つの座標系の中に小さな相対回転がある．その回転行列を  $R_{cs}$  とする．世界座標系の空間点を  $P$ ，世界座標系とカメラ座標系の回転行列を  $R_c$ ，世界座標系と慣性センサーとの回転行列を  $R_s$  とすると点  $P$  の二つの座標系の投影点  $P_c$ ， $P_s$  はそれぞれ

$$P_c = R_c P \quad P_s = R_s P \quad (3.11)$$

となる．よって

$$P_s = R_s R_c^{-1} P_c \quad (3.12)$$

相対姿勢を表す回転行列は、以下となる。

$$R_{cs} = R_s R_c^{-1} \quad (3.13)$$

その相対姿勢を実験室で事前に測定できる。次は相対姿勢で補正したデータより同期することについて述べる。

### 3.4.2 慣性センサーの同期

車が移動するとき、全天周カメラと慣性センサーとの同期は慣性センサーからの角速度と画像列から推定した全天周画像列のフレーム間の角度データ列を用いて行われる。同期する流れは次の通りである。

- 本研究で用いられている慣性センサーデータの速率はカメラと違って 60 Hz である。全天周画像列のフレーム速率に対応するために画像列と同じ速率でリサンプリングする。
- 慣性センサーの角速度と画像列から推定した全天周画像列のフレーム間の角度データ列を 3.3.1 で述べた手法でデータ列間の相関係数を計算し、同期時刻  $\Delta t$  を求める。

### 3.4.3 全天周画像の傾き補正

理論上、同期のとれた慣性センサーの角度よりカメラのすべての回転を補正することができる。しかし、激しい揺れがあるとき、加速度で計測される慣性センサーの角度には誤差が大きく、それに基づいたカメラの正確な姿勢補正が難しい。本研究で注目するのは坂道や傾斜などの路面状況の緩慢な変化である。坂道の傾斜角を慣性センサーから計測し、それを利用して全天周画像の傾きを補正する。処理の流れは次の通りである。

- 慣性センサーの角度データ列を 50 フレームの範囲で平均して平滑化する。ここでの 50 フレームは本研究で実験から決められた数字である。慣性センサーが加速度を計っているため、車の運動時の揺れに対して敏感である。図 3.3 が示すように、50 フレームの範囲で平滑化することにより、滑らかなデータ列を得ることができる。
- 式(13)に基づいた相対姿勢を補正した慣性センサーの角度データから、カメラの傾きによる回転行列を計算する。
- 回転行列より全天周画像を回転して傾きの補正後の画像を得る。

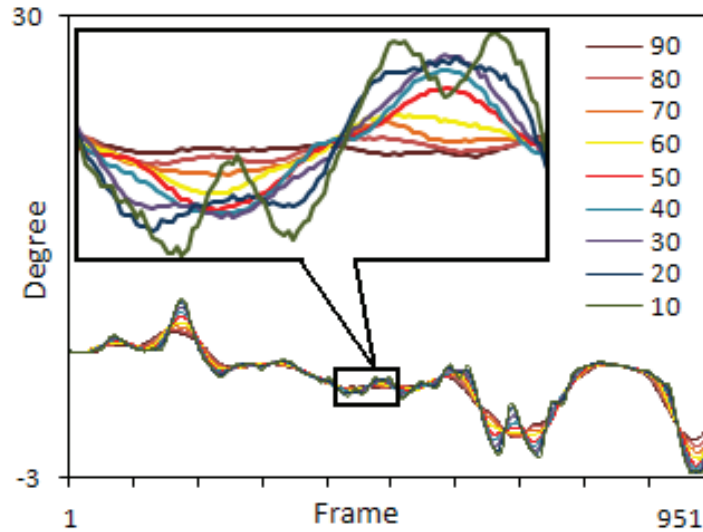


図 3.3 慣性センサーの角度データ列を 10 から 90 レームの範囲で平滑化した結果

### 3.5 GPS データの同期

本研究で使用されている GPS センサーは位置情報と方位情報を測定できるハイブリッド GPS で、測定データの速率は 4 Hz である。我々はこの方位角のデータより GPS センサーと全天周カメラを同期する。処理の流れは次の通りである。

- GPS センサーの方位角データにノイズ除去フィルタをかける。図 3.4 が示すように、車の滑らかな運動に対して、GPS で得た方位角に急な飛びが現れている。本研究では、以下の処理で方位角データのノイズを除去する。
  - ① まず、全方位角のデータ列に対して、フレーム間の方位角の平均変化量  $\overline{\Delta X}$  を算出する。
  - ② 次にフレーム間の変化量  $\Delta X_i = X_i - X_{i-1}$  を計算する。もし、 $\Delta X_i > \overline{\Delta X}$  であれば、 $X_i$  をノイズ点の候補として見なす。
  - ③ さらに、以下の式で、データ点位置  $i$  から範囲  $D$  内において、それらのデータ点の  $X_{i-1}$  との変化の最小値の位置  $k$  を計算する。

$$\arg\{\min_j \{\Delta X_j = X_j - X_{i-1}; (i \leq j < i + D)\}\} \quad (3.14)$$

ただし、本研究では、 $D$  を 10 とする。

- ④ もし、 $k > i$  であれば、 $X_i$  をノイズとして見なし、 $X_i$  の値を  $X_k$  の値で置き換える。

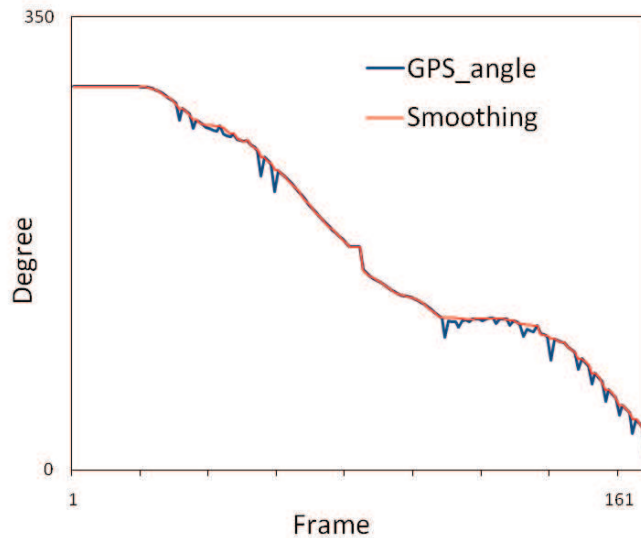


図 3.4 GPS センサーの方位角データのノイズを除去した結果

- GPS センサーデータの速率はカメラより低いため、線性補間して全天周画像列と同じ速率にする。
- 全天周画像列から推定した回転角度が隣接している二つのフレーム間の角度であり、GPS センサーの方位角データを全天周画像列のデータと同じく表示するため、隣接している二つの方位角の差を計算する。
- GPS センサーの方位角データ列と画像列から推定したヨー角のデータ列間の相関係数を式(4)により計算し、同期時刻  $\Delta t$  を求める。

同期のとれた GPS データから全天周画像列の各画像の地面における位置がわかる。しかし、連続録画される全天周画像列の地面における配置は車の速度変化に影響され、不均一になる。本論文では、連続録画された全天周画像列を同期のとれた GPS の位置情報より均等距離でリサンプリングしてデジタルマップに登録することができる。

### 3.6 実験と結果

本実験では、市販の部品で組み立てた全天周カメラ、データ・テック社の慣性センサーとパイオニアナビコム社のハイブリッド GPS 受信機 (GPS-MIZZ) を図 3.1 のように車両に搭載する。全天周カメラは二つの視野が 185 度の魚眼カメラで構成されている。

図 3.5 に、二つの魚眼カメラから撮像された画像を示す。その画像に対して、まず文献 [16] に示された手法で大きな固有値をもつコーナーの特徴点を最大 200 個抽出する。次に、それらの特徴点に対して、次の画像において LK-Tracker [17] で小さい誤差で追跡できた特徴点 (図 3.5 で赤と青点で示す) を用いて、カメラの移動を推定する。具体的には、上記の特徴点の発見と追跡は、画像処理ツール OpenCV [18] を用いて行われている。



図 3.5 魚眼カメラの隣接している二つのフレームのサンプル画像. 上の点はカメラの運動算出に抽出した特徴点になる

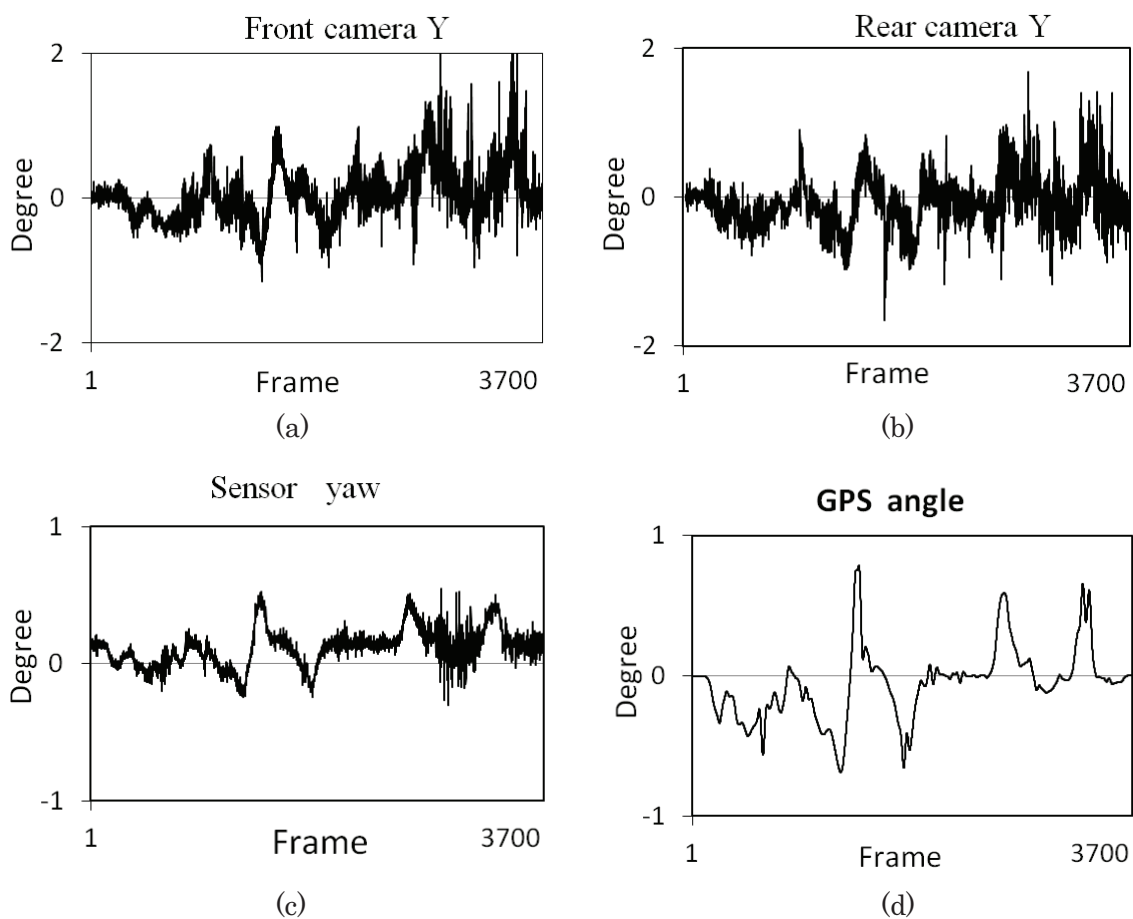
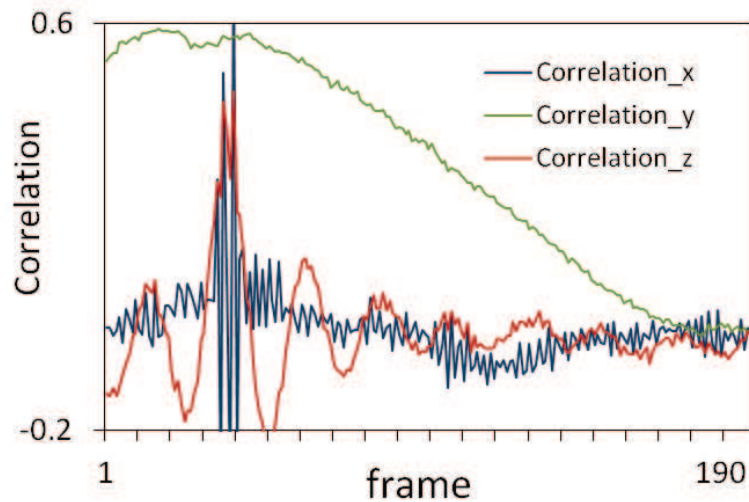


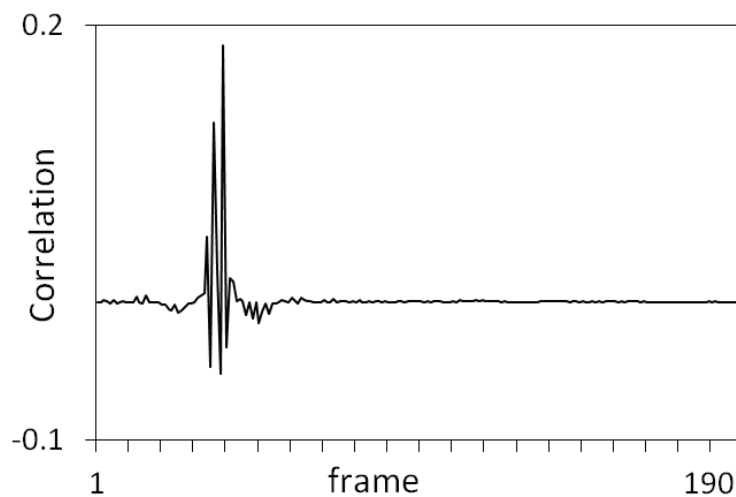
図 3.6 (a)と(b)はそれぞれ二つの魚眼カメラの画像列から推定した回転角度ヨー(y)データ列. (c)は慣性センサーの角度ヨーのデータ列 (d)は処理した GPS 方位角データ列

二つの魚眼カメラの回転をそれぞれの画像列から推定したヨー角の値を図 3.6 の(a)(b)で示している. (c)は慣性センサーからのヨー角度値で, (d)が GPS の方位角値をフレーム間の角度変化に変換したデータである. その結果が示すように, 4 つのデータ列の間には強い相関が存在する.

魚眼カメラのピッチ (x), ヨー (y) と ロール (z) 角度の相関係数の計算結果は図 3.7(a)で示している. 図 3.7(b)で示したように三つの相関係数の乗積を計算した回転運動全体の相関係数に著しいピークが存在する, その同期時刻は 38 フレームである.



(a)

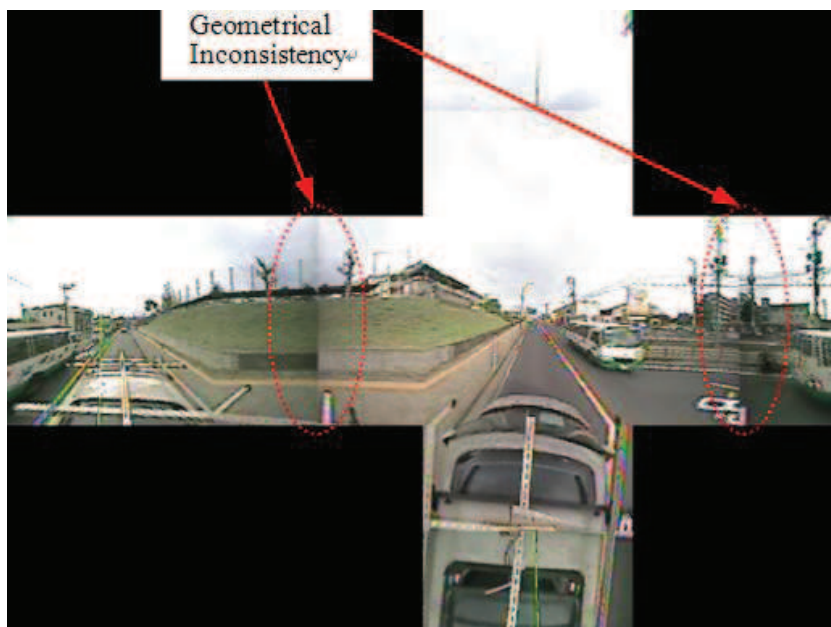


(b)

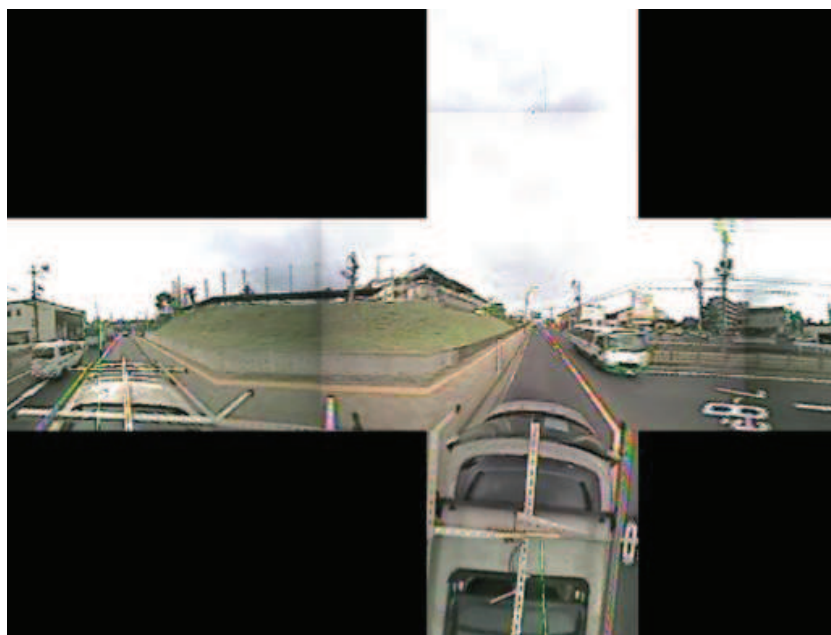
図 3.7 (a) 魚眼カメラのピッチ (x), ヨー (y) と ロール (z) 角度の相関係数の計算結果 (b) 3つの相関係数の乗積で表す回転全体の相関係数の結果



二つの魚眼カメラを同期して全天周画像作成した結果を図 3.8 で示す. 図 3.8 の(a)は二つの魚眼カメラの映像を同期する前の結果を示す. その図において半球面の境界に沿って明らかな幾何学的な不整合が見られる. (b)は同期後の結果となる. その幾何学的な不整合がなくなっている.



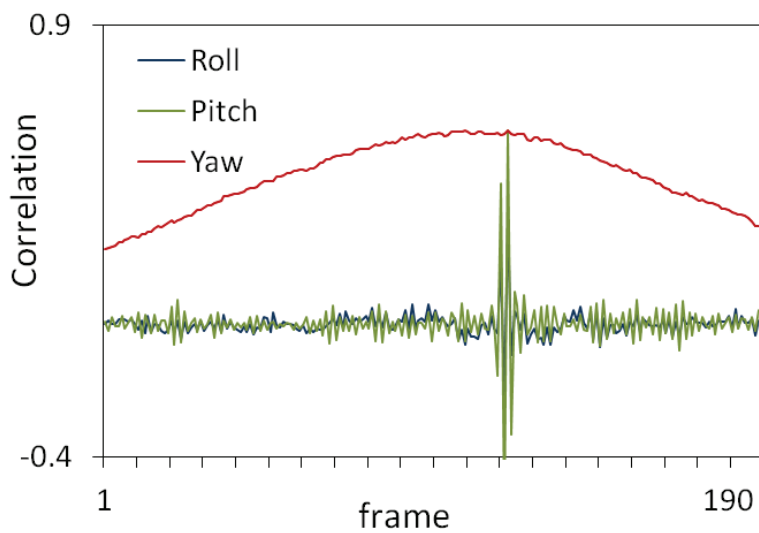
(a)



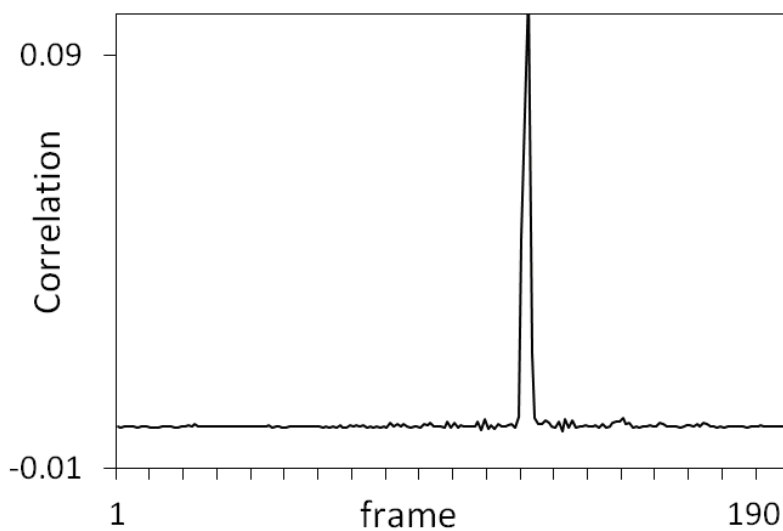
(b)

図 3.8 二つ魚眼カメラを同期して作成した全天周画像. (a)同期前の合成全天周画像 (b)同期後の全天周画像

慣性センサーの角速度と魚眼カメラの画像から推定した角度を図 3.2 のように各軸に対応させ、それによる、ピッチ、ヨーと ロール角度の相関係数の計算結果を図 3.9 の(a)で示す。また、三つの相関係数の乗積の計算結果を図 3.9 の(b)で示す。その著しいピークで定める同期時刻は 122 フレームであるのがわかる。



(a)



(b)

図 3.9 (a) 魚眼カメラと慣性センサーのピッチ (x), ヨー (y) と ロール (z) 角度の相関係数の計算結果. (b) 3つの相関係数の乗積で表す回転全体の相関係数の結果

同期した慣性センサーの角度より全天周画像の傾きを補正した結果は図 3.10 になる。この結果は車が坂道を上っている時に撮った画像である。左の図が補正前の様子で右の図が補正後の結果で道路標識の歪みが改善されている。



図 3.10 魚眼全天周画像の歪みの補正結果

GPS センサーの方位角データからフレーム間の角度変化を求め、それと全天周画像列のヨー角のデータとの相関係数を計算し、その結果を図 3.11(a)で示す。その同期時刻は 121 フレームである。

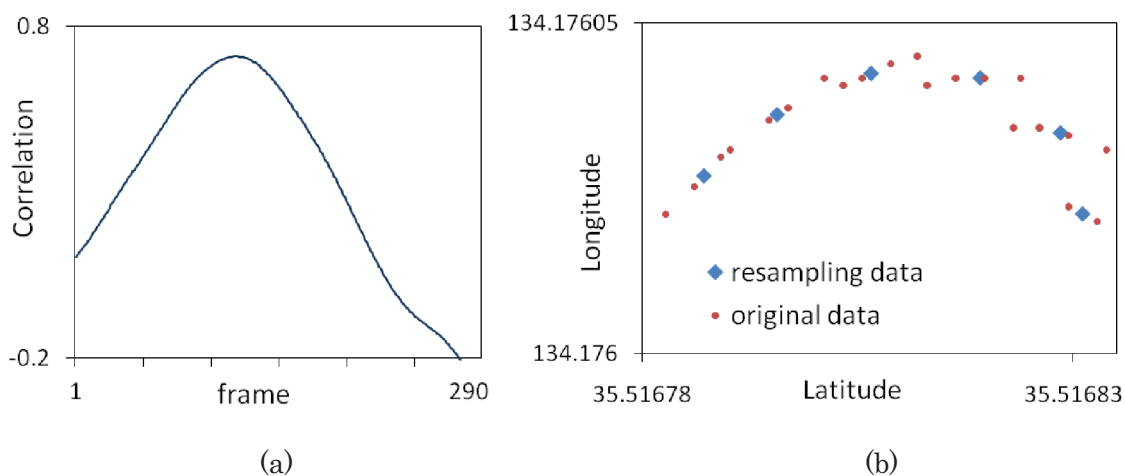


図 3.11 (a)GPS センサーの方位角と全天周画像列のヨー角の相関係数の計算結果。  
(b) 全天周画像列をリサンプリングした結果

同期のとれた全天周画像列を GPS 位置より均等な距離でリサンプリングした結果を 3.11(b)で示す。赤点は同期のとれた GPS の位置情報で示された全天周画像列の位置で、青点はリサンプリングした結果となる。車の速度は交通状況の影響で赤点のように不均一であるが、リサンプリングすることにより全天周画像はデジタルマップにおおよそ等しい間隔で登録されている。

最後に、均等距離でリサンプリングした全天周画像をオープン Google Map API を利用して Google マップに登録した結果を著者のウェブサイト[15] で公開している。図 3.12 は Google マップでクリックした位置に登録された全天周画像を表示する例を示す。

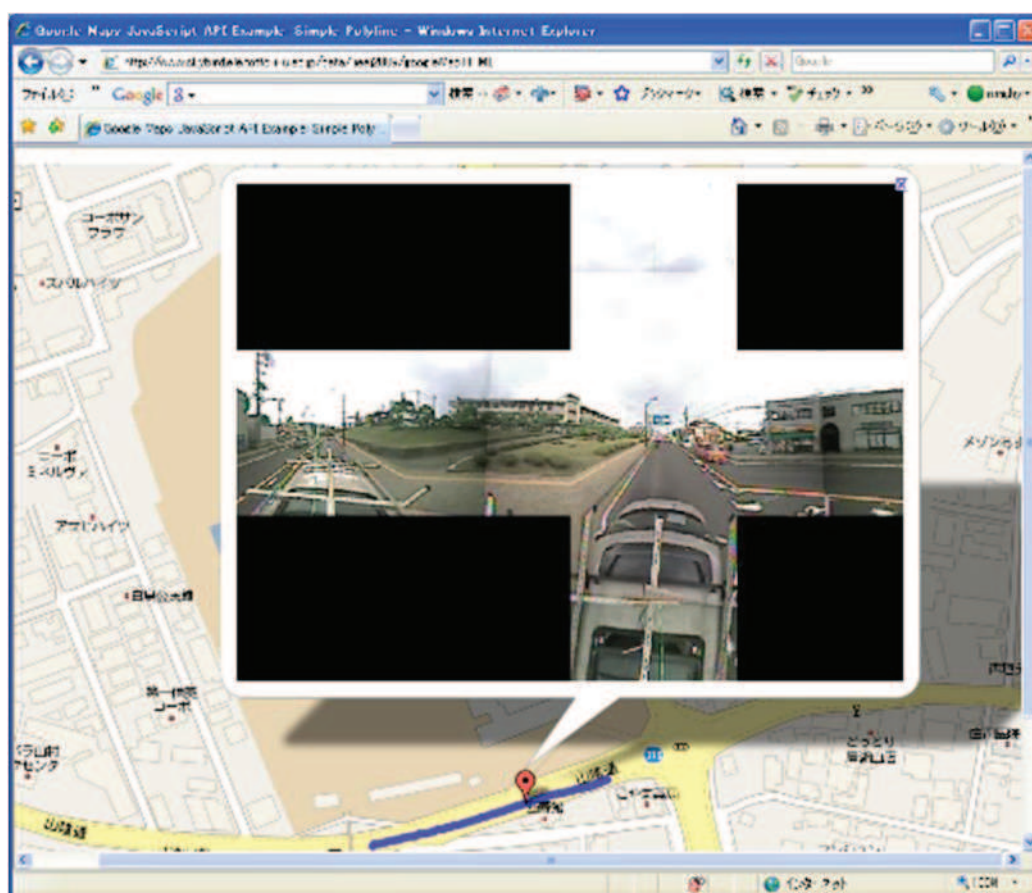


図 3.12 Google マップでクリックした位置の全天周画像のサンプル

### 3.7 まとめ

本研究では、全天周道路地図を構築するために、二つの魚眼カメラからなる全天周カメラと慣性センサーおよびハイブリッド GPS センサーを車に搭載し、それぞれ独立に記録したデータを共通の運動情報により同期する手法を提案した。主な成果は次の通りである。

- 複数のカメラを同期する方法として従来は重複視野を利用していた。本研究で我々は重複視野がほとんど存在しない二つの魚眼カメラの間の運動の整合性を利用してカメラ間の同期を行っている。カメラ間の回転運動の相関は、推定されたロール、ピッチ、ヨーの三つの角度の相関係数の乗積で算出される。その有効性が実験結果から示される。
- 坂道や傾斜などの路面状況の緩慢な変化で生じる全天周画像の傾きを全天周カメラと慣性センサー同期することにより補正する手法を提案した。
- 最後に全天周画像をハイブリッド GPS と同期して、ほぼ等距離でマップの道路に沿って登録する手法を提案した。

本提案手法の有効性を実験結果によって示した。この提案方法は低コストのシステムで簡単に全天周道路地図を構築することができる。

## 参考文献

- [1] J.Y. Zheng and S. Tsuji, "Panoramic representation for route recognition by a mobile robot", IJCV, 9(1), pp.55-76, 1992.
- [2] 李仕剛, 辻三郎, パノラマ表現内の三次元物体による経路シーンの記述, 電子情報通信学会論文誌, D-II, Vol. J76-D-II, No.10, pp.2177-2184, 1993年10月.
- [3] 越智亮, 李仕剛, 八木康史, 谷内田正彦, 経路と交差点の観測による環境地図の作成, 電気学会論文誌 C, 第118巻, 第4号, pp.510-519, 1998年4月.
- [4] S. Li and A. Hayashi, "Robot Navigation in Outdoor Environments by Using PGS Information and Panoramic Views", Proceeding of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.570-575, 1998.
- [5] J.Y. Zheng, "Route panorama", IEEE Multimedia, 10(3), pp.57-68, 2003.
- [6] J.Y. Zheng and S. Li, Employing fish-eye camera for scene tunnel scanning, ACCV, pp.106-117, January, 2006.
- [7] C. Frueh and A. Zakhor, "3D model generation for cities using aerial photographs and ground level laser scans", CVPR, 2001.
- [8] J. Hu, S. You and U. Neuman, "Approaches to large-scale urban modeling", IEEE Computer Graphics and Applications, 23(6): 62-69, 2003.
- [9] N. Cornelis, K. Cornelis and L.V. Gool, "Fast compact city modeling for navigation pre-visualization", CVPR, 2006.
- [10] H. Kawasaki, A. Miyamoto, Y. Ohsawa, S. Ono and K. Ikeuchi: "Multiple video camera calibration using EPI for city modeling", ACCV, Vol. 1, pp. 569-574 (2004).
- [11] Anthony Whitehead, Robert Laganier, "Temporal Synchronization of Video Sequences in Theory and in Practice", IEEE Workshop on Motion and Video Computing, Vol.2, pp. 132-137, 2005
- [12] Ushizaki.M et al. "Video Synchronization Based on Co-occurrence of Appearance Changes in Video Sequences", Proc of ICPR, Vol.3, pp. 71-74, 2006
- [13] Shigang Li, "Monitoring around a vehicle by a full-view spherical image sensor", IEEE Trans. On Intelligent Transportation System, Vol.7 No. 4, pp.541-550, 2006.
- [14] D.A. Forsyth and J. Ponce, Computer Vision A Modern Approach, Prentice Hall, 2003.
- [15] <http://www.skybird.ele.tottori-u.ac.jp/data/iieej2009/googleMap.html>
- [16] J. Shi and C. Tomasi, "Good Features to track", CVPR, pp. 593-600, 2004.
- [17] J. Y. Bouguet, "Pyramidal Implementation of the Lucas Kanade Feature Tracker: Description of the algorithm", Intel Corporation, Microprocessor Research Labs, 2000
- [18] <http://sourceforge.net/projects/opencvlibrary/>

## 第四章 全天周画像の表現法

大量の全天周画像を保存、圧縮、転送、処理するために有効な全天周画像表現法が必要である。全天周画像はよく球面画像を用いて表現される。本章では、全天周画像を表現する球面画像のフォーマット SIF (Spherical Image Format) を提案する。SIF は測地学モデルの球面構造に基づくコンパクトな矩形二次元配列で表し、その 2 次元配列では注目される画素に対する隣接画素を簡単に算出できるうえ、球面座標と球面画像画素の間の写像を行うことも可能である。本提案手法の応用例として、SIF 画像を介して全天周画像から高速で透視画像表示を行う実験結果を示す。

### 4.1 はじめに

ロボットや車など移動体の視覚センサーとしては、周りのシーンを覆う全天周画像を撮像できる全天周カメラが有用であり、環境マップ構築[1]、視覚監視[3]、移動ロボット、車の運転補助 [6]などの様々な研究領域で活用されている。また、Google のストリートビューのような仮想環境マップの構築にも全天周画像が用いられている。

全天周画像を獲得する手法としては、1 台のカメラで周囲を撮像した複数の画像を 1 枚

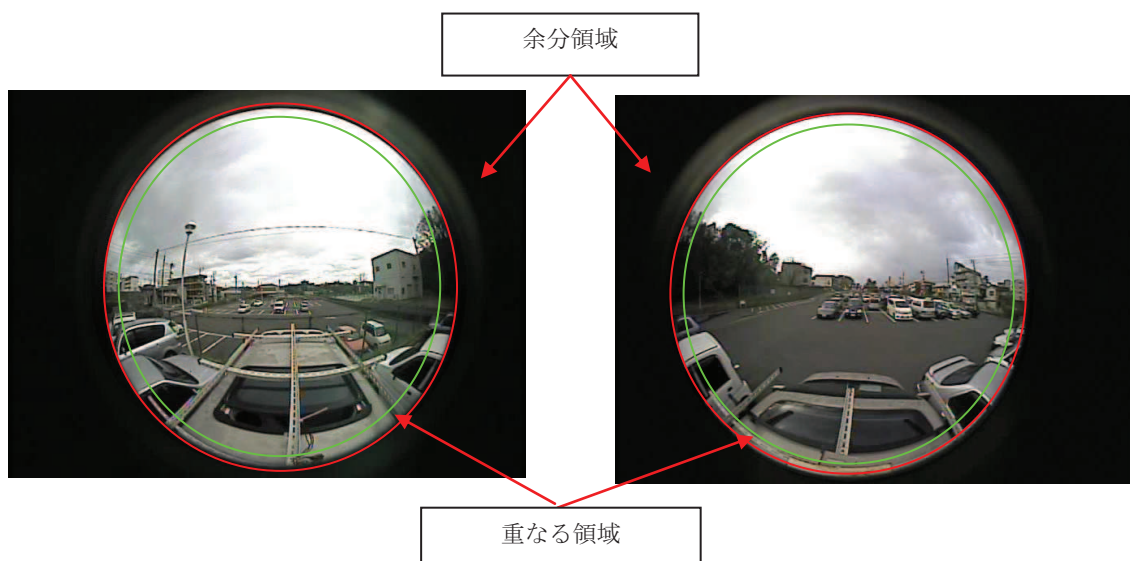


図 4.1 魚眼カメラから作成する全天周画像の余分領域と重なる領域

の全天周画像に合成する手法[1][2]と、周囲のシーンを複数のカメラで同時に撮像する手法[3][4][2]と、1台のカメラで異なる視野を覆う複数のレンズからの像を撮像する手法[6]に大別できる。ゆえに、1つの全天周画像は複数の子画像から構成される。全天周画像データを有効的に保存と転送するのは重要な課題である。

図 4.1 は、第二章で提案した一対の魚眼画像から全天周画像を獲得する例を示す。その全天周画像は、2枚の魚眼画像から構成されており、それぞれの魚眼画像には長方形の CCD 板で円形の半球視野のシーンを撮像することによって生じる余分な領域と、画像をつなぎ合わせるために生じる重複な領域が含まれる。上記の例では、1つの全天周画像は2つの魚眼子画像からなる。また、全天周画像に処理を施すには、カメラの内部パラメータ、子画像間の相対姿勢パラメータが必要である。

これに対しては、カメラパラメータを用いて全天周画像を球面画像に写像することができる。写像された球面画像ではカメラパラメータが現れなくなる。即ち、球面画像が正規化された全天周画像と見なすことができる。もし撮像された全天周画像を球面画像に変換して保存すれば、後処理は図 4.1 に示す余分領域と重複領域のような冗長なデータを持たない、撮像カメラのパラメータと無関係な球面画像に対して施せばよい。

本章では冗長なデータがない全天周画像情報だけに着目し、具体的な撮像デバイスに依存しない全天周画像を表現する球面画像フォーマット SIF を提案する。撮像デバイスから全天周画像をカメラパラメータを用いて測地学モデルの球面構造[6]に基づく球面画像に写像し、球面画像を1つのコンパクトな2次元配列 SIF 画像で表現する。その2次元配列で注目される画素に対する隣接画素を簡単に算出できるうえ、球面座標と球面画像画素の間の写像を行うことも可能である。提案手法の応用例として、SIF 画像を介して全天周画

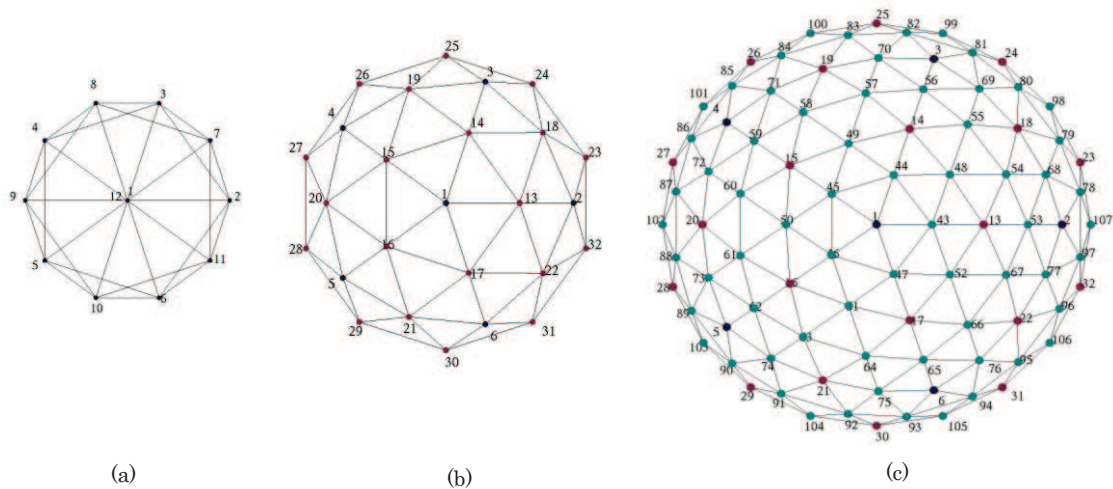


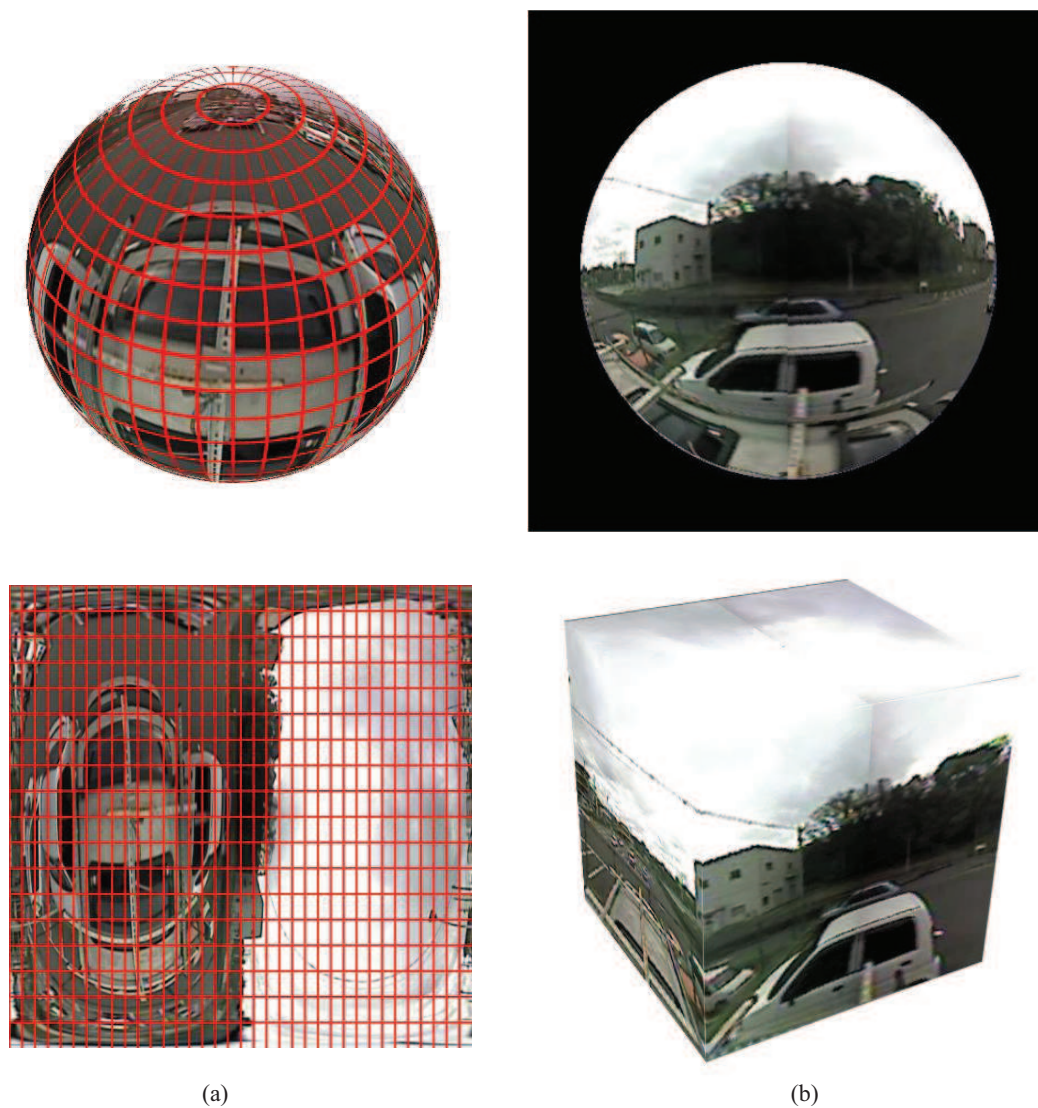
図 4.2 正 20 面体からの測地ドーム分割過程



像から高速で透視画像表示を行う実験結果を示す。

## 4.2 関連研究

デジタル計算機で球面画像を表現し処理を施すには、まず球面を標本化する必要がある。それはコンピュータビジョンとコンピュータグラフィックスの基本課題であり、球面を立方体で表現するキューブマップ、経緯マップ [5]がある。図 4.3 で示したように、球面画像の画素当たり、マップの面積が不均一である。図 4.2 に示すように、測地ドーム法に基づいた正 20 面体から出発し、その正 20 面体を構成する 3 角形を繰り返し分割すること



(a)

(b)

図 4.3 (a) 経緯度を表した球面画像と展開された経緯マップ  
(b) 球面画像とキューブマップ

により得られる頂点で行うのが一般的であり、ほぼ均一画素で表現することができる。

上記の球面の標本化手法で得られた画像は、拡張ガウス画像[7][8]、または、SCVT (Spherical Centroidal Voronoi Tessellation)画像[9]と呼ばれる。本論文では、後者の名称を用いる。計算機でSCVT画像を扱うには、それを保存するデータ構造が必要である。文献[7][8]では、正二十面体を繰り返し分割する過程に対応した階層的な木構造が用いられているが、そのデータ構造には画素間の隣接関係が保っていない。文献[10]では、図4.4が示すように、SCVT画像が展開される5つの平行四辺形として表される。平行四辺形の数を $i$ で、平行四辺形内の要素位置を $(j, k)$ で表すと、SCVT画像を1つの3次元配列 $(i, j, k)$ で表すことができる。その表現法の利点は、各平行四辺形において画素の隣接関係が保存されることである。その利点を利用した球面画像処理法も提案されている[10]。しかし、与えられた球面座標 $(\theta, \varphi)$ に対応する画素(球面分割の頂点)を決定する際には、その3次元配列に対する探索を行う必要がある。

図4.4では、1つの平行四辺形は正20面体の4つの三角形に対応する。その上下の2つの三角形の両側の辺に位置する頂点が隣平行四辺形の上下三角形の隣側の辺にある頂点と同一なものなので、重複して現れることになる。

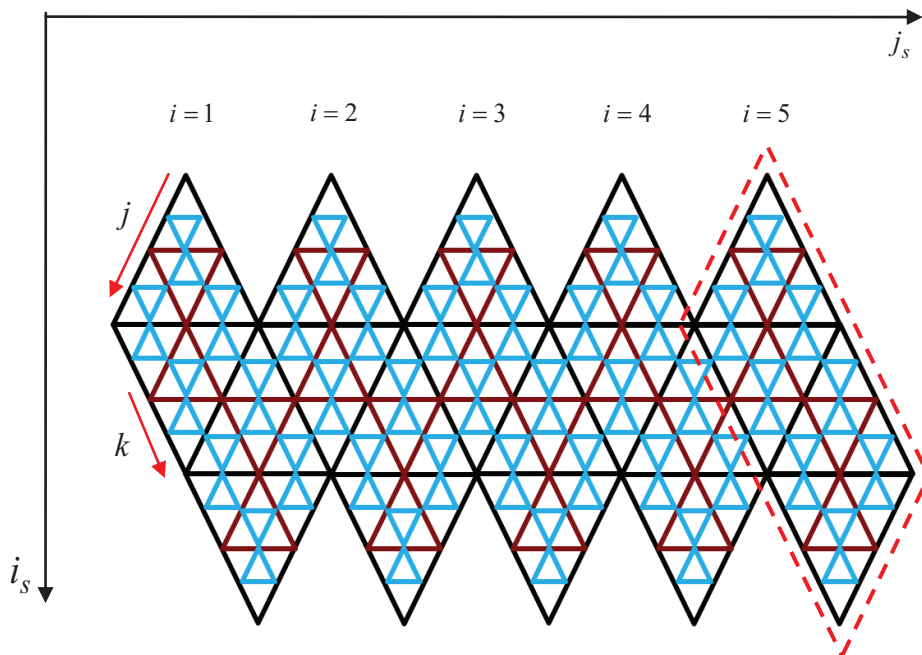


図4.4 SCVT画像を5つの平行四辺形として表すデータ構造 (ドットラインで一つの平行四辺形を表した)

本論文では、SCVT画像を2次元配列として表現する手法を提案する。提案手法では、与えられた球面座標 $(\theta, \varphi)$ に対応する画素の決定する際に、その2次元配列に対する探索を

行えばよい。また、その2次元配列に基づいて、注目される画素の隣接画素を簡単に算出することもできる。

### 4.3 二次元配列球面画像フォーマット

#### 4.3.1 SCVT 画像の二次元配列

本研究では、図 4.4 に示す従来法で SCVT 画像を 5 つの平行四辺形で表す代わりに、それらの分割頂点（画素に対応する）を左に詰め寄せて、1 つの 2 次元配列で表す。ただし、平行四辺形の上下の 3 角形の両側の辺にある重複な頂点は省く。本研究では、この 2 次元配列を SCVT 配列と名付ける。図 4.2 で示される正 20 面体、それを 1 回分割と 2 回分割して得られた頂点からなる SCVT 配列を図 4.5 に示す。

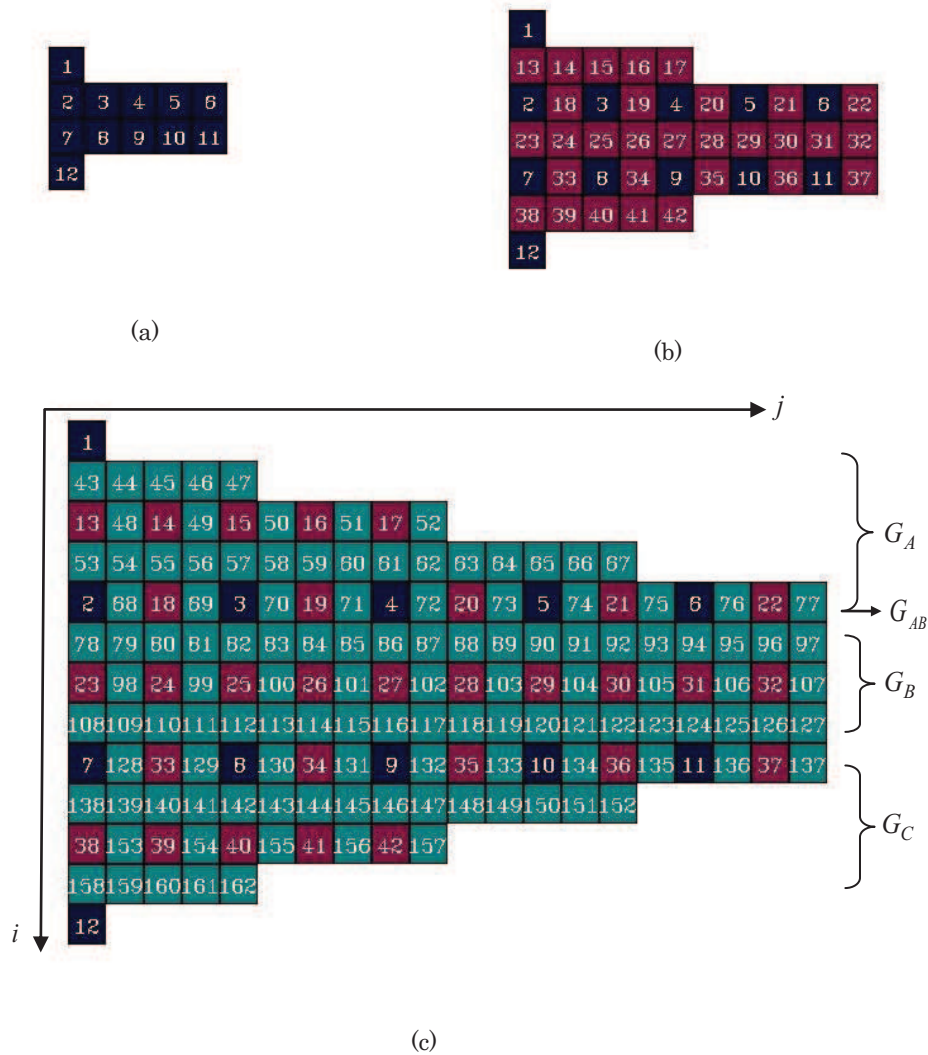


図 4.5 測地ドーム分割で得られた頂点からなる 2 次元 SCVT 配列

球面の分割回数を  $L$  とすると、対応の SCVT 配列の行数は  $3 \cdot 2^L + 1$  で、行中の頂点数は 1 から  $5 \cdot 2^L$  である。  $i$  を行番号で表記すると、行中の頂点数  $N_i$  を以下の式で計算できる。

$$N_i = \begin{cases} 1, (i = 0) \\ 5i, (0 < i \leq 2^L) \\ 5 \times 2^L, (2^L < i \leq 2^{L+1}) \\ 5 \times (R_L - i), (2^{L+1} < i < R_L) \\ 1, (i = R_L) \end{cases} \quad (4.1)$$

ただし、  $R_L = 3 \cdot 2^L$  である。

$L_v$  回分割時の頂点の位置を  $(i_v, j_v)$  とすると、  $L$  ( $L > L_v$ ) 回分割時の頂点  $(i, j)$  の位置  $(i, j)$  は以下の式で表せる。

$$i = 2^{(L-L_v)} i_v, j = 2^{(L-L_v)} j_v. \quad (4.2)$$

式 (4.2) より、  $L$  分割レベルの SCVT 配列での頂点はどの分割レベルで現れたものかを決めることができる。

### 4.3.2 隣接頂点算出

SCVT 配列では、球面上の頂点の隣接関係を行内に保っているが、隣り合う行に位置する頂点との隣接関係は一見保っていないように見える。実は、測地ドーム分割の規則性により、隣り合う行に位置する隣接頂点を簡単に算出できるのである。

次に分割回数  $L$  における注目される頂点  $P(i, j)$  に対する隣接頂点の算出法を説明する。図 4.5(c) で示すように SCVT 配列の頂点を  $G_A, G_B, G_C$  という 3 つのグループに分ける。  $G_A$  の末行は隣接頂点算出手法の特徴から  $G_{AB}$  と名付ける。注目される頂点  $P(i, j)$  がどのグループに属するかは、式 (4.1) を用いて判別できる。

球面に対する測地ドーム分割で得られる頂点は 3 種類に分類することができる。出発点とする正 20 面体の 12 個の頂点 (図 4.2 図 4.5 参照) は 5 つの隣接頂点をもつので、5 連結頂点と名付ける。残りのすべての頂点は 6 つの隣接頂点をもつので、同様に 6 連結頂点と名付ける。その 6 連結頂点では、多くの頂点の 6 つの隣接頂点は現在行と 2 つの隣接行からのそれぞれ 2 つの隣接点で構成されるが、図 4.4 に示す上下それぞれ 5 つの三角形の両側の辺に位置する頂点の隣接頂点の分布は異なる。本研究では、それらの 6 連結頂点をそれぞれ普通 6 連結頂点と特別 6 連結頂点と名付ける。頂点種類の判別は、式 (4.2) より

以下に行う.

注目される頂点  $P(i, j)$  に対して,

- $i\%2^L = 0$  ,  $j\%2^L = 0$  ならば,  $P$  が 5 つの隣接点をもつ 5 連結頂点である.
- $P \in G_A$ , または,  $P \in G_C$ ; かつ,  $j\%(N_i/5) = 0$  ならば,  $P$  が 6 つの隣接点をもつ特別 6 連結頂点である.
- それ以外ならば,  $P$  が 6 つの隣接点をもつ普通 6 連結頂点である.

ただし, “%” は余りを求める演算子である.


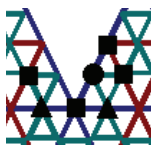
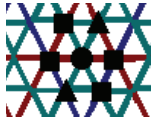

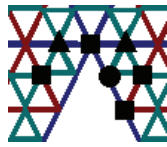
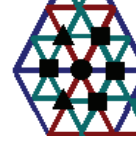
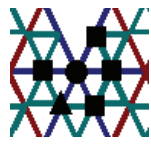
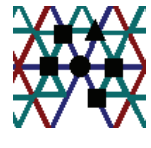
次に, 注目される頂点  $P(i, j)$  の種類を判別できたら, その隣接点を算出する方法について述べる. まず, SCVT 配列の上下に位置する 2 つの極点を特別な点とし, その隣接頂点が SCVT 配列での隣接行の 5 つの頂点になる. ほかの頂点に対しては, 上  $(ui, uj)$  下  $(di, dj)$  右  $(i, rj)$  左  $(i, lj)$  の 4 つの隣接点を以下に算出する.

$$\begin{cases} ui = i - 1 \\ uj = N_{i-1}/5 * (j/(N_i/5)) + j\%(N_i/5) \\ di = i + 1 \\ dj = N_{i+1}/5 * (j/(N_i/5)) + j\%(N_i/5) \\ lj = (j - 1 + N_i)\%N \\ rj = (j + 1)\%N \end{cases} \quad (4.3)$$

残りの 2 つの隣接頂点は表 1 で示す処理で算出する. 表 1 では,  $\Delta i_k (k=1,2)$  が注目される頂点  $P$  の行  $i$  に対する相対位置で,  $\Delta j_k (k=1,2)$  が注目される頂点  $P$  の列  $j$  に対する相対位置である. また, 5 隣接頂点の算出は 6 隣接頂点の算出の例外として扱うことができる. 具体的には, 2 つの極点以外のすべての頂点に対して, まず 6 隣接点と見なして計算する. ただし, その頂点が  $G_{AB}$  に属する場合  $\Delta i_k = -1$  に対応する点,  $G_C$  に属する場合には,  $\Delta i_k = 1$  に対応する点を削除すればよい.

まとめると, 本研究で提案される SCVT 配列にける隣接頂点を算出するアルゴリズムは以下の通りである.

表 4.1 SCVT 配列の接続点位置

	頂点種類	最後に算出される 2 つの隣接頂点の位置決定	サンプル図
6 接続	$G_A$ に属する普通 6 連結頂点	$\Delta i_1 = -1, \Delta j_1 = -1$ $\Delta i_2 = 1, \Delta j_2 = 1$	
	$G_A$ に属する特別 6 連結頂点	$\Delta i_1 = 1, \Delta j_1 = 1$ $\Delta i_2 = 1, \Delta j_2 = -1$	
	$G_B$ に属する普通 6 連結頂点	$\Delta i_1 = -1, \Delta j_1 = 1$ $\Delta i_2 = 1, \Delta j_2 = -1$	
	$G_C$ に属する普通 6 連結頂点	$\Delta i_1 = -1, \Delta j_1 = 1$ $\Delta i_2 = 1, \Delta j_2 = -1$	
	$G_C$ に属する特別 6 連結頂点	$\Delta i_1 = -1, \Delta j_1 = 1$ $\Delta i_2 = -1, \Delta j_2 = -1$	
	$G_{AB}$ に属する普通 6 連結頂点	$\Delta i_1 = -1, \Delta j_1 = -1$ $\Delta i_2 = 1, \Delta j_2 = -1$	
5 接続	$G_A$ に属する 5 連結頂点	$\Delta i_1 = \exists, \Delta j_1 = \exists$ $\Delta i_2 = 1, \Delta j_2 = -1$	
	$G_C$ に属する 5 連結頂点	$\Delta i_1 = -1, \Delta j_1 = 1$ $\Delta i_2 = \exists, \Delta j_2 = \exists$	

● 注目される頂点   ■ 上下右左隣接点   ▲ 未決定隣接点   ∃ 不存在

SCVT 配列における注目される頂点  $P(i, j)$  の属するグループを表す変数  $S$  を  $G_A$  の場合  $S = -1$ , それ以外の場合  $S = 1$  とする. 出力される隣接頂点配列を  $N_p$  とすると, 次の手順で隣接頂点を決めることができる.

- 1)  $P(i, j)$  が極点の場合に, 隣接頂点の 5 点を  $N_p$  に出力する.
- 2)  $P(i, j)$  が極点の以外の場合
  - ① まず, グループ変数  $S$  の値を決め, 式(3)で算出される上下右左の 4 つの隣接点を  $N_p$  に出力する.
  - ② 次に残りの 2 つの隣接頂点 (5 隣接の場合には 1 つの隣接頂点) を以下の通りに算出する.
    - 以下の式で  $(\Delta i_1, \Delta j_1), (\Delta i_2, \Delta j_2)$  を初期化する

$$\begin{aligned}\Delta i_1 &= -S, \Delta j_1 = 1 \\ \Delta i_2 &= S, \Delta j_2 = -1\end{aligned}\tag{4.4}$$

- $G_{AB}$  の場合に,  $\Delta j_1 = \Delta j_2 = -1$ . それ以外は, 特別 6 連結頂点の場合に  $\Delta i_1 = \Delta i_2 = -S$ .
- 5 連結頂点の場合に  $\Delta i_k = S; (k = 1, 2)$  の点を削除する.
- 次の式で  $\Delta i, \Delta j$  から位置を計算する.

$$\begin{aligned}i' &= \Delta i + i \\ j' &= (j_{N_p[\Delta i + 1]} + \Delta j + N_{\Delta i + i}) \% N_{\Delta i + i}\end{aligned}\tag{4.5}$$

ただし,  $j_{N_p[\Delta i + 1]}$  は  $N_p[\Delta i + 1]$  の列番号である.  $(i'_k, j'_k)$  を  $N_p$  に出力する

また, 頂点の属するグループ, その隣接頂点数などの情報を記録したルックアップテーブル (LUT) を球面分割と同時に作成しておくこと, 上記の処理を高速で行うことができる.

### 4.3.3 球面座標と SCVT 配列要素の間の写像

画像処理を行う際に, 画素の光線方向, あるいは, 与えられた光線方向の対応画素を計算する必要がある. そのため, SCVT 配列における頂点の球面座標, と, 与えられた球面座標の対応頂点を計算する必要がある. 頂点の球面座標は球面分割の際に生成される頂点の球面座標を LUT で保存すれば, その算出を高速で行うことができる. 一方, SCVT 配列では, 行間の角度間隔と行内の頂点間の角度間隔が不均一なので, 球面座標から SCVT

配列での対応頂点を算出する際に、探索が必要である。

本研究では文献[9]と同様の探索方法を用いる。与えられた球面座標  $(\theta, \phi)$  に対して、まず行間の平均角度間隔と行内の頂点間の平均角度間隔で初期位置を推定する。次にその初期位置から出発して、前小節で述べた手法で隣接点頂点を算出し、その中から球面座標  $(\theta, \phi)$  との誤差が最も小さい頂点を見つけ、見つけられた頂点を新しい出発点として繰り返し探索して行き、算出された頂点が更新されないとき、その頂点が解である。詳細手順は次のとおりである

- ① 行間の平均  $\bar{\theta}$  より球面座標  $(\theta, \phi)$  に対応する SCVT 配列の初期行  $i$  を決める。
- ②  $i$  の平均  $\bar{\phi}$  より球面座標  $(\theta, \phi)$  に対応する SCVT 配列の初期列  $j$  を決める。
- ③ SCVT 配列のセル  $(i, j)$  の隣接セルを探して球面座標  $(\theta, \phi)$  との最小距離値を更新する。
- ④ 最小距離値を更新した場合距離最小の隣接セルを現在セル  $(i, j)$  として③に戻る。最小距離値を更新しない場合、その SCVT 配列のセル  $(i, j)$  が球面座標  $(\theta, \phi)$  の対応セルとなる。

#### 4.3.4 二次元矩形球面画像フォーマット

図 4.5 で示すように SCVT 配列では行によってその行に含む頂点数が異なる。しかし、図 4.6 で示すように  $G_c$  の頂点を右に詰め寄せたあと、上に移動することでコンパクトな SCVT の 2 次元矩形配列へ簡単に変換できる。本論文では、その SCVT の 2 次元矩形配列を球面画像フォーマットとして用い、SIF (Spherical Image Format) と名付ける。SIF 画像には 2 つの極点以外のすべての SCVT 配列頂点を含んでいる。実際に、数百や数千万画素を持つ画像に対して 2 つの画素を省略しても、画像のもつ情報量にはほとんど影響がないと言える。また、必要があれば、2 つの極点の輝度値はその隣接頂点の輝度値から補間して算出することができる。図 4.7 は、図 4.1 に示す全天周画像を提案手法で球面にマッピングして得られた SIF 画像を示す。



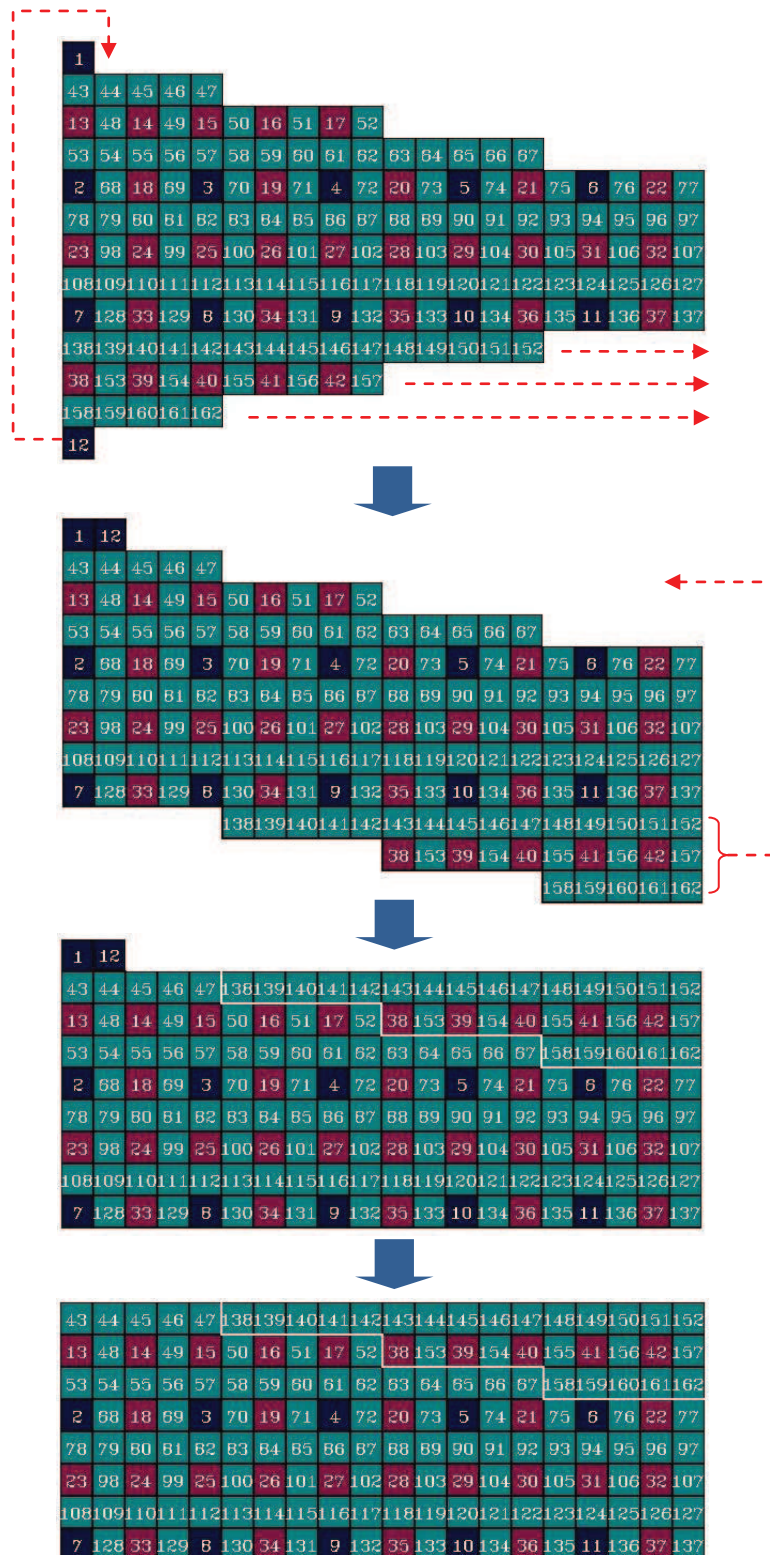


図 4.6 コンパクト 2D 矩形配列で表す球面画像フォーマット SIF の構成する手順

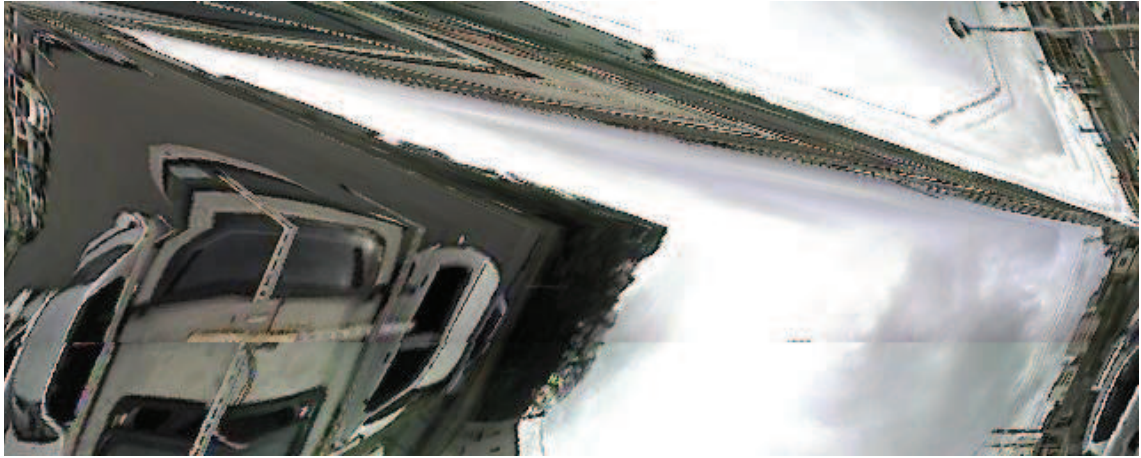


図 4.7 SIF で表現した全天周画像

#### 4.4 SIF 画像の透視画像表示

Google のストリートビューのように，全天周画像の環境マップをユーザーに提示する際には，ユーザーの視点に応じて透視画像を高速で生成する必要がある．視点が固定しているならば，その視点に対応する透視画像を生成するルックアップテーブルを事前に作成しておけば良い．しかし，その手法は，滑らかに連続に変化する視点に対応した透視画像の生成には対応できない．

透視画像の生成には，透視画像上の 1 点に対して，カメラのパラメータを用いて，入力画像上のその点の対応位置を算出する必要がある．図 4.1 に示す魚眼画像からなる全天周画像で，第二章でのカメラ歪みモデルを用いると，その算出過程は以下の通りとなる．(図 4.8 参照)

$$\begin{bmatrix} x_p \\ y_p \\ f_p \end{bmatrix} \rightarrow \begin{bmatrix} \theta \\ \varphi \end{bmatrix} \xrightarrow{(k_1, k_3, k_5, P_1, P_2)} \begin{bmatrix} r_i \\ \varphi_i \end{bmatrix} \rightarrow \begin{bmatrix} x_f \\ y_f \end{bmatrix} \quad (4.6)$$

ただし， $(x_p, y_p, f_p)$  が透視画像の画素の座標， $(\theta, \varphi)$  が光線方向の球面座標であり， $(r_i, \varphi_i)$  が撮像魚眼画像の極座標， $(x_f, y_f)$  が撮像魚眼画像の直交座標である． $(k_1, k_3, k_5, P_1, P_2)$  が魚眼カメラのパラメータである．

式(4.6)では，光線方向の球面座標から対応の魚眼画像の極座標を算出する際に，式(2.11)の非線連立方程式を解く必要がある．そこで球面は仮想モデルでルックアップテーブルにより快速化するには困難である．したがって，SIF 画像フォーマットで球面を表現する場

合決めたカメラの魚眼画像と SIF 画像の間ルックアップテーブルを作成することができる。もし撮像された全天周画像を SIF 画像に変換して持つと、生成される透視画像の画素の算出過程は以下になる。

$$\begin{bmatrix} x_p \\ y_p \\ f_p \end{bmatrix} \rightarrow \begin{bmatrix} \theta \\ \varphi \end{bmatrix} \xrightarrow{\text{Search}} \begin{bmatrix} i \\ j \end{bmatrix} \xrightarrow{\text{LUT}} \begin{bmatrix} x_f \\ y_f \end{bmatrix} \quad (4.7)$$

ただし、 $(i, j)$  は SCVT 配列の頂点座標である。また、 $(i, j)$  と  $(x_f, y_f)$  の対応関係はルックアップテーブルを用いると、高速で算出できる。

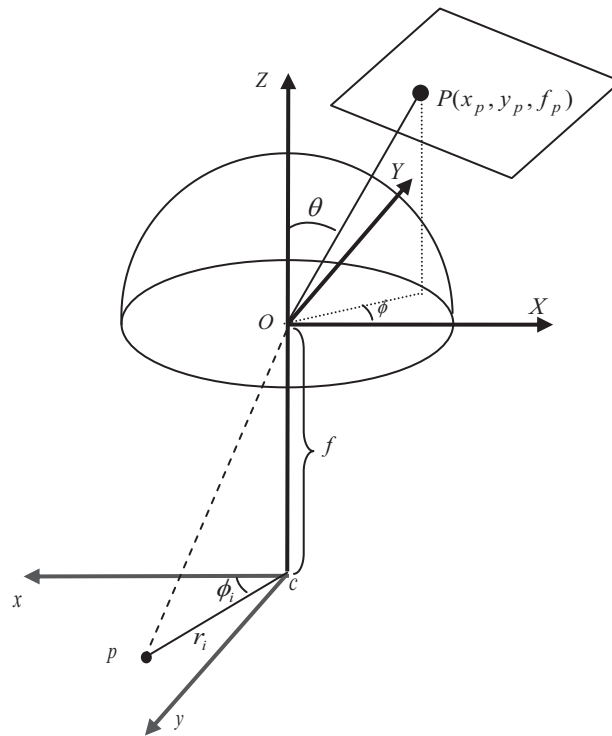


図 4.8 透視画像表示と魚眼画像との関係

## 4.5 実験

提案された SIF 球面画像フォーマットの透視画像作成効率をテストするために、第 2 章で説明した全天周カメラで撮られた一対の魚眼画像を用いて透視画像作成する実験を行ない、その結果を従来法と比較した。そこで、魚眼カメラから撮った画像の解像度  $640 \times 480$

である.

図 4.7 のような SIF 画像をからから提案手法により作成された投透視画像の結果を図 4.9 の (b) で示し, (a) は従来手法で作成された透視画像である. 比べて見るとほとんど同じ結果であることがわかる. 表 4.2 では, ランダムに解像度  $300 \times 300$  画素の透視画像を 1000 枚作成し, 一枚当たりの平均処理速度を示し, その処理は CPU Intel Pentium D 2.8-GHz メモリ 512-MB のデスクトップパソコンで行われた. 非線形方程式を解かないことにより, 提案法の処理速度を約 21 倍にまで向上することができた. この結果は提案球面画像フォーマットの有効性を示している.

表 4.2 処理時間を比べた結果

手法	透視画像一枚当たりの平均処理速度
非線形方程式で計算	3010 ms
SIF画像	139 ms

## 4.6 まとめ

本章では, 全天周画像を表現する球面画像フォーマット二次元矩形配列 SIF について述べた. SIF 画像における隣接画素の探索や画素と球面画像との間の対応関係を算出するアルゴリズムも提案した. その応用例として, SIF 画像を介して全天周画像から高速で透視画像表示を行う実験結果を示した.

全天周カメラの撮像部では, 内部で全天周画像を SIF 画像に変換し出力すると, ユーザーが扱うのは SIF 画像になる. その SIF には冗長なデータが含まれないうえ, カメラ内部パラメータにも依存しない. そのため, ユーザーには扱いやすい.



(a)



(b)

図 4.9 (a) 従来手法で作成した 6 面の透視画像 (b) 提案手法で SIF 画像から生成された 6 面の透視画像

## 参考文献

- [1] S. Coorg and S. Teller. “Spherical mosaics with quaternions and dense correlation”. In *International Journal of Computer Vision*, Vol. 37, pp. 259-273, 2000.
- [2] 中野誠士, 李仕剛, 千葉則茂: 球面画像獲得のための魚眼カメラの校正, 電子情報通信学会論文誌 D-II Vol.J88-D-II, No.9, pp.1847-1856, 2006.
- [3] N. D. Jankovic and M. D. Naish, “Developing a modular active spherical vision system”, ICRA, 2005.
- [4] Ladybug , <http://www.ptgrey.com/products/ladybug2/index.asp>
- [5] OpenGL, Open Graphics Library , <http://www.opengl.org>
- [6] S. Li, “Monitoring around a vehicle by a spherical image sensor”, *IEEE Trans. on Intelligent Transportation System*, Vol.7, No.4, pp.541-550, 2006
- [7] B. K. P. Horn, “Extended Gaussian image”, *Proc. IEEE*, Vol.72, No. 12, pp.1671-1686, 1984.
- [8] K. Ikeuchi, “Generating an interpretation tree from a cad model for 3D-object recognition in bin-picking tasks” , *Int. J. Computer Vision*, Vol. 1, No. 2, pp.145-165, 1987.
- [9] Q. Du, M. Gunzburger, and L. Ju, “Constrained centroidal Voronoi tessellations on general surfaces” , *SIAM J. Sci. Comput.*, 2003.
- [10] C.H. Chen and A.C. Kak, “A robot vision system for recognizing 3-D objects in low-order polynomial time”, *IEEE Trans. on Systems, Man, and Cybernetics*, Vol.19, No.6, pp.1535-1563, 1989.
- [11] 木室義彦, 長田正, “球面六角形画像ピラミッドを用いた球面投影による全方位画像処理” , Vol.14, No.6, pp.809-819, 1996.
- [12] 中野誠士, 李仕剛, 千葉則茂, “球面モデルに基づくしま模様パターンを用いた魚眼カメラの校正”, 電子情報通信学会論文誌 D, Vol.J90-D, No.1, pp.73-82, 2007.
- [13] [http://www.skybird.ele.tottori-u.ac.jp/SIF\\_img.html](http://www.skybird.ele.tottori-u.ac.jp/SIF_img.html)

## 第五章 バイナリ運動履歴画像による交差点における側面からの接近車両の検出

車載カメラ映像から全運転を支援するもう一つの重要な研究は環境の認識である。交差点では運転者が側面から接近する車両に十分に注意を払わなかったことが原因である交通事故が発生しやすい場所である。特に、周りに建物がある場合側面からの接近車両を正確に把握するのが困難である。本研究では、バイナリ運動履歴画像による交差点における側面からの接近車両の検出手法を提案する。バイナリ運動履歴画像は、一定の時間間隔における動画像列のフレーム間の差分画像を2値化して得られるバイナリ差分画像を重ねて構成される。そのため、従来の運動履歴画像に比べると、バイナリ運動履歴画像ではその時間間隔内の運動履歴が完全に保存される。さらに、新たに生成されるバイナリ差分画像をバイナリ運動履歴画像の上位ビットに配置することにより、従来の運動履歴画像のようにバイナリ運動履歴画像では運動領域の輝度値が持続フレーム数に応じて減衰していく。実験結果では、本提案手法は従来法に比べ、より高速にかつ正確に交差点における側面からの接近車両を検出することができる。

### 5.1 はじめに

交差点における交通事故が多発している。その多くは、運転者が側面から接近する車両に十分に注意を払わなかったためである。図 5.1 はこのような交通環境の一例を示す。小道の周りに建物がある場合に、ドライバの視野が狭くなり、側面からの接近車両を正確に把握するのが困難である。トヨタ社のフロントモニターは超広角カメラで見通しの悪い交差点の状況を撮影し、図 5.2 のようにモニターに表示している。これにより、交差点での接近車両や自転車、歩行者の確認、そして発進時の前方確認がスムーズかつ広範囲に行えることができる。しかし、前方交差点で接近車両を自動検出できないために、ドライバはモニターに注意を払って接近車両を探す必要がある。本研究では、半球視野を持つ魚眼カメラを車両の前方に設置し、広い視野で交差点を観測して、側面接近車両の検出を行う。

自車両は交差点に向かってゆっくり前方に移動する、または、停止していると仮定し、自車両前方に設置した魚眼カメラから交差点における側面接近車両を検出するには、以下の課題を解決する必要がある。

- 魚眼カメラの画像列から動物体の位置を検出し、その処理領域を決定する必要がある

る。

- 検出された動物体の運動方向を推定し，その動物体が接近してくるかを判別する。

動物体の検出について，運動履歴画像（MHI（Motion History Image））と呼ばれる時間テンプレートが提案され，運動領域と運動方向を推定することができる．本研究では，リアルタイムで側面接近車両を検出するために，バイナリ運動履歴画像（BMHI）とよばれる新しいテンポラルテンプレート手法を提案する．

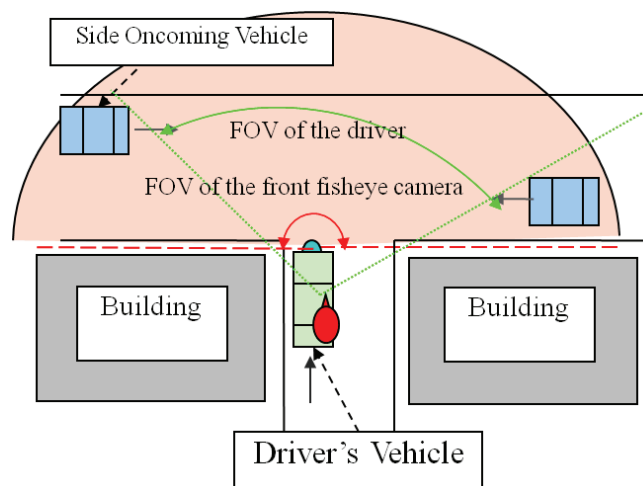


図 5.1 本研究で解決問題の略図.



図 5.2 トヨタ社のフロントモニターの映像.

## 5.2 関連研究

近年様々な車両検出手法が提案されている，文献[1][8]はステレオに基づく手法を利用して車両検出を行っている．しかし，ステレオ計算には高精度のカメラパラメータの測定が必要である．文献[5][11]などで学習に基づく手法では複雑な戶外環境(照明条件，乱雑な



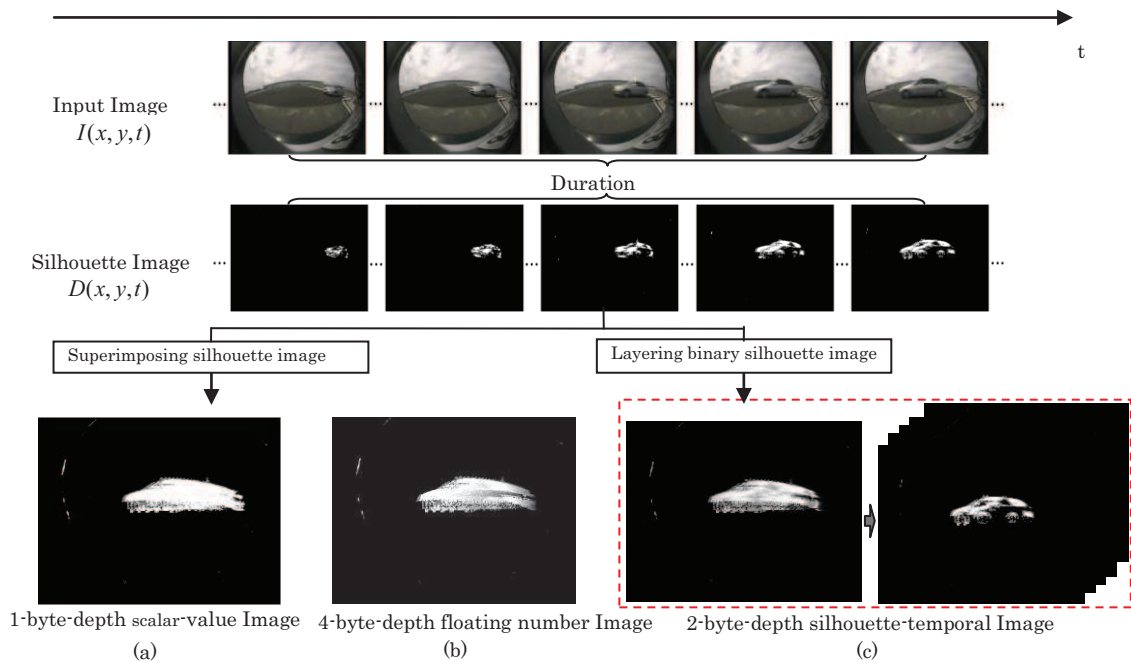


図 5.3 (a)運動履歴画像(MHI). (b) 時間的運動履歴画像 (tMHI). (c) 提案手法: バイナリ運動履歴画像(BMHI). 以上すべての運動履歴画像は連続 16 フレームの差分画像から構成されている

背景など) においては、誤検率が高まる問題があり、その結果、実際の応用ではパフォーマンスが大幅に低下している。本研究では、実際の戸外環境にリアルタイムで運動情報を利用した運動物の検出手法に着目する。これに対して、文献[6][4]では運動履歴画像と呼ばれる静的な環境における高速に運動領域を検出し、その運動方向を推定する時間テンプレート手法が提案されている。

文献[6] では、静的な環境における物体の運動を表現するには、運動速度を考えて一定間隔フレーム間での差分を行い、その差分画像を基本パターンとして運動履歴画像を更新する。新しいフレームが到着する時に、指定間隔の前のフレームとバイナリ差分画像列  $D(x, y, t)$  を作成し MHI 画像  $H_{\delta}(x, y, t)$  を次の式で更新する。

$$H_{\tau}(x, y, t) = \begin{cases} \tau & \text{if } D(x, y, t) = 1 \\ \max(0, H_{\tau}(x, y, t-1) - 1) & \text{otherwise} \end{cases} \quad (5.1)$$

ただし、 $\tau$  が運動履歴画像の持続フレーム数である。(図 5.3 の(a)を参照)

文献[4]では、浮動小数点数型のシステム時間をタイムスタンプとして運動履歴画像 (tMHI (timed MHI))  $tMHI_{\delta}(x, y)$  を次の式で更新する。

$$tMHI_{\delta}(x, y) = \begin{cases} \tau & \text{if } D(x, y, \tau) = 1 \\ 0 & \text{else if } tMHI_{\delta}(x, y) < (\tau - \delta) \end{cases} \quad (5.2)$$

ただし、 $\tau$  が現在システム時間で、 $\delta$  が最大時間持続定数である。この表現法の利点としては、システムの処理速度あるいはフレームレートに関係なく、一定時間間隔内の運動履歴情報が保存される。(図 5.3 の(b)を参照)

しかし、上記の手法では、バイナリ差分画像  $D(x, y, t)$  に複数の運動領域が現れる時、前の時刻の差分画像の運動領域は、そのあとの時刻の差分画像の運動領域に上書きされ、それらの運動領域の情報が消失する問題点がある。本論文では、この問題を運動領域消失と呼ぶ。

本論文では、運動領域消失問題を解決できるバイナリ運動履歴画像(BMHI)という新しい時間テンプレートを提案する。バイナリ運動履歴画像は、一定の時間間隔における動画像列のフレーム間の差分画像を2値化し、その2値画像をビット画像として重ねて構成される。そのため、バイナリ運動履歴画像ではその時間間隔内の運動履歴が完全に保存され、運動領域消失問題がなく、動物体の運動履歴を解析できる(図 5.3 の(c)を参照)。さらに、新たに生成されるバイナリ差分画像をバイナリ運動履歴画像の上位ビットに配置することにより、従来の運動履歴画像のようにバイナリ運動履歴画像では運動領域の輝度値が持続フレーム数に応じて減衰していく。なおかつ、ビットのシフト演算がシステムバスの範囲内であれば、ビット数が変わってもほとんど同じ計算コストで処理できる。提案手法は、従来法に比べ、より高速にかつ正確に交差点における側面からの接近車両を検出することができる。

### 5.3 バイナリ運動履歴画像の生成

本節ではバイナリ運動履歴画像 (BMHI) の生成について述べる。ここで  $tMHI$  と同じくフレーム間のバイナリ差分画像  $D(x, y, t)$  を入力データとする。BMHI 画像のビット数  $\tau$  が運動持続時間内のビット画像数 ( $\tau$  フレーム) である。 $t$  時刻のバイナリ運動履歴画像  $B_{\tau}(x, y, t)$  を次の式で更新する。

$$B_{\tau}(x, y, t) = \begin{cases} B_{\tau}(x, y, t-1)/2 + 2^{\tau-1} & \text{if } D(x, y, t) = 1 \\ B_{\tau}(x, y, t-1)/2 & \text{else if } B_{\tau}(x, y, t-1) > 0 \end{cases} \quad (5.3)$$

ただし、上式における “/” と “+” という演算がビットのシフト演算に対応することが

できる。

式(5.3)が以下のように高速に計算できる。BMHI 画像のビット数 $\tau$ が決まっていれば、 $2^{\tau-1}$ が定数である。C= $2^{\tau-1}$ とすると、 $B_{\tau}(x,y,t-1)/2+2^{\tau-1}$ が  $B_{\tau}(x,y,t-1) \gg 1|C$  というビット演算に対応する。つまり、BMHIを更新する際に、まず元のBMHIのデータが下位ビットへ移動し、つぎに新たに入ってくる $D(x,y,t)$ を上位ビットに入れる。そのため、BMHI画像では、更新するごとに前の時刻の運動領域の輝度値が半減される。BMHI画像

表 5.1  $D(x,y,t)$ , tMHI および BMHI のそれぞれの特徴

	$D(x,y,t)$	tMHI	BMHI
<b>データ構造</b>	バイナリ差分画像列	運動領域を時刻順に上書きして生成される1枚の浮動小数点数型画像	バイナリ差分画像列からなる1枚の整数型画像
<b>運動領域消失問題</b>	なし	あり	なし
<b>運動領域輝度値の減衰</b>	なし	持続時間に応じて輝度値が減衰していく	持続フレーム数に応じて輝度値が半減していく
<b>データ更新</b>	最新のバイナリ差分画像を含む持続フレーム数のバイナリ差分画像列に対応する	最新のバイナリ差分画像で浮動小数点数型の運動履歴画像を書き換える	最新のバイナリ差分画像を上位ビットに配置するビット演算で行う

の生成例を図 5.3 の(c) で示す。

BMHI はバイナリ差分画像から構成されるため、運動領域に関する三次元時空間画像に対応する。また、tMHI と同じく、BMHI を1枚の2次元画像と見なせば、新たに現れた運動領域が高い画素値を持ち、tMHI と同じく輝度値の変化の勾配に基づく運動領域分割も可能である。表 5.1 には、バイナリ差分画像列  $D(x,y,t)$ 、従来の運動履歴画像 tMHI、および提案されるバイナリ運動履歴画像 BMHI に対して、データ構造、運動領域消失問題、運動領域輝度値の減衰、およびデータ更新に関するそれぞれの特徴を示す。

以下は、BMHI の性質を利用して、交差点における側面からの接近車両の検出を行う。

## 5.4 側面から接近車両の検出

### 5.4.1 背景と側面から接近車両の分離

5.1 節で述べたような交差点において、BMHI を用いて側面からの接近車両を検出する。

ここでは、カメラの内部パラメータと外部パラメータを既知とし、自車両がとまっているか或は交差点に向かってゆっくり直進すると仮定する。この状態で自車両の運動による生じる背景の画像上での運動方向（図 5.4 で示す細い矢印）をカメラパラメータから事前に計算しておく。他車両が側面から接近と、その運動方向が背景の運動方向と反対になる（図 5.4 で示す太い矢印）。この特性を利用して背景運動と側面から接近車両の運動を区別でき、背景から側面接近車両領域を分離する。

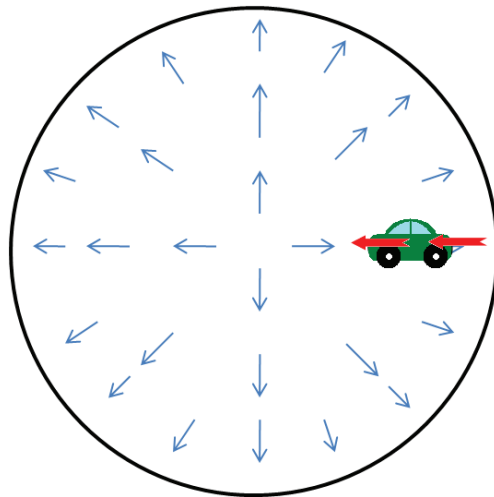


図 5.4 側面から接近車両を検出手法

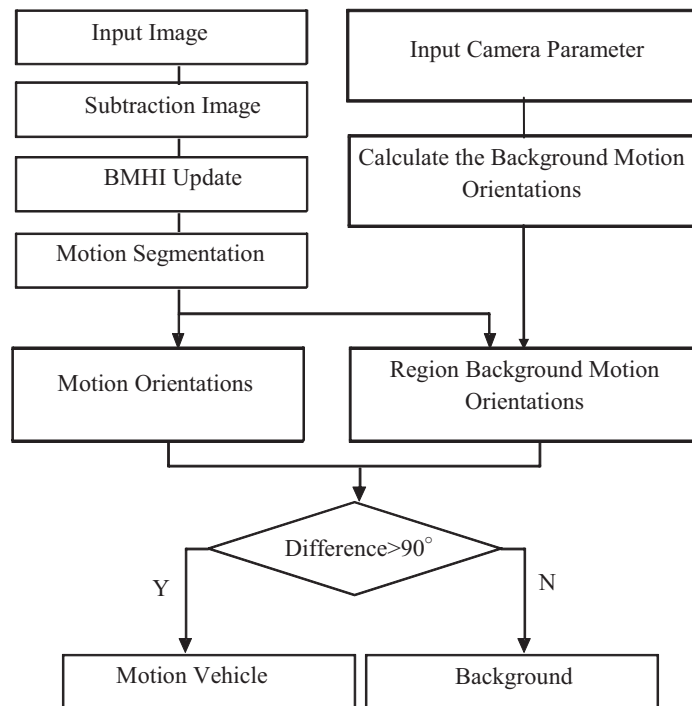


図 5.5 本手法の処理流れ

## 5.4.2 運動領域分割と運動方向推定

本処理の流れを図 5.5 で示す. 運動分割については, tMHI と同様に BMHI を 1 枚の二次元画像と見なし, 運動領域分割を行う. BMHI に対して, 画素値が零ではない画素値を 1 とした二値化画像  $BB_t(x,y,t)$  を作成し, オープニングとクロージングのモルフォロジ演算で小さなノイズを除去し穴埋めを行う. BMHI 画像において最上位ビットが 1 の画素をシート点として選び,  $BB_t(x,y,t)$  画像値が 1 の連続領域を一つの領域として新しい値を付けて領域を記述する. 分割した領域以外新しいシート点選び続けて領域分割を行い, 図 5.6 で領域分割結果を表している.

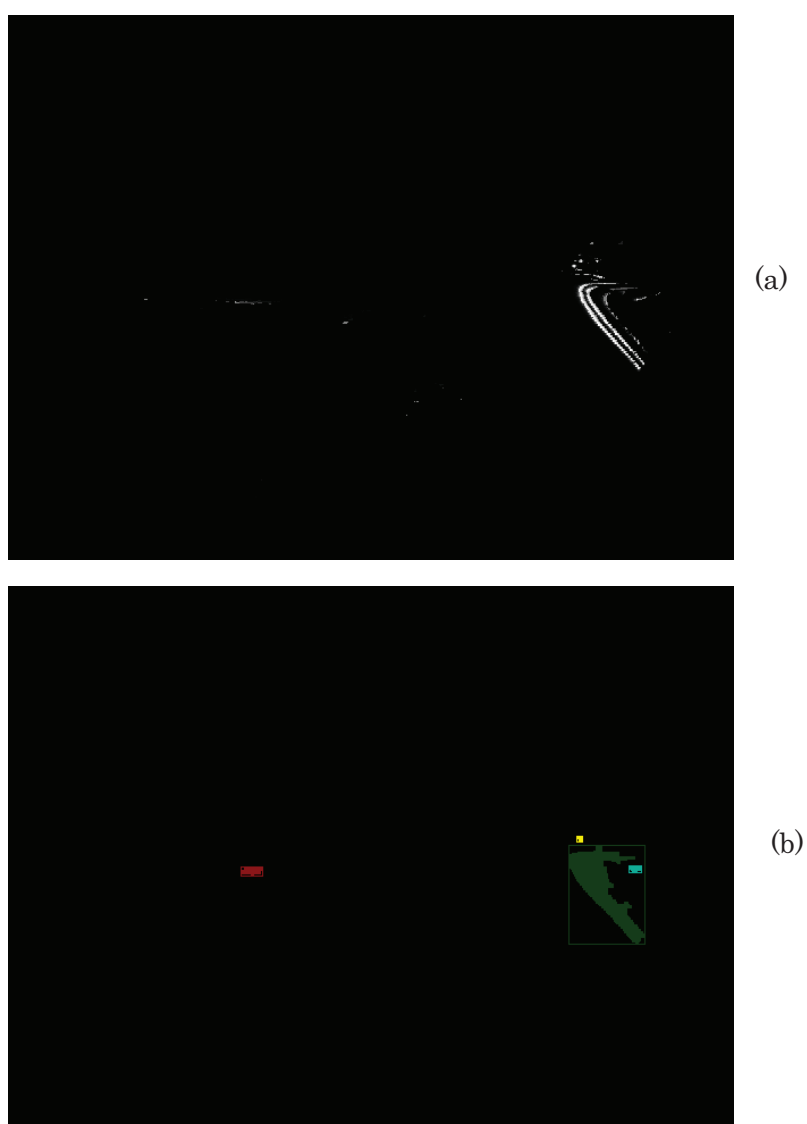


図 5.6 (a) BMHI 画像 (b) BMHI 画像の領域分割結果

次に、分割された領域から車両の運動候補領域を見つける。各領域に対して、外接矩形を算出して、その矩形の縦の長さ対横の長さの比（アスペクト比）と領域面積対長方形面積の比（画素密度）という2つの指標を用いて、車両運動領域候補を決定する。

次に、上記で得られた車両の候補領域の運動方向を推定する。BMHIを差分のビット画像の時系列とみなし、 $\tau$ 時間間隔内の時空間画像として解析する。 $\tau$ 時間間隔内で交差点における側面接近車両の運動が連続的直線運動であるため、本論文では時空間画像での車両領域中心の変化からその車両領域の運動方向を推定する。具体的に、本論文では分割領域内のBMHIを均等に上位ビットグループと下位ビットグループに分けてそれぞれの分割領域中心を計算し、その下位ビット画像における運動領域の中心から上位ビット画像における運動領域の中心へのベクトル方向を車両候補領域の運動方向とする。図5.7は分割領域中心の計算結果である。ただし、上位ビットグループの領域を青色で、下位ビットグループの領域を黄色で、重なる領域を水色で示しておく。また、上位と下位ビットグループの領域中心をそれぞれ緑色と赤色で示し、運動方向は赤点から黄点へ方向である。

最後に、上記で算出された車両候補領域の運動方向を、カメラの内部と外部パラメータを用いて事前に算出された背景の運動方向と比較する。背景の運動方向とほぼ反対になる運動方向を持つ車両候補領域を側面から接近車両領域として決定する。

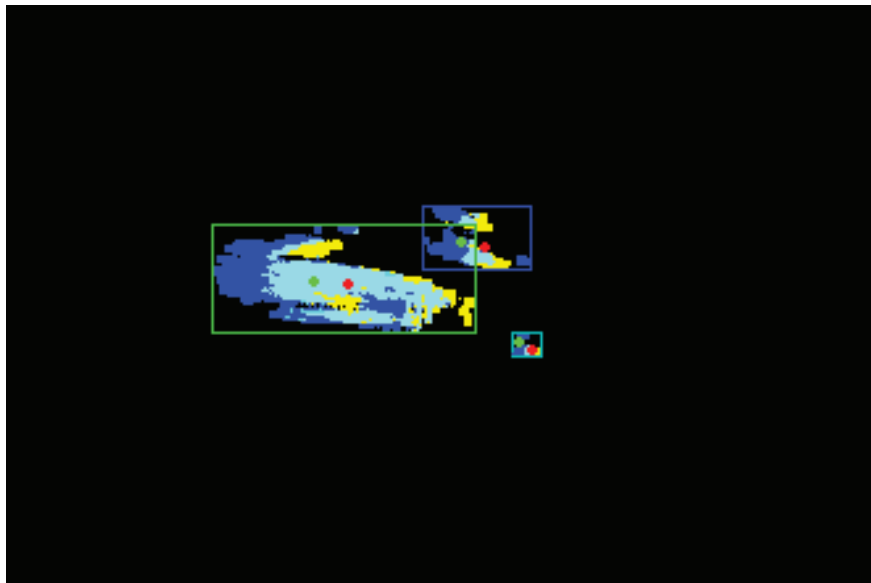


図 5.7 分割領域中心をビット分けて計算した結果

### 5.3.3 実験

本手法の有効性を検証するため、車載魚眼カメラから撮った  $640 \times 480$  解像度の映像を用いて CPU Intel Pentium D 2.8-GHz メモリ 512-MB のデスクトップパソコンで処理し、その結果を OpenCV[5]の MASK を利用した tMHI のアルゴリズムの処理結果と比較した。

提案手法 BMHI 表現による実験では、そのビット数を , 画素密度の閾値を 0.4, アスペクト比を 3, 領域内有効画素数を 200 とした。背景運動方向と検出車両運動方向はほぼ反対なので、運動領域の方向ベクトルと背景運動方向ベクトルがなす角度が  $90^\circ$  以上であれば、その運動領域が車両領域の候補とした。

図 5.8 は提案手法 BMHI と既存手法 tMHI との比較実験の結果を示す。tMHI での円は検出された運動領域を表し、中心からの線分はその運動領域の向きを表す。図 5.8 が示すように、提案手法は従来法より安定的に運動領域の向きを検出している。この実験のビデオデータを研究室のウェブサイト[6]に公開している。

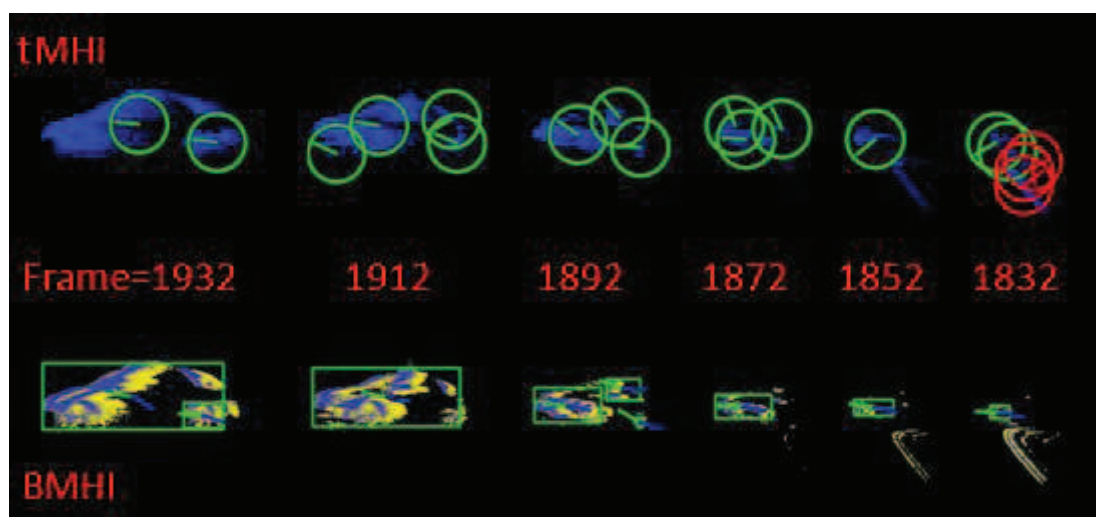


図 5.8 運動領域分割と推定方向を比べた結果

最後に 1000 フレームのデータを用いて tMHI と BMHI の処理速度のテストした結果を表 5.2 で示す。tMHI のサンプルプログラムの処理スピードはフレーム当たり平均 35 ms であった。提案手法では、ビット演算によりフレーム当たり平均 16 ms の処理スピードを得た（ここで運動履歴画像関する処理だけの時間が示され、他のビデオの読み込みと表示などの時間は含まれていない）。BMHI の処理時間は tMHI の半分以下になった。

表 5.2 処理時間を比べた結果

Method	The average processing speed of one frame
tMHI	35 ms
BMHI	16 ms

## 5.4 まとめ

本章では、バイナリ運動履歴画像による交差点における側面からの接近車両の検出手法を提案した。従来の運動履歴画像に比べると、バイナリ運動履歴画像ではその時間間隔内の運動履歴が完全に保存される。また、ビット演算を用いることにより、バイナリ運動履歴画像に対して比較的速い処理速度を得た。実験結果では、本提案手法は従来法に比べ、より高速にかつ正確に交差点における側面からの接近車両の運動領域を検出することができた。



## 参考文献

- [1] B.Leibe, K.Schindler, N.Cornelis, “Coupled Object Detection and Tracking from Static Cameras and Moving Vehicles”, IEEE Transactions on Pattern Analysis and Machine Intelligence, p1683-1698, Oct 2008
- [2] C.Du, Q.Wu, J.Yang, X.He, “Subspace Analysis Methods plus Motion History Image for Human Action Recognition” , Digital Image Computing: Techniques and Applications, p606-611, Dec 2008
- [3] F. Moutarde, B. Stanciulescu, A. Breheret, “Real-time visual detection of vehicles and pedestrians with new efficient adaBoost features”, IEEE International Conference on Intelligent Robots Systems, 2008.
- [4] GR.Bradoski, JW.Davis, “Motion segmentation and pose recognition with motion history gradients”, Machine Vision and Applications, p174-184,2002
- [5] H.Cheng, N.Zheng, C.Sun, “Boosted Gabor Features Applied to Vehicle Detection”, ICPR,p662-666,Aug 2006
- [6] J. Davis and A. Bobick, “The representation and recognition of human movement using temporal templates”, CVPR, pp.928-934, 1997
- [7] Q Yuan, A Thangali, V Ablavsky, “Learning a Family of Detectors via Multiplicative Kernels”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.99,2010.
- [8] S.Gehrig, C.Rabe, L.Krueger, “6D Vision Goes Fisheye for Intersection Assistance”, Computer and Robot Vision, p34-41,May 2008
- [9] S.Kamijo, K.Ikeuchi, M.Sakauchi, “Vehicle Tracking in Low-angle and Front-View Images based on SpatioTemporal Markov Random Field Model”, 8th World Congress on ITS, 2001
- [10] T.Teshima, H.Saito, S.Ozawa, K.Yamamoto, “Estimation of FOE Without Optical Flow for Vehicle Lateral Position Detection”, ICPR,p626-629,Aug 2006
- [11] Z.Sun, G.Bebis, R.Miller, “Quantized Wavelet Features and Support Vector Machines for OnRoad Vehicle Detection”, ICARCV, Dec 2002
- [12] Z.Yin, R.Collins, “Moving Object Localization in Thermal Imagery by Forward-backward MHI ” , CVPR Workshop,p133,June 2006
- [13] <http://sourceforge.net/projects/opencvlibrary/> (OpenCV 1.1)
- [14] <http://www.skybird.ele.tottori-u.ac.jp/BMHI.html>

## 第六章 結論

本論文では車載魚眼カメラを用いて運転支援を目指して、全天周画像道路地図登録および交差点における側面から接近車両検出に関する研究について述べた。具体的には、魚眼画像列と慣性センサー及び GPS の同期手法、全天周画像を表現する球面画像フォーマット SIF、バイナリ運動履歴画像、交差点での側面からの接近車両を検出手法を提案した。主な成果は次の通りである。

### (一) 全天周画像道路地図登録

全天周画道路地図を構築するためにそれぞれ独立に記録した一対の魚眼カメラで撮像された二つの魚眼画像列と慣性センサー及び GPS データを共通の運動情報により同期手法を提案した。

- 複数のカメラを同期する方法として従来は重複視野を利用していた。本研究で我々は重複視野がほとんど存在しない二つの魚眼カメラの間の運動の整合性を利用してカメラ間の同期を行っている。カメラ間の回転運動の相関は、推定されたロール、ピッチ、ヨーの三つの角度の相関係数の乗積で算出される。その有効性が実験結果から示される。
- 坂道や傾斜などの路面状況の緩慢な変化で生じる全天周画像の傾きを全天周カメラと慣性センサー同期することにより補正する手法を提案した。
- 最後に全天周画像をハイブリッド GPS と同期して、ほぼ等距離にマップの道路に沿った自動登録手法を提案した。

### (二) 全天周画像の表現法

全天周道路地図の多くの全天周画像データを有効的に管理するために本研究では二次元配列で表現した球面画像フォーマットを提案した。SIF 画像における隣接画素の探索や画素と球面画像との間の対応関係を算出するアルゴリズムも提案した。この球面画像フォーマットの主な成果は次の通り。

- 本球面画像フォーマットはキューブマップよりほぼ均一画像を持ち、入力画像の一対の魚眼画像からなる全天周画像内の冗長なデータをなくしてコンパクトな二次元配列で表現することができる。
- 有効な SIF 画像における隣接画素の探索アルゴリズムと LUT (Look-up table) の導入によってより、全天周画像から透視画像表示を高速で生成する実験結果を示した。
- 全天周カメラの撮像部では、内部で全天周画像を SIF 画像に変換し出力すると、ユーザーが扱うのは SIF 画像になる。その SIF には冗長なデータが含まれない。カメラ内部パラメータにも依存しない。

### (三) 交差点における接近車両検出

バイナリ運動履歴画像による交差点における側面からの接近車両の検出手法を提案した。主な成果は次の通り。

- 新しい時間テンプレートとしてのバイナリ運動履歴画像を提案した。従来の運動履歴画像に比べると、バイナリ運動履歴画像ではその時間間隔内の運動履歴が完全に保存される。また、ビット演算を用いることにより、バイナリ運動履歴画像に対して比較的の高い処理速度を得た。
- 魚眼画像で自車両と接近する車両の運動方向が反対である特徴を利用して側面からの接近車両検出手法を提案している。実験結果では、本提案手法は従来法に比べ、より高速にかつ正確に交差点における側面からの接近車両の運動領域を検出する結果を得た。

### (四) 今後の課題

第三章では、全天周画道路地図を構築する手法を提案したが、画像録画時のフレーム落ちや GPS 誤差がセンサーデータ間の同期や地図登録精度に影響を与える。その対処法は課題として残されている。

第四章では、球面画像フォーマット SIF を提案して、球面画像から透視画像生成の例を示したが、SIF 画像に対する様々な処理法の開発は将来の課題である。

第五章では、交差点における側面からの接近車両の検出手法について、安全運転に対してより遠く離れた接近車両の検出が必要である。また、検出された側面車両の運動領域からパターン認識手法を用いて側面車両を同定するのは、今後の課題として挙げられる。

## 謝辞

本研究を遂行するにあたり，終始御懇切なご指導と御鞭撻を賜りました鳥取大学工学部電気電子工学科李仕剛教授に深く感謝致します．研究から生活までいろいろな面で同教授のいつもの親切なご指導，ご協力をいただき心よりお礼申し上げます．

本論文の執筆にあたり，多くの懇切な御指導を賜りました鳥取大学工学部電気電子工学科伊藤良生教授，中西功准教授に深く感謝致します．

また，本研究を進めるにあたり，終始御協力を頂きました鳥取大学工学部電気電子工学科電子素子応用工学研究室の皆様方に深く感謝致します．

最後に，筆者の研究生活を支えて下さった家族に感謝致します．

## 発表論文

### 学術論文

- [1] 海鷹, 李仕剛, 水野友人, “道路地図作成のための全天周画像列と慣性センサーおよびGPSデータの同期”, 画像電子学会誌, Vol.39, No.1, pp.45-52, 2010年1月.
- [2] 海鷹, 李仕剛, “バイナリ運動履歴画像による交差点における側面からの接近車両の検出”, 電子情報通信学会誌, Vol.J94-D, No.2, pp.501-504, 2011年2月.
- [3] Shigang Li, Ying Hai, “Fast Generation of View-Direction-Free Perspective Display from Distorted Fisheye Image”, IEICE Transactions on Information and Systems, Vol. E92D, No.8, pp.1588-1591, August 2009
- [4] Shigang Li, Ying Hai, “Easy Calibration of a Blind-Spot-Free Fisheye Camera System using a Scene of a Parking Space”, IEEE Transactions on Intelligent Transportation Systems, (accepted)

### 国際会議

- [1] Ying Hai, Shigang Li, Yuto Mizuno, “Synchronization of Full-view Image and GPS Data for Route Map Building”, MVA2009 IAPR Conference on Machine Vision Applications, pp.78-81, May 2009
- [2] Shigang Li, Ying Hai, “A Full-View Spherical Image Format”, 2010 20th International Conference on Pattern Recognition, pp. 2337-2340, August 2010
- [3] Shigang Li, Ying Hai, “Estimating Camera Pose from H-Pattern of Parking Lot”, IEEE International Conference on Robotics and Automation, pp. 3954-3959, May 2010