

## 流星の自動検出プログラムの開発

竹田 育弘 井口 幸洋  
明治大学理工学部情報科学科  
y-takeda@cs.meiji.ac.jp

## 背景と目的

## 流星自動検出の必要性

現在は人間の目を使った観測が主流  
見間違いが多い、多大な労力



データチェックを自動化

2005/10/6

日本天文学会2005年秋季年会



3

## 目的

- ◆ 係数観測の補助ツール
- ◆ 取得する情報
  - ◆ 出現時刻
  - ◆ 出現・消滅位置
  - ◆ 等級
  - ◆ 継続時間
  - ◆ 単位時間当たりの個数

2005/10/6

日本天文学会2005年秋季年会



4

## アルゴリズムについて

## 方針

- ◆ 流星の特徴をつかむ
  - ◆ 流星は直線状に移動する光点(動画上)
  - ◆ 徐々に明るくなる etc.
- ◆ マスク減算による恒星の消去
  - ◆ フレームの加算平均によりマスクを作成
- ◆ 継続時間による制約
  - ◆ 恒星、人工衛星、飛行機、その他発光現象等と区別
- ◆ 眼視チェックと同等程度の検出精度を目指す
- ◆ 実時間以下の処理を目指す
- ◆ Windows搭載ノートパソコンで動作

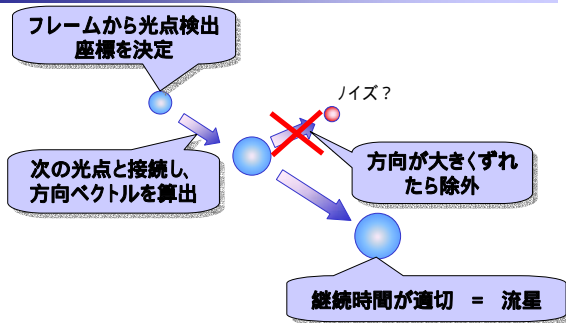
2005/10/6

日本天文学会2005年秋季年会



6

## 基本アルゴリズム



2005/10/6

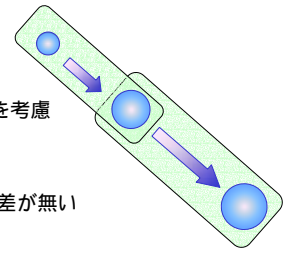
日本天文学会2005年秋季年会



7

## 流星判定条件

- ◆ **光点接続**
  - ◆ 光点間の距離が最小
  - ◆ 徐々に明るくなる
  - ◆ 大気の揺れによる減光を考慮
- ◆ **流星の継続**
  - ◆ 方向がほぼ同じ
  - ◆ ベクトルの長さに大きな差が無い
- ◆ **流星決定**
  - ◆ 継続時間が適切(0.2秒~3秒)



2005/10/6

日本天文学会2005年秋季年会



8

## パラメータ

- ◆ **マスク**
  - ◆ マスクはフレームの加算平均で作成
  - ◆ 作成に使用する枚数を指定
- ◆ **光点検出しきい値**
  - ◆ 光点を検出する際の輝度を指定
  - ◆ 自動設定も可能
- ◆ **ベクトル**
  - ◆ 光点接続ベクトルの長さの上限を指定
- ◆ **継続時間**
  - ◆ 継続時間の最小・最大値を指定

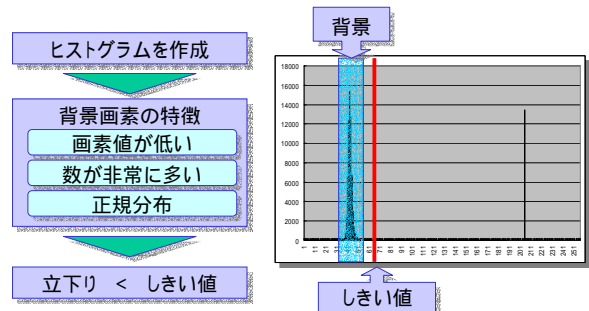
2005/10/6

日本天文学会2005年秋季年会



9

## 光点検出用しきい値の設定



2005/10/6

日本天文学会2005年秋季年会



10

## 光点座標の決定

光点内で**最も光度の高い点**を光点の座標とする  
(複数ある場合はその中間をとる)



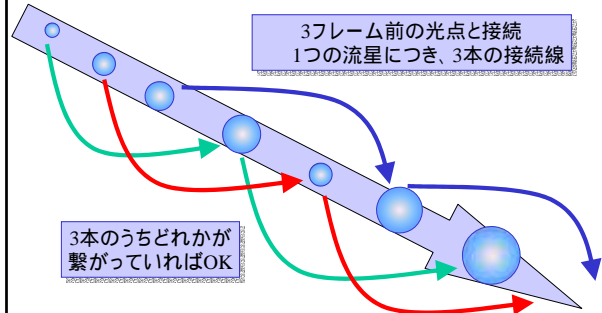
2005/10/6

日本天文学会2005年秋季年会



11

## 流星の継続



2005/10/6

日本天文学会2005年秋季年会



12

## 自動検出システムと実験結果

## インターフェース



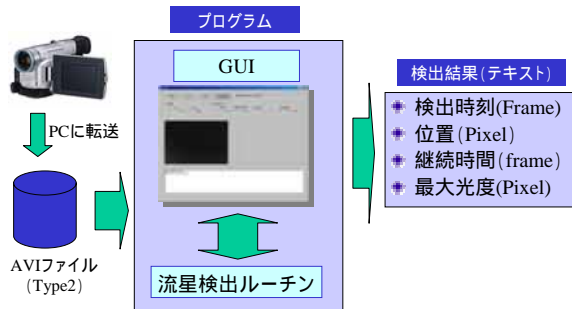
2005/10/6

日本天文学会2005年秋季年会



14

## プログラムの仕様



2005/10/6

日本天文学会2005年秋季年会



15

## 予備実験

- 使用PC
  - Pentium4 2.4GHz, 512MBの自作PC
  - 実時間以内での検出が可能
- 入力ファイル
  - しし座流星群、画角:対角約100°、ししの大鎌方向
- 検出結果(7個中6個を検出, 3分弱の短い動画)
  - ほぼ同時に出現した2個の流星も検出
  - 画面端で切れた流星あり
  - 画角が輻射点方向のため、短経路流星が多いが検出できた

2005/10/6

日本天文学会2005年秋季年会



16

## 出力データ形式

番号	出現時刻 (frame)	出現位置 (Pixel)		消滅位置 (Pixel)		最大光度 (Pixel)	継続時間 (frame)
		x	y	x	y		
1	1198	138	134	145	133	20	7
2	15	74	77	74	68	12	6

2005/10/6

日本天文学会2005年秋季年会



17

## 出力データの変換

- 出現時刻: フレーム 時:分:秒
- 出現・消滅位置: ピクセル 赤緯・赤経
- 最大光度: 面積 等級
- 継続時間: フレーム 秒

上記のような変換が必要

2005/10/6

日本天文学会2005年秋季年会



18

## 今後の予定

---

- ◆ 検出精度の評価
- ◆ 条件を煮詰める 精度向上
- ◆ ユーザーインターフェース改良
- ◆ さらに効率を追求 高速化
- ◆ 安定性の向上、エラー処理追加
- ◆ ダイレクトキャプチャ、リアルタイム処理
- ◆ 検出結果の変換ツール作成
- ◆ Web上での公開