

# 重なり滞留を起こす 複数牛個体の高速追跡法

宮崎大学 工学部 情報システム工学科

67180570 AHMAD AMINNIN BIN MAT NOOR (アミン)

指導教員 椋木 雅之 教授

2022/02/16

# 研究背景

---

肉牛や乳牛の酪農では、  
農業資源である牛の管理が必要



牛の行動を自動で監視することがますます重要になり、情報技術への依存度が高まっている

# 研究背景

---

従来の複数物体追跡：

常に動き続ける物体が追跡対象

例：通路を歩く人の追跡



重なり滞留をあまり考慮していなかった

# 研究目的

---

**重なり滞留**が多く発生する牛の群れに対して、

従来手法(SORT)の**高速な追跡処理**を保ちつつ



SORTの**追跡の誤り** (IDスイッチ) 数を  
**削減するために改良**

# 従来の複数物体追跡 : SORT

# Simple Online and Real Time Tracking (SORT)

4つのプロセスを経て、  
複数物体追跡を実現する

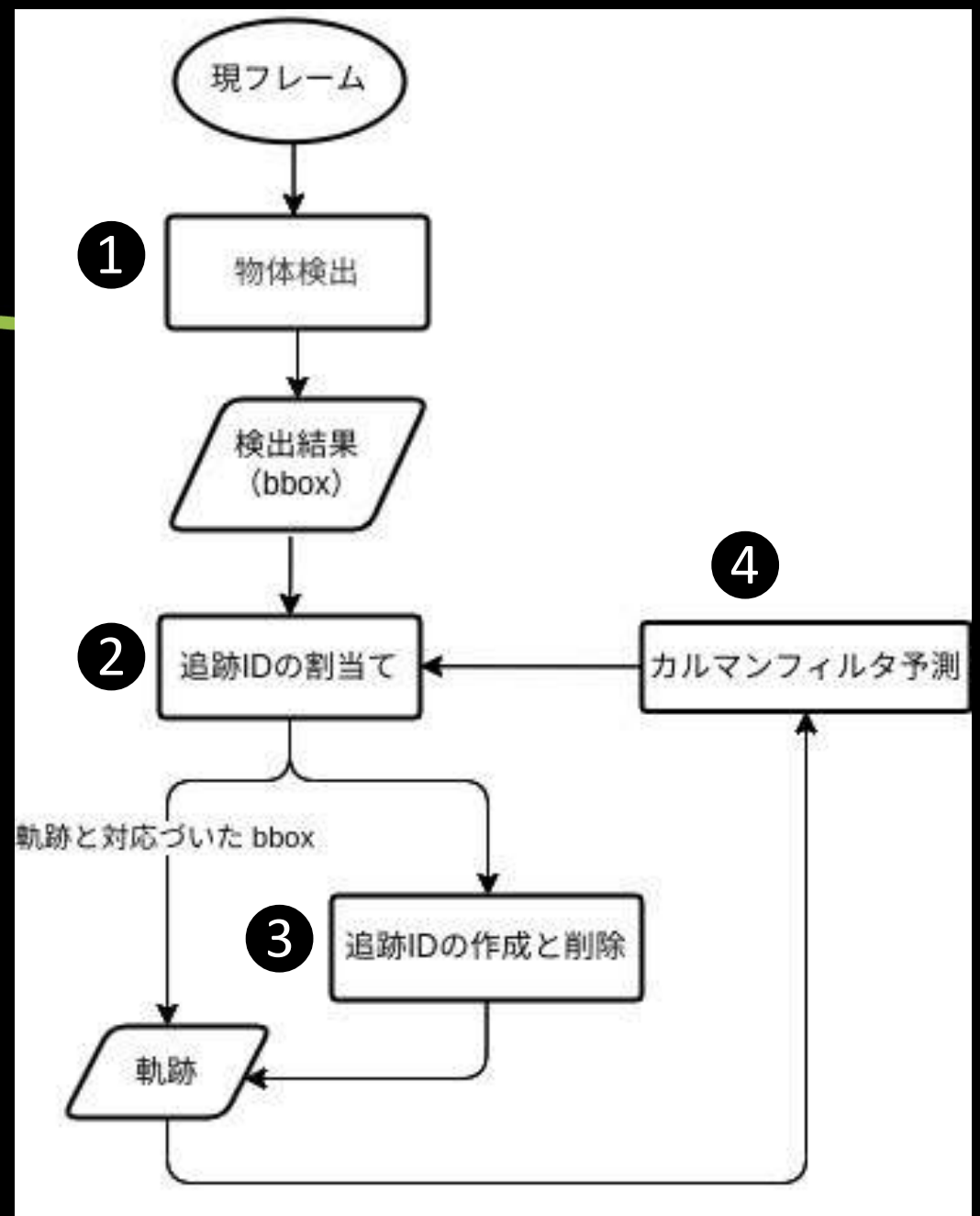
1. 物体検出

2. 追跡IDの割り当て

3. 追跡IDの作成と削除

4. カルマンフィルタ予測

高速な処理で追跡できる



# 1. 物体検出

各フレームに対して物体検出を適用する。

物体検出結果は、物体を囲う矩形  
(バウンディングボックス；bbox) で表現される。

bbox

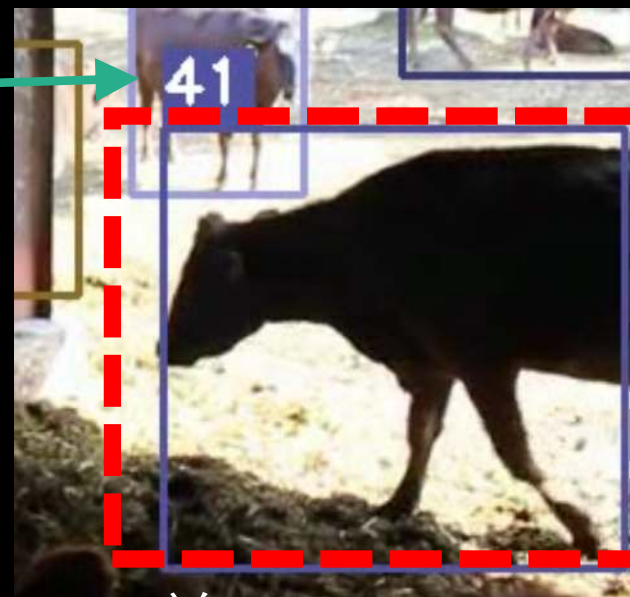


## 2. 追跡IDの割り当て

bboxに対して、軌跡の予測位置と比較して、ハンガリアン法により追跡IDを割り当てる



現フレーム

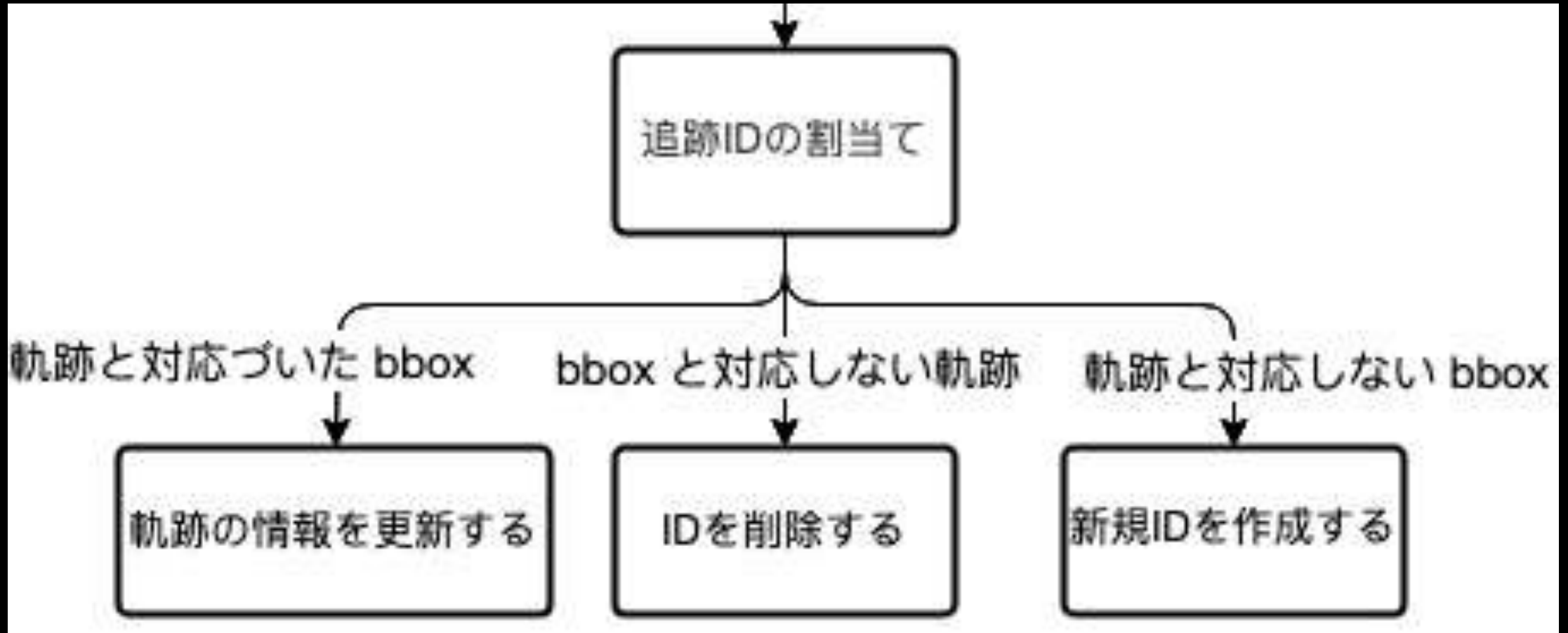


前フレーム

軌跡の  
予測位置



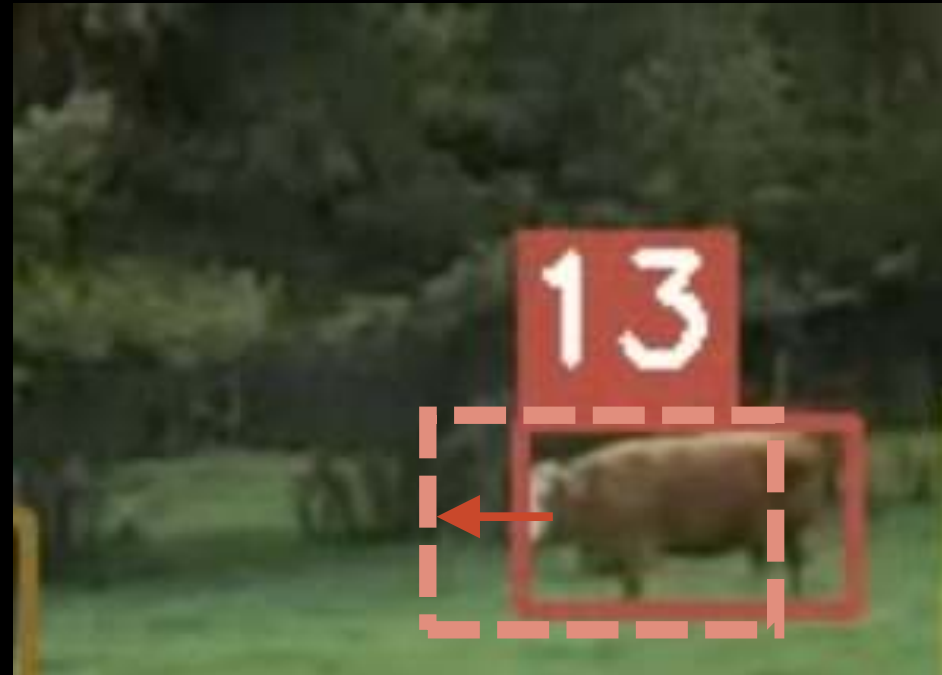
### 3. 追跡IDの作成と削除



## 4. カルマンフィルタ予測

---

次フレームの処理の前に、  
カルマンフィルタで次フレームでの位置を予測する



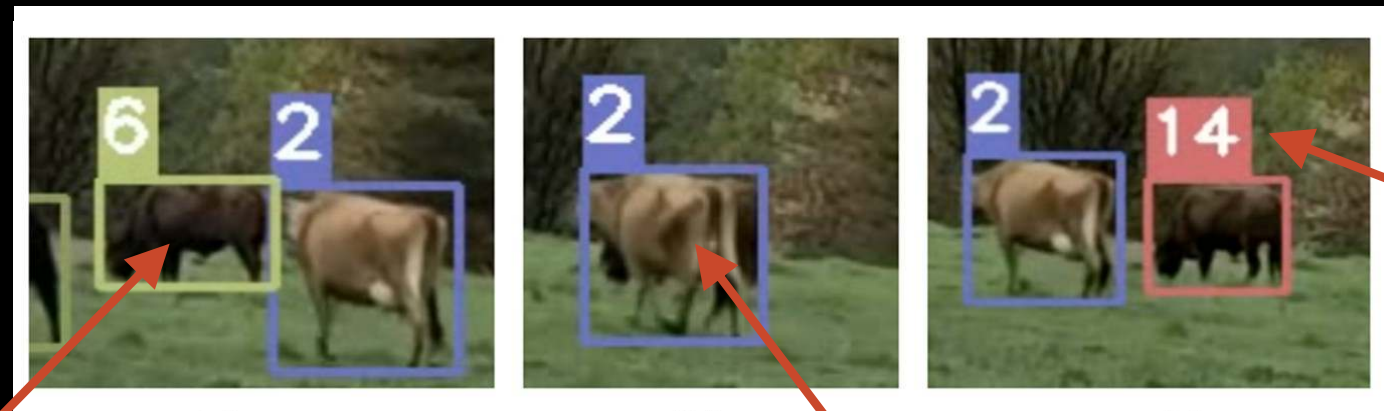
# SORTの問題点

# SORTの問題点

SORTは追跡処理の簡易性と高速性を重視している



重なり滞留によるIDスイッチを考慮していない



滞留

重なり

IDスイッチ

12

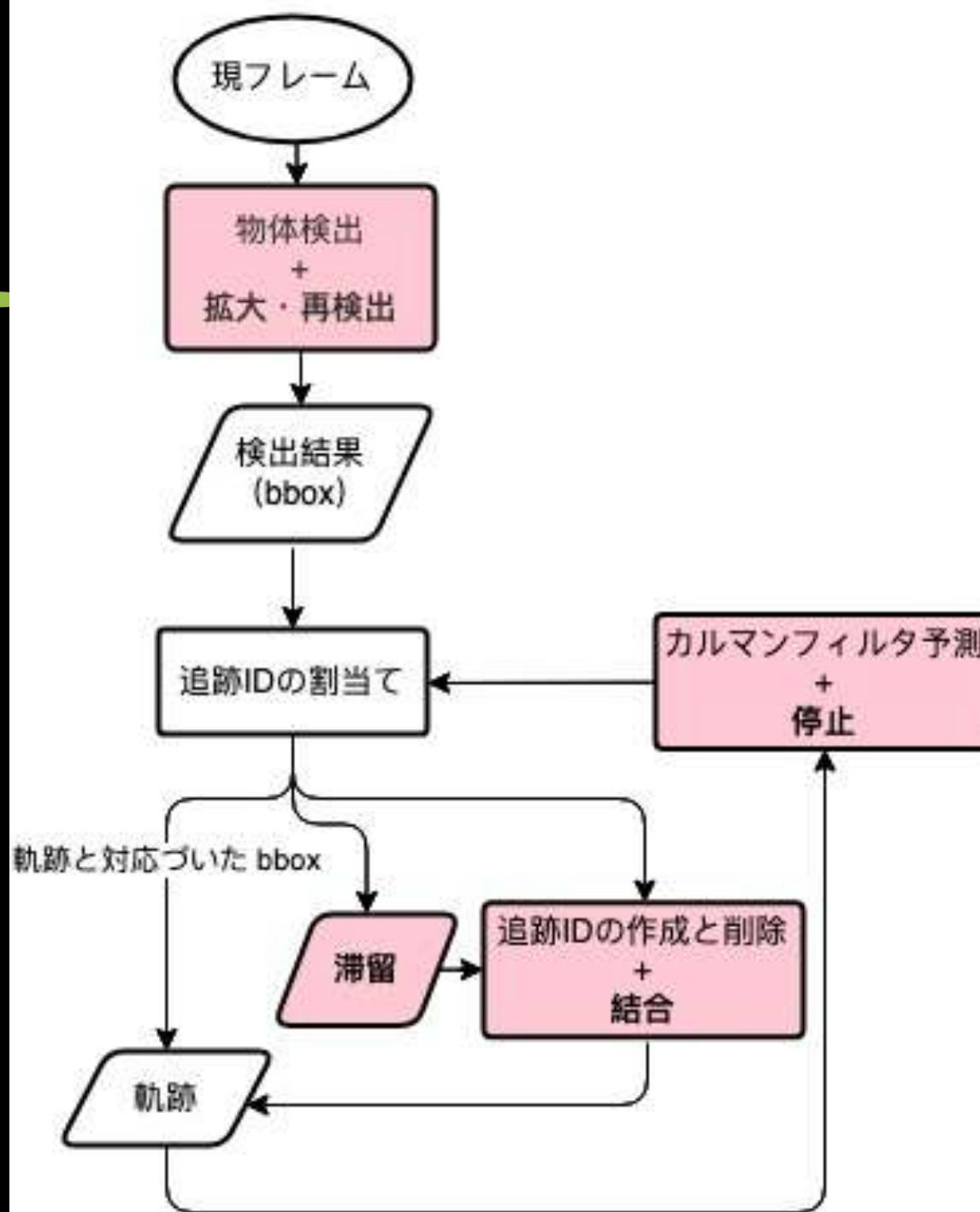
# 重なり滞留への対処： 提案手法

# 提案手法 の流れ

SORTの高速  
な追跡処理を  
保ちつつ



SORTのIDス  
イッチ数を削  
減する

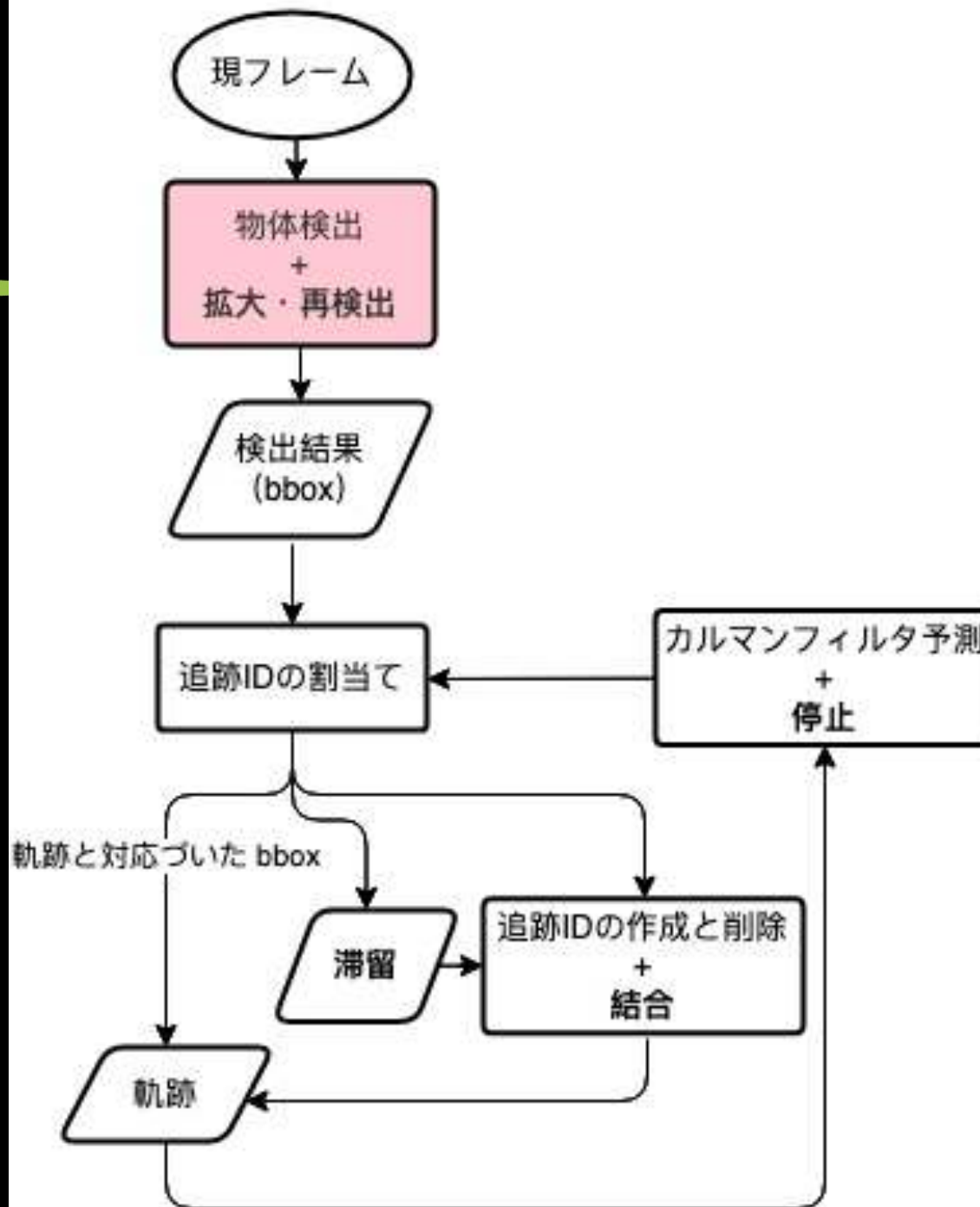


# 提案手法 の流れ

SORTの高速  
な追跡処理を  
保ちつつ



SORTのIDス  
イッチ数を削  
減する



# 1.画像の拡大と再検出

遠くに小さく撮影されている牛では、  
検出に失敗することが多い



画像を拡大して再検出することで、  
検出失敗を低減する



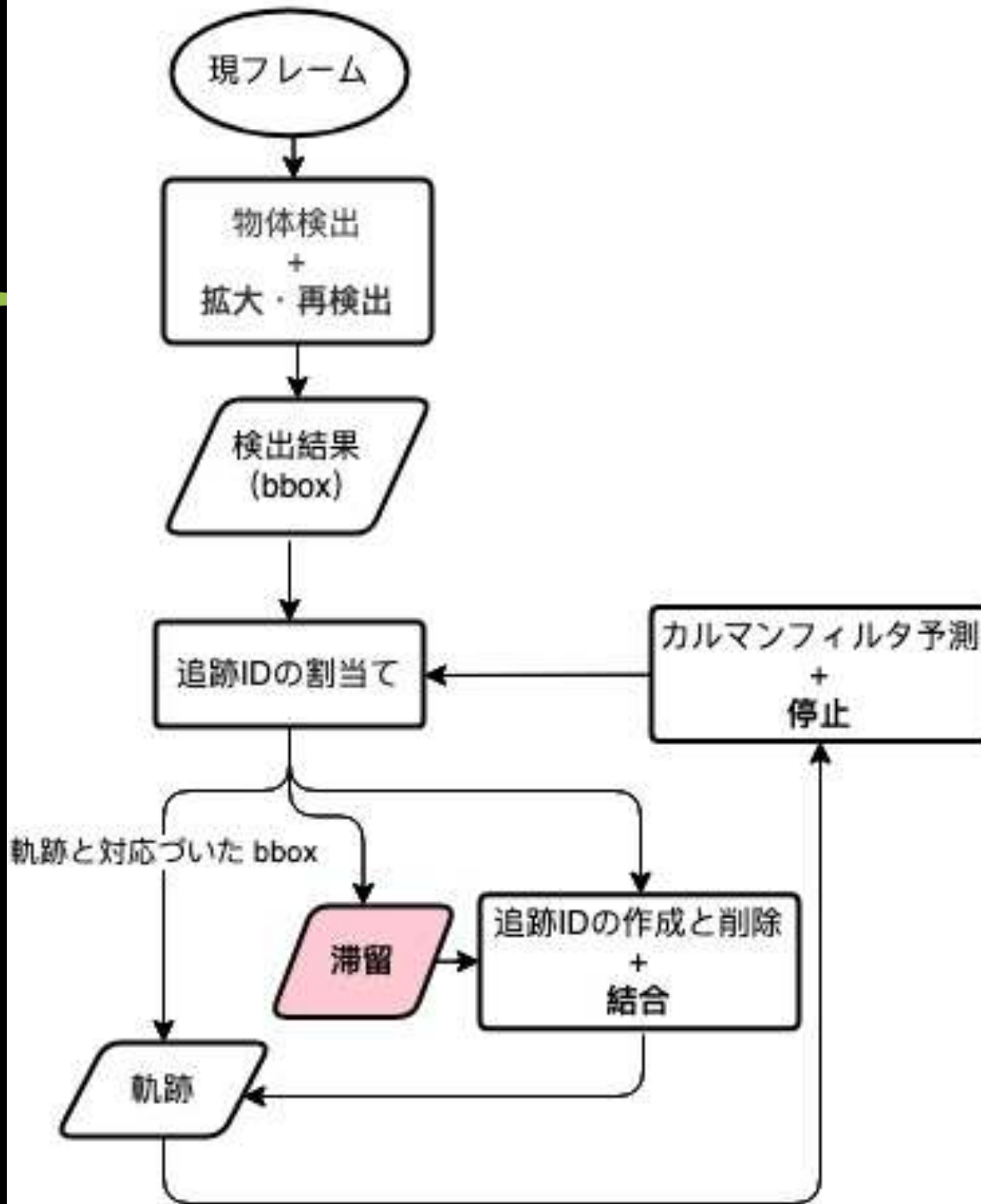


# 提案手法 の流れ

SORTの高速  
な追跡処理を  
保ちつつ



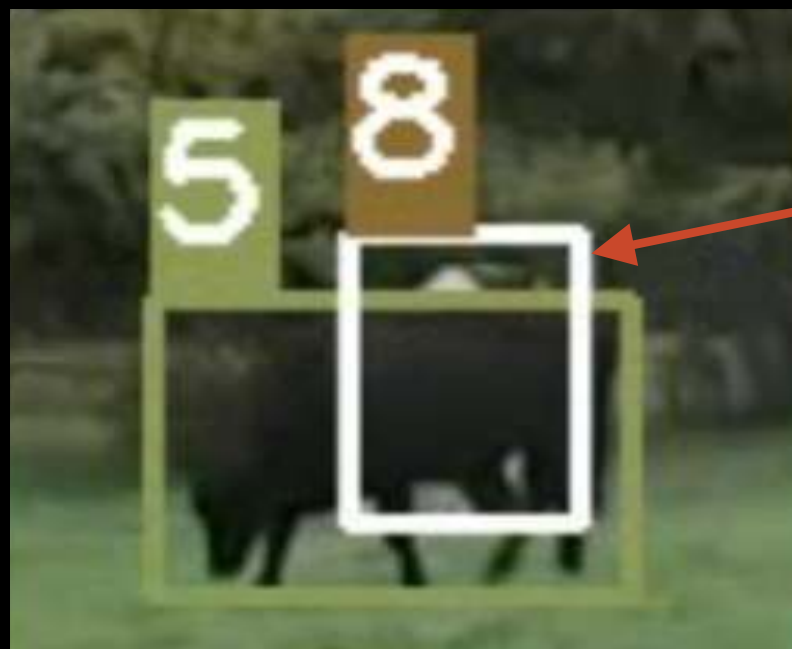
SORTのIDス  
イッチ数を削  
減する



## 2. 「滞留」状態の導入

---

長時間滞留している牛についても、  
継続して追跡が行えるようにする



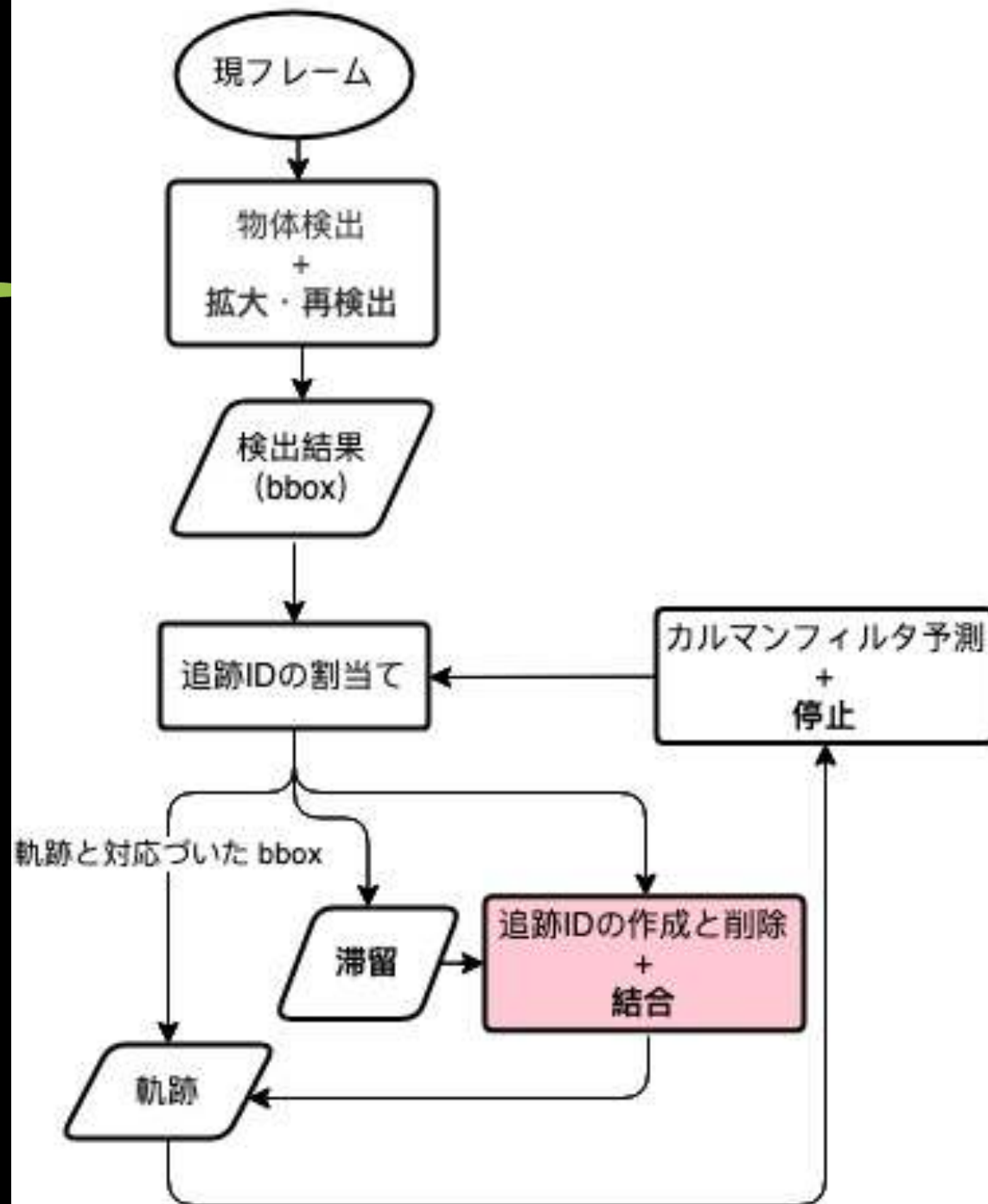
「滞留」状態

# 提案手法 の流れ

SORTの高速  
な追跡処理を  
保ちつつ

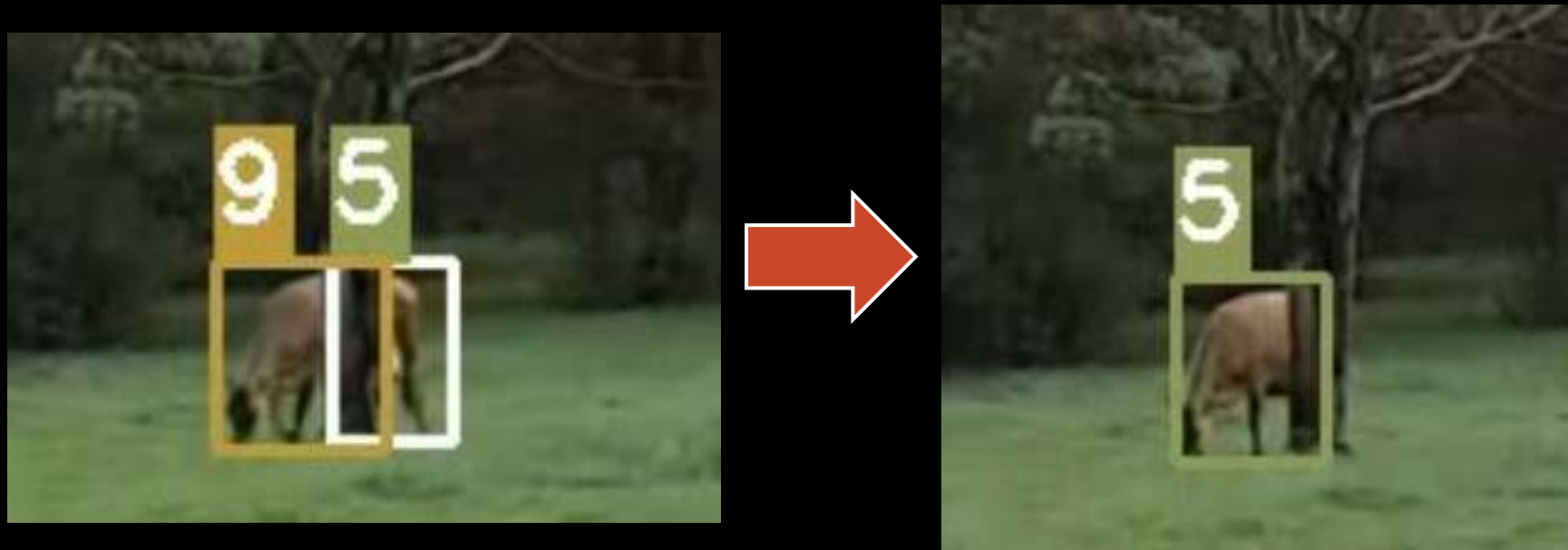


SORTのIDス  
イッチ数を削  
減する



### 3. 「滞留」状態での追跡データの保持と追跡IDの結合

「滞留」状態の軌跡に対し、近くに新規IDの bbox が検出された場合、追跡IDを結合して一つにする

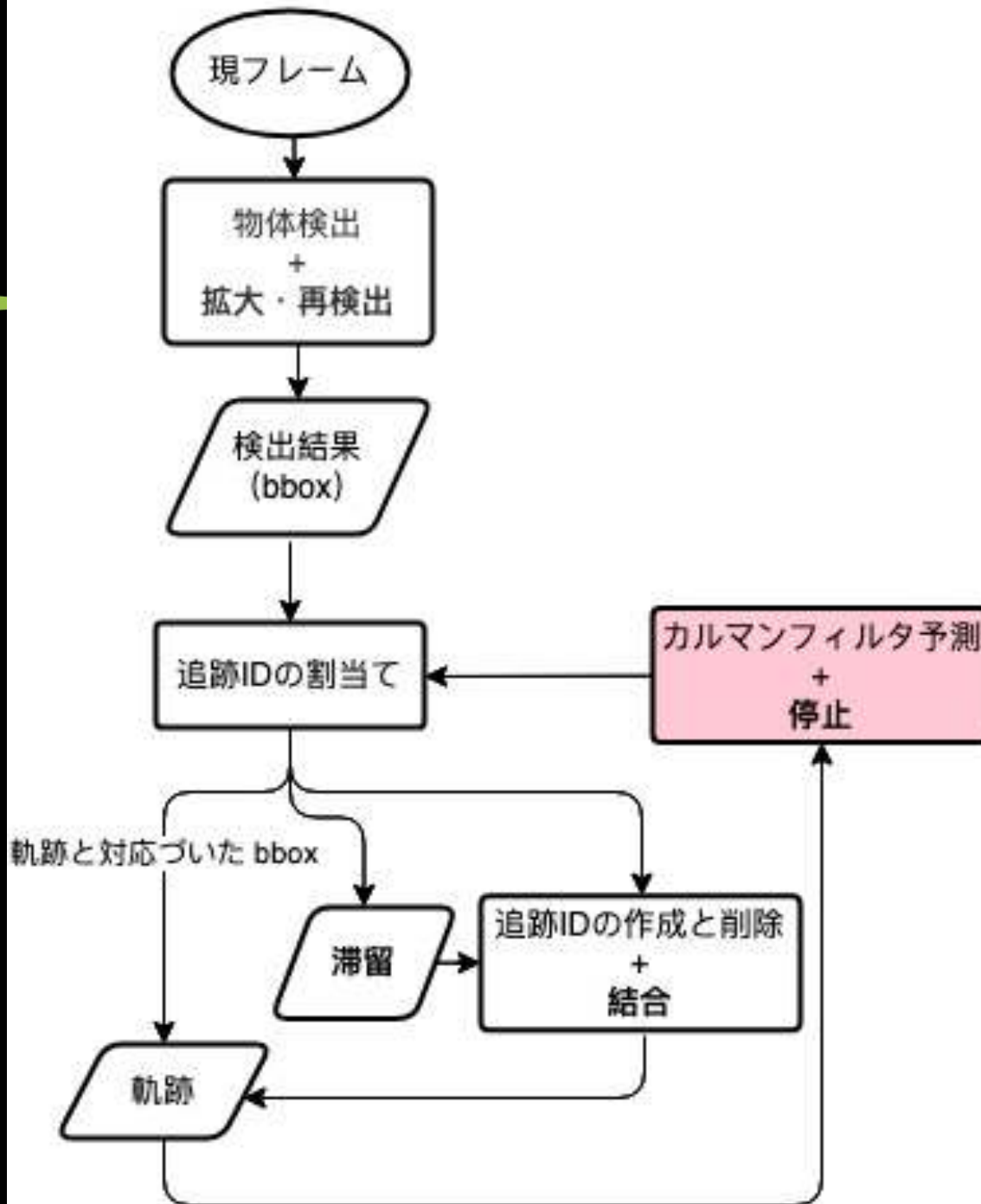


# 提案手法 の流れ

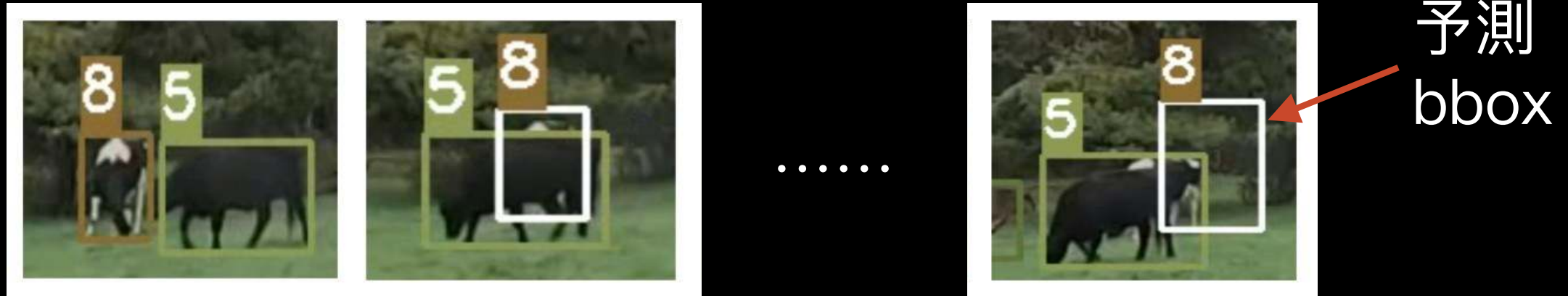
SORTの高速  
な追跡処理を  
保ちつつ



SORTのIDス  
イッチ数を削  
減する



# 4.カルマンフィルタ位置予測の停止



「滞留」状態の軌跡に対して、等速運動で長期間予測を行うと、滞留している牛の位置から予測位置が大きく外れてしまう



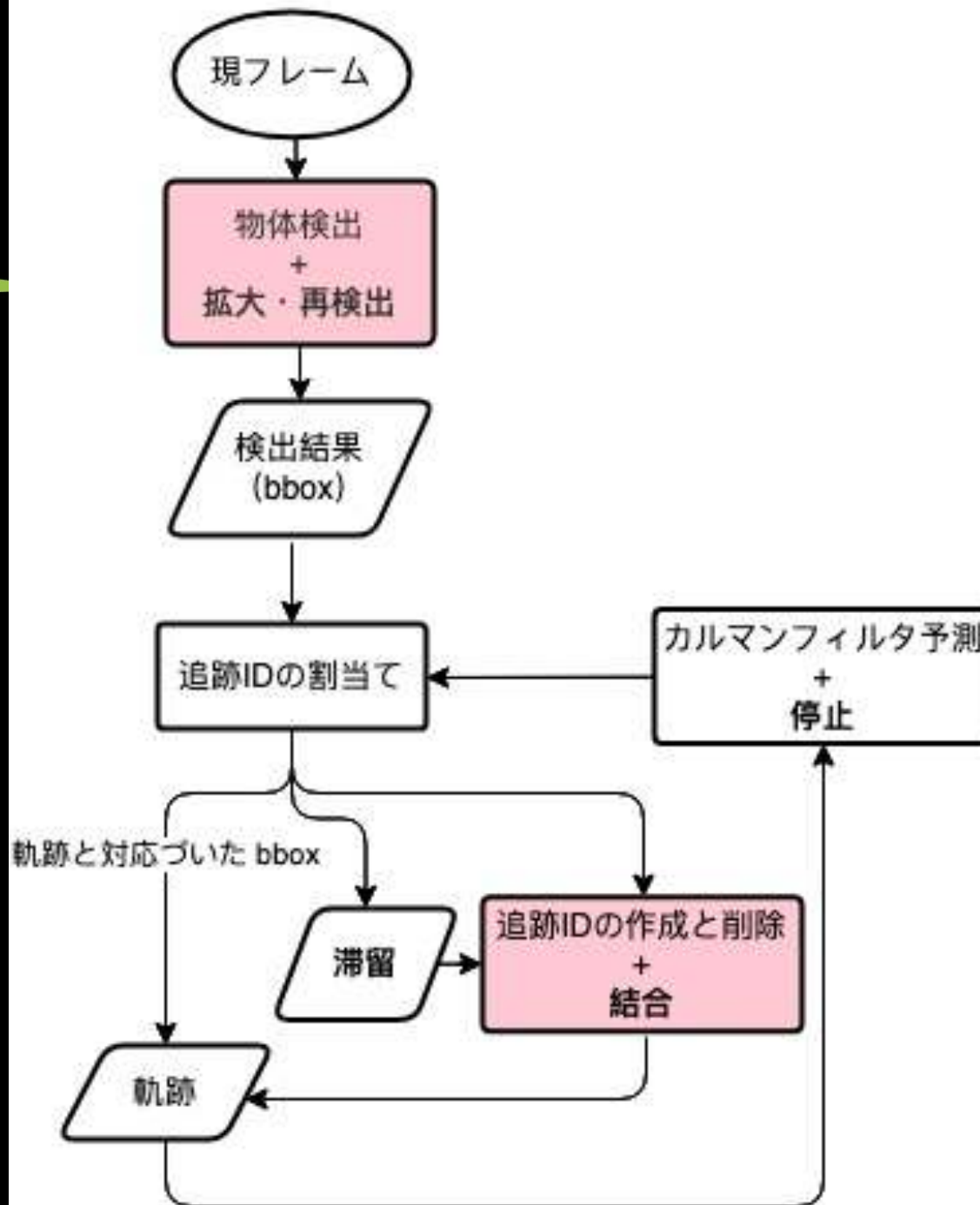
予測を停止して、これを避ける

# 提案手法 の流れ

SORTの高速  
な追跡処理を  
保ちつつ



SORTのIDス  
イッチ数を削  
減する



# 5.リフレッシュレート

---

精度と処理速度のトレードオフを考慮して



画像の拡大と再検出、および追跡IDの結合は、一定間隔離れた時間（リフレッシュレート）毎に行う

リフレッシュレート = 60



# 実験

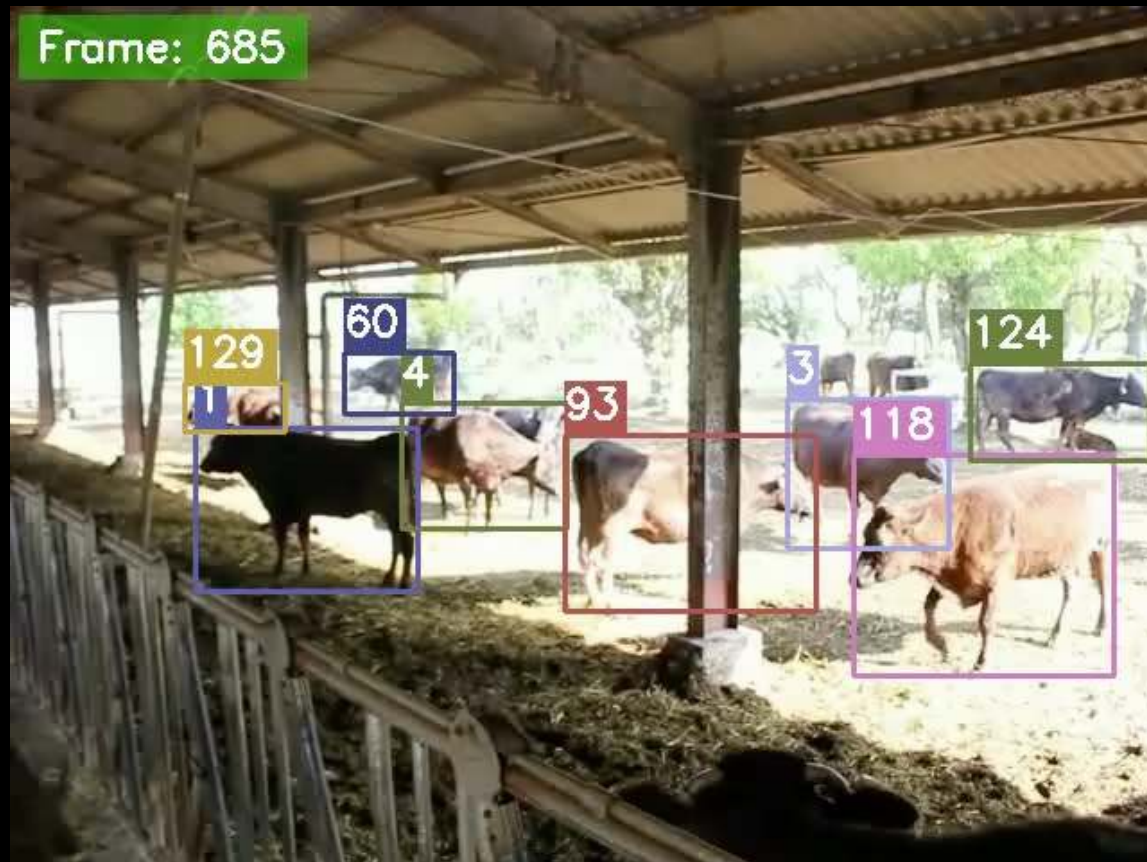
# 評価実験

---

3分程度の長さの動画中から29秒（60 FPS、1772フレーム）を切り取り、処理対象とした

SORTと DeepSORT (SORT の性能を向上させるために外観情報も使用する)と比較する

# SORTの追跡結果 36.52 FPS



# DeepSORTの追跡結果 21.48 FPS

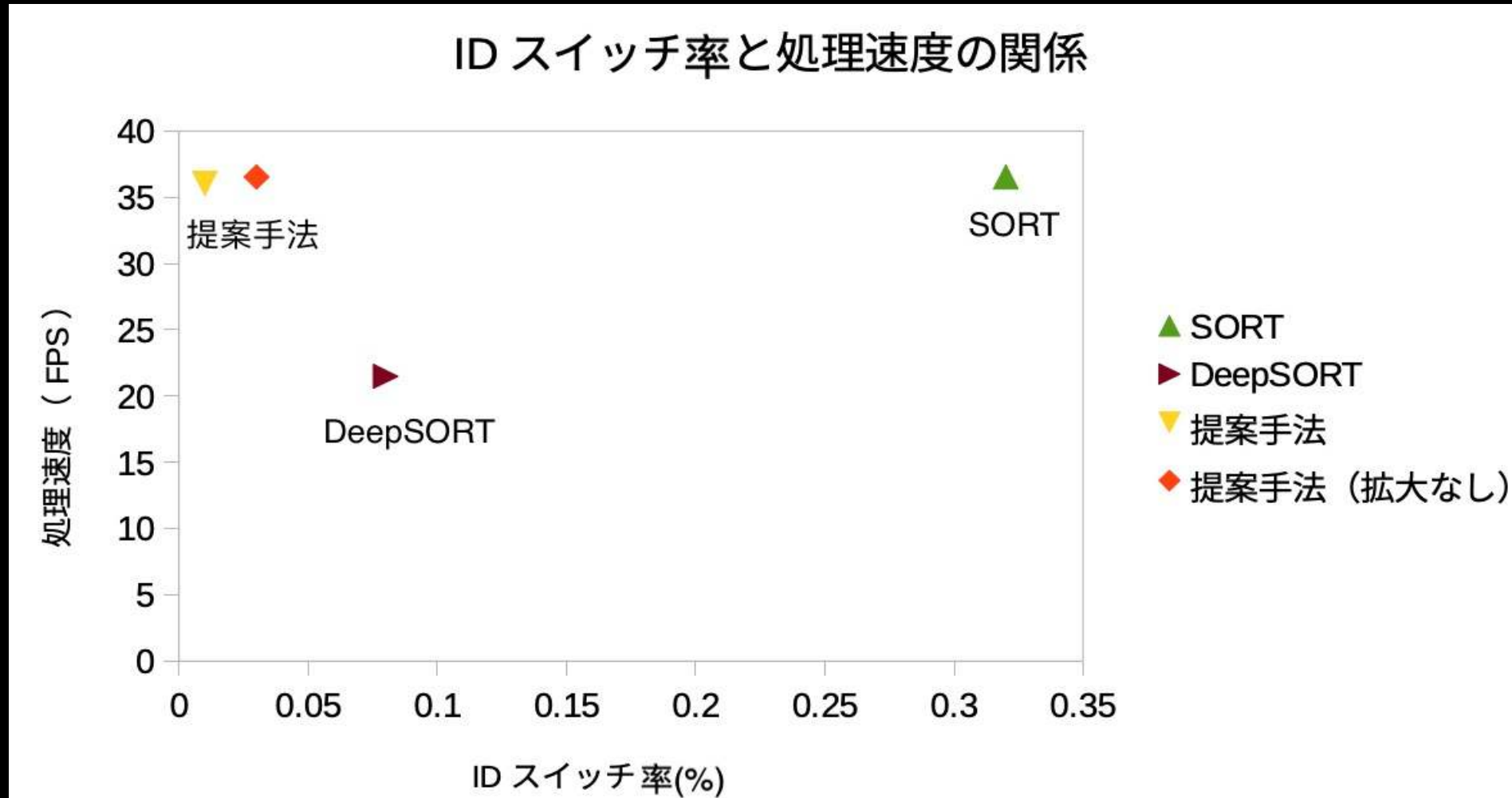


# 提案手法の追跡結果 36.04 FPS





# 全体の結果



# まとめ

---

重なり滞留が多く発生する牛の群れに対して



提案手法はSORTの高速性を維持しつつ、  
DeepSORTを超える追跡精度を達成した

## 今後の課題

さらなる検出精度の向上