

열화상 카메라 기반 주차면 입출차 감지 알고리즘 설계·구현

바닥에서 천장 방향으로 설치한 열화상 카메라에서 차체 하부가 천장 패턴을 가리는 특성을 이용해 점유 여부를 판단했습니다. 절대 온도 기준의 한계를 NCC 패턴 매칭과 ROI baseline 보정으로 해결했습니다.

도메인

열화상 카메라 기반 실시간 주차면 입출차 감지

담당 역할

알고리즘 설계 + 실시간/오프라인 시스템 전담 구현

기간

약 2개월+ / 알고리즘 5세대 반복 개선

산출물

실시간 감지 **GUI**, 오프라인 분석 **GUI**, 학습 모델 포맷, 기술 발표 자료

프로젝트 개요

카메라를 바닥에서 천장 방향으로 설치하면, 차량이 없을 때는 천장 패턴이 보이고 차량이 있으면 차체 하부가 시야를 가립니다. 이 프로젝트는 **온도 절대값**이 아니라 **패턴 유사도 변화**로 점유 여부를 판정하도록 접근을 바꾼 경험입니다.

열화상은 시간대·계절에 따라 전체 온도가 이동하고, ROI 위치와 질감에 따라 기준선이 달라집니다. 따라서 단순 임계값이 아니라 정규화, NCC, 상태 머신, SNR 기반 baseline 예측을 결합해 안정적인 입출차 이벤트를 만들었습니다.

기술 스택

Python

NumPy

OpenCV

WebSocket

tkinter

matplotlib

npz

NCC

상태 머신

알고리즘 설계 - 문제·원인·해결·결과

1. 절대 온도 판정의 실패 → NCC 패턴 매칭

문제

온도 평균 임계값으로 점유를 판정했지만, 아침/오후와 계절 변화에 따라 온도가 통째로 이동해 오 판이 잦았습니다. 분포 너비로 바뀌어도 실내 천장은 균일해 차체와 구분이 어려웠습니다.

해결

각 프레임에서 평균을 제거하고 표준편차로 정규화한 뒤, 학습 reference와의 패턴 상관도인 NCC를 판정 신호로 채택했습니다.

결과

정규화가 온도의 전역 이동을 상쇄해, 전체 온도가 올라가거나 내려가도 NCC 기준의 패턴 유사도는 안정적으로 유지되었습니다.

```
def normalize(t):  
    return (t - t.mean()) / t.std()  
  
def ncc(a_norm, b_norm):  
    return float(np.mean(a_norm * b_norm)) # -1 ~ 1 pattern similarity
```

2. 노이즈로 인한 상태 진동 → 상태 머신

문제

NCC가 순간 노이즈로 튀면 점유/빈 상태가 반복적으로 흔들려 허위 입출차 이벤트가 발생했습니다.

해결

EMPTY → ENTERING → PARKED → EXITING 상태 머신을 도입하고, 일정 시간 연속으로 안정될 때만 상태를 확정했습니다.

결과

단일 프레임의 순간값은 무시되고 실제 입출차 흐름만 이벤트로 기록되어 이벤트 신뢰도가 높아졌습니다.

3. ROI마다 임계값을 다시 잡아야 하는 문제 → SNR 기반 baseline 예측

문제

빈 상태의 NCC 값이 ROI마다 0.6~0.9처럼 다르게 형성되어, ROI를 옮길 때마다 임계값을 새로 잡아야 했습니다. 원인은 ROI의 질감과 노이즈 차이였습니다.

해결

학습 시 픽셀별 노이즈 σ 맵을 저장하고, ROI 마스크가 주어지면 신호 세기와 노이즈 세기로 baseline B를 예측했습니다. 이후 $occ = 1 - NCC / B$ 로 환산해 ROI별 차이를 상쇄했습니다.

결과

질감이 강한 ROI와 밋밋한 ROI 모두 같은 점수 범위로 환산되어, 하나의 임계값으로 여러 ROI를 판정할 수 있게 되었습니다.

```
ref = frames.mean(0)
ac = frames - frames.mean(axis=(1, 2), keepdims=True)
sigma = ac.std(0)

def predict_baseline(ref, sigma, mask):
    sv = ref[mask].var()          # ROI pattern strength
    nv = (sigma[mask] ** 2).mean() # ROI temporal noise
    return 1 / sqrt(1 + nv / sv)

occ = 1 - NCC / B
```

그 밖의 핵심 기술 의사결정

학습 reference 고정

초기에는 빈 상태에서 reference와 baseline을 자동 갱신했지만, 시간이 지나며 baseline이 부풀어 점유 판정 기준이 오염되는 문제가 있었습니다. NCC가 온도 변화를 흡수하므로 자동갱신을 제거하고 학습값을 고정했습니다.

시간(ms) 기반 상태 머신

프레임 수 기준 검증은 오프라인 1fps와 실시간 5fps에서 실제 대기 시간이 달라집니다. 이동평균, 안정 확정, 프레임 갭 판정을 모두 ms 기준으로 바꿔 FPS와 무관하게 동일한 판단이 가능하도록 했습니다.

실내/실외 환경 분류

실외는 EMPTY/PARKED의 NCC 분리가 명확하지만 실내는 분리 폭이 좁았습니다. 학습 시 환경을 선택하고 환경별 임계값을 자동 적용하도록 분리했습니다.

학습 자기 일관성 검사

라이브 학습 중 사람이나 차량이 지나가면 reference가 오염될 수 있어, 학습 프레임들의 상호 NCC 평균/최솟값을 검사하고 기준 이하이면 재학습을 유도했습니다.

시스템 구성

구성 요소	구현 내용
실시간 감지 GUI	WebSocket으로 다중 카메라 프레임을 받아 학습된 reference(npz)를 자동 로드하고, 환경별 임계값을 적용합니다. 매 프레임 NCC, 히스테리시스, 안정성 검증을 거쳐 입출차 이벤트를 CSV로 기록하고 카메라별 카드 UI에 표시합니다.
오프라인 분석 GUI	tmap 파일을 불러와 캔버스에서 ROI를 직접 그리고, 빈 구간을 지정해 baseline을 학습합니다. σ 맵 기반 $\text{NCC} = 1 - \text{NCC}/B$ 판정을 그래프와 이벤트 리스트로 시각화해 ROI 불변 판정을 검증합니다.
학습 모델 포맷	정규화 reference, σ 맵, 환경 정보, 메타데이터를 npz로 함께 저장해 워밍업 없이 즉시 로드할 수 있도록 구성했습니다.

핵심 성과

이 프로젝트의 핵심은 단일 알고리즘을 한 번에 정한 것이 아니라, 오판 원인을 데이터로 진단하며 접근을 바꿔간 과정입니다. 절대 온도 기준 → NCC 패턴 매칭 → 상태 머신 → ROI baseline 예측으로 단계적으로 개선하며, 현장 환경 변화와 ROI 위치 변화에 더 강한 판정 구조를 만들었습니다.