

쉬운설명_GhiVideo_기술이야기

GhiVideo, 쉽게 풀어쓴 기술 이야기 (그림판)

이 문서는 수학을 잘 몰라도, 중학생이 읽어도 이해할 수 있게 쓴 설명서입니다. 어려운 공식 대신 **비유와 그림**으로 "이 프로그램이 무슨 일을, 왜, 어떻게 하는지"를 풀어 씁니다. (더 깊은 기술/수식은 같은 폴더의 `기술명세_GhiVideo_종합기술문서.pdf` 참고.)

0. 한 문장으로 말하면

드론이 철길 위를 날며 찍은 영상을 보면서, 화면 속 다리·터널·역·선로 위에 "여기는 어디, 저건 무슨 다리" 같은 정보를 자동으로 딱 붙여서 보여주는 프로그램입니다.

마치 포켓몬GO나 내비게이션의 증강현실(AR) 길안내처럼, 실제 영상 위에 정보가 떠 있는 거예요.

1. 가장 큰 그림 먼저

우리가 가진 재료는 딱 두 가지입니다.

1. **드론이 찍은 영상** — 매 순간 "드론이 어디에 있었고 어느 방향을 보는지"가 같이 기록되어 있어요.
2. **지도 정보** — 다리·터널·역·측점이 지구상 어디에 있는지(위도·경도) 적힌 데이터.

이 둘을 합치면 "저 다리는 화면의 어느 위치에 보일까?" 에 답할 수 있어요. 그게 핵심 기술입니다.

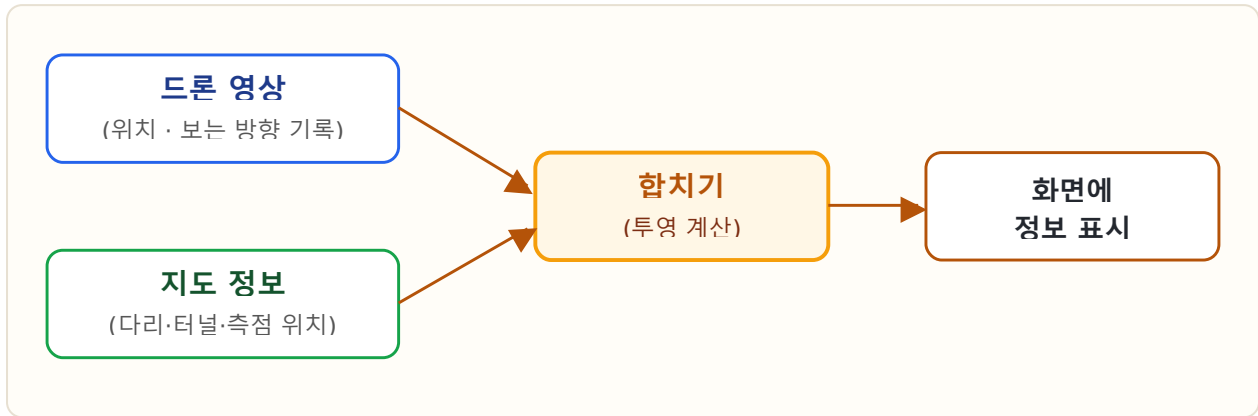


그림 1. 드론 영상 + 지도 정보 → "투영 계산"으로 합쳐 → 화면에 정보 표시

2. 영상 위에 글자를 정확히 붙이는 마법

2-1. 먼저, 위치를 '숫자'로 바꿔요

지구 위 위치는 보통 위도·경도(각도)로 말하는데, 거리 계산이 불편해요. 그래서 "**기준점에서 동쪽 몇 m, 북쪽 몇 m**" 처럼 **미터(m)**로 바꿉니다.

비유: 학교 정문을 (0,0)으로 정하고 "동쪽 50m, 북쪽 30m 지점"이라 말하는 것과 같아요.

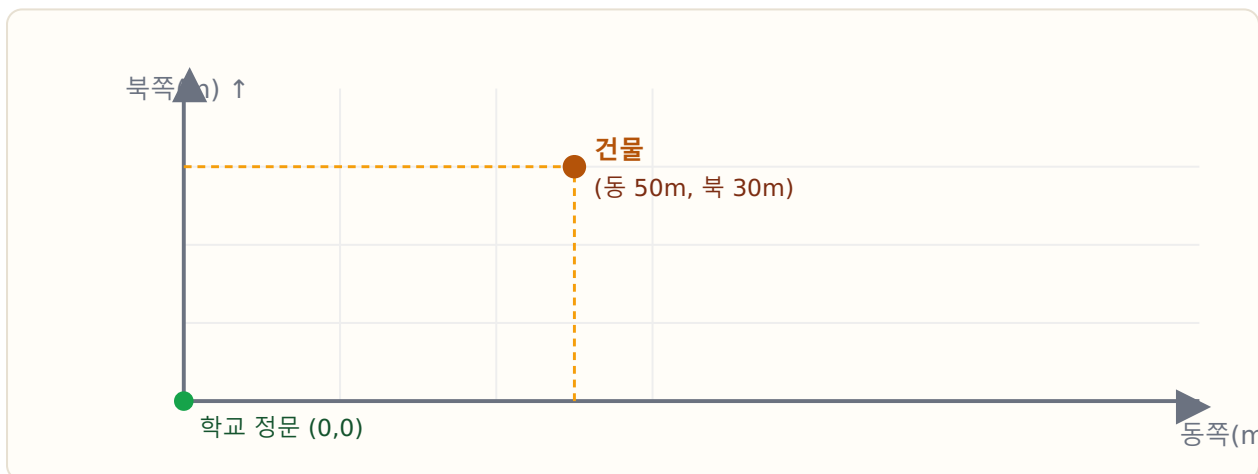


그림 2. 위치를 "기준점에서 동/북 몇 m"로 바꾸면 거리·계산이 쉬워집니다.

2-2. 드론이 '어디서, 어느 쪽을' 보는지 알아야

영상의 각 순간마다 드론은 **위치**(어디 떠 있나)와 **자세**(어느 쪽을 보나)를 기록합니다. 이 둘을 알면, 카메라가 **무엇을** 보는지 계산할 수 있어요.

비유: 친구가 어디 서서 어느 쪽을 가리키는지 알면, 손가락이 무엇을 가리키는지 알 수 있죠.

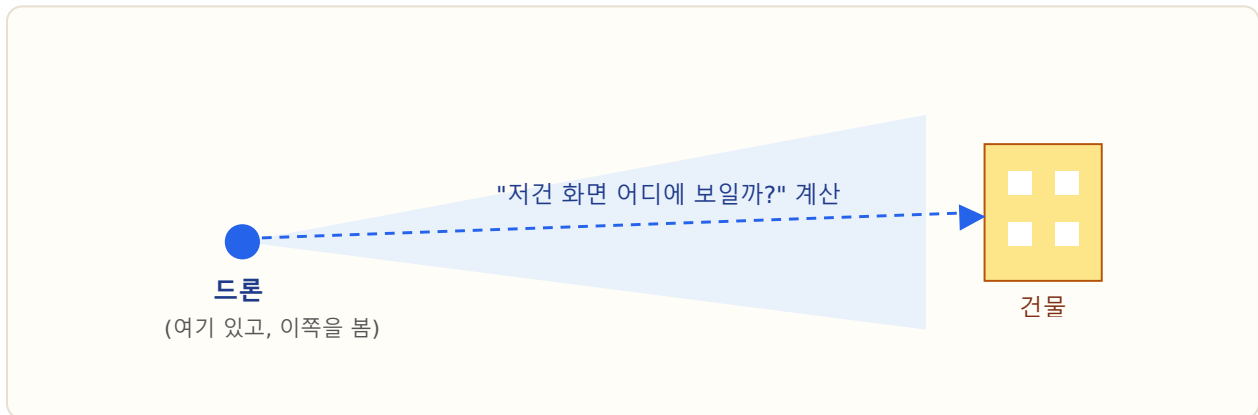


그림 3. 드론의 위치 + 보는 방향을 알면, 대상이 화면 어디에 보일지 계산됩니다.

2-3. 3D 세상을 납작한 화면(2D)에 그려요 — '원근법'

입체(3D)를 화면(2D)에 그릴 때 쓰는 게 **원근법**입니다. 규칙은 누구나 알아요: **가까운 건 크게, 먼 건 작게.**

비유: 바늘구멍 사진기처럼, 카메라는 3D를 2D로 눌러 담습니다.

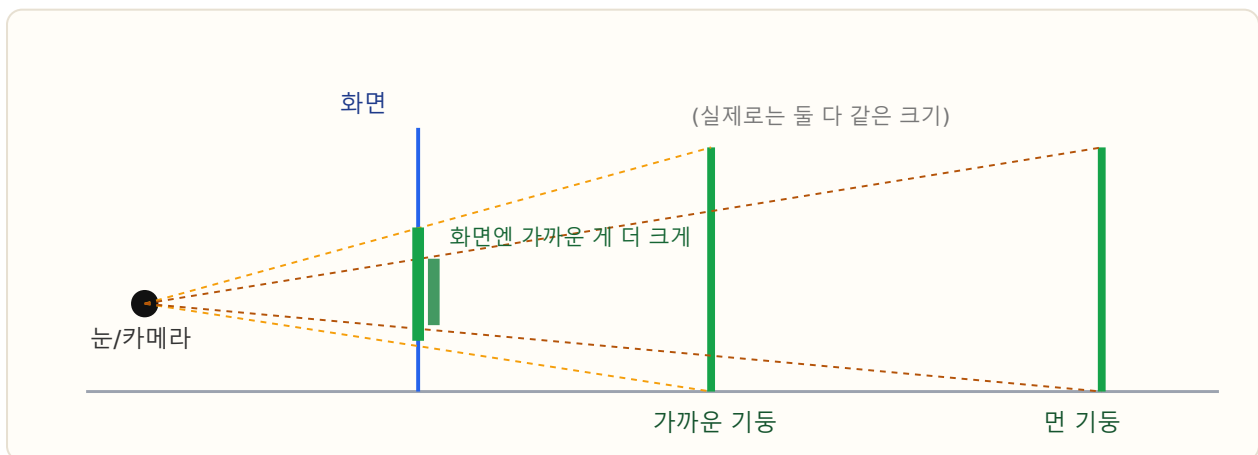


그림 4. 실제 크기가 같아도, 화면에는 가까운 것이 크게·먼 것이 작게 찍힙니다(원근법).

2-4. '높이'에는 함정이 있어요 (지오이드 이야기)

높이를 재는 기준이 **두 가지**라서 헷갈려요.

- **GPS 높이:** 지구를 매끈한 타원으로 본 기준
- **지도 해발고도:** 평균 바닷물 높이 기준(지오이드)

둘은 지역마다 수십 m 차이 나요(대전 약 25.8m). 안 맞추면 글자가 엉뚱한 데로 갑니다.

비유: "1층 바닥 기준"과 "지하 바닥 기준"으로 잰 높이는 같은 건물인데 숫자가 다르죠.

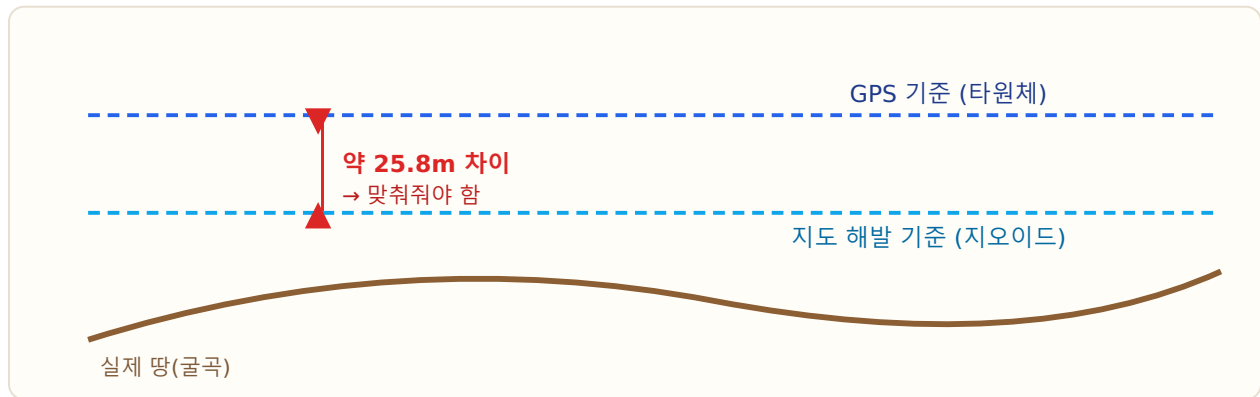


그림 5. 높이 기준이 둘이라 수십 m 차이. 이걸 맞춰줘야 글자가 제자리에 붙습니다.

또 영리한 점: 우리 데이터엔 **측점마다 실제 측량한 높이**가 있어서, 비싼 3D 지형 데이터를 안 사고 **가장 가까운 측점의 높이**를 가져다 쓰면 선로 굴곡까지 자연스럽게 반영됩니다. (특히 후보 — 6장)

2-5. '화각' 맞추기 — 줌렌즈 vs 광각렌즈

같은 자리에서도 **광각**은 넓게, **줌**은 좁게 보이죠. 이 "얼마나 넓게 보나"가 **화각(FOV)** 입니다. 화각이 안 맞으면 글자가 위아래로 어긋나는데, 사용자가 글자 하나를 제자리로 끌어주면 프로그램이 화각을 스스로 맞춥니다.

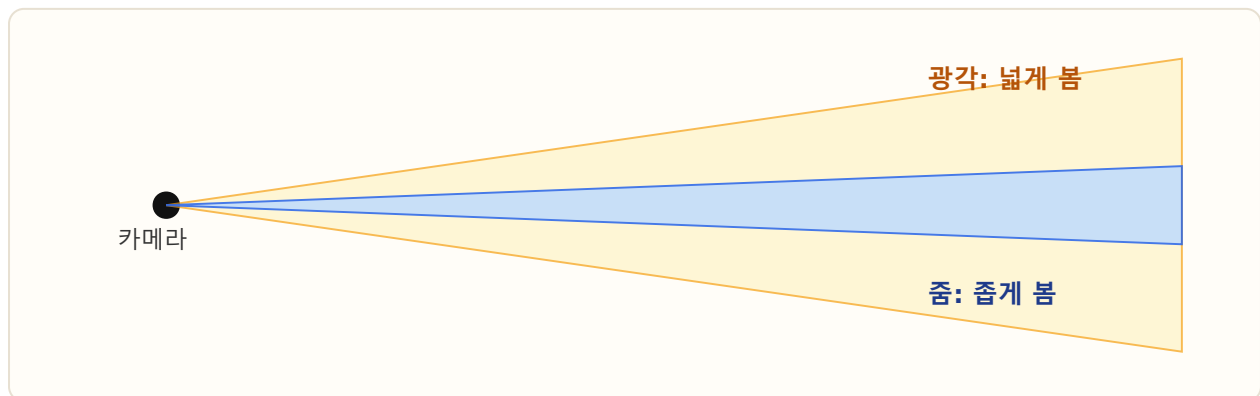


그림 6. 화각(FOV) = 얼마나 넓게 보는지. 광각은 넓게, 줌은 좁게.

2-6. 흔들림 잡기 — '평균 내기'

GPS-자세 기록엔 미세한 **떨림**이 있어 글자가 부르르 떨려요. 여러 순간을 **평균** 내면 부드러워집니다. 단, 방향을 획 트는 순간까지 섞으면 늦게 따라오니까 — **곧게 갈 땐 많이, 틀 땐 적게** 평균 내요.

⇒ **비유**: 떨면서 그은 선을 옆 점들과 평균 내어 매끈하게 다듬는 것.

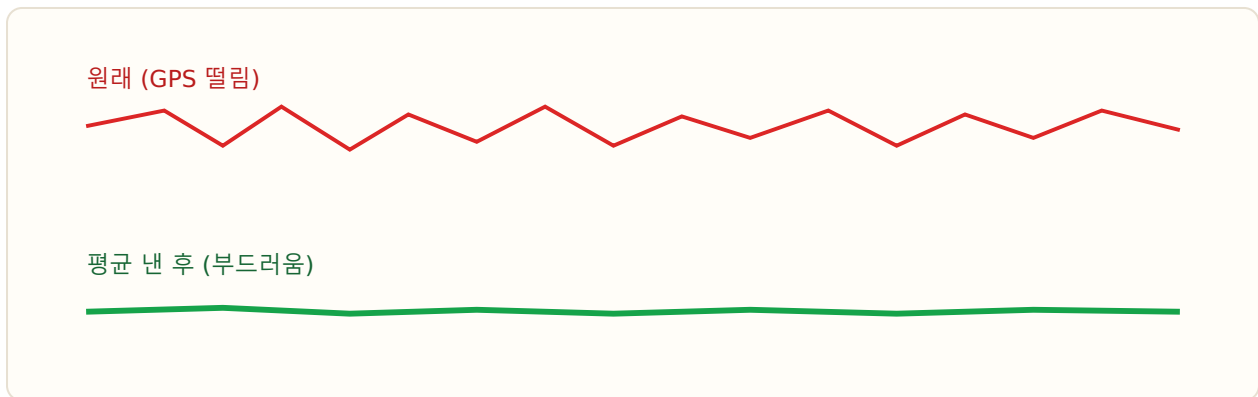


그림 7. 떨리는 값(빨강)을 평균 내면 부드러운 선(초록)이 됩니다.

2-7. 손으로 끌어서 위치 고치기 — '거꾸로 계산'

지도 위치가 가끔 틀려요. 화면에서 글자를 **제자리로 끌면**, 프로그램이 그 화면 위치를 **거꾸로 따라가** 실제 위치를 찾아 고칩니다. (2-3을 반대로 돌리는 것!) 고친 위치는 저장돼 다음에도 적용돼요.

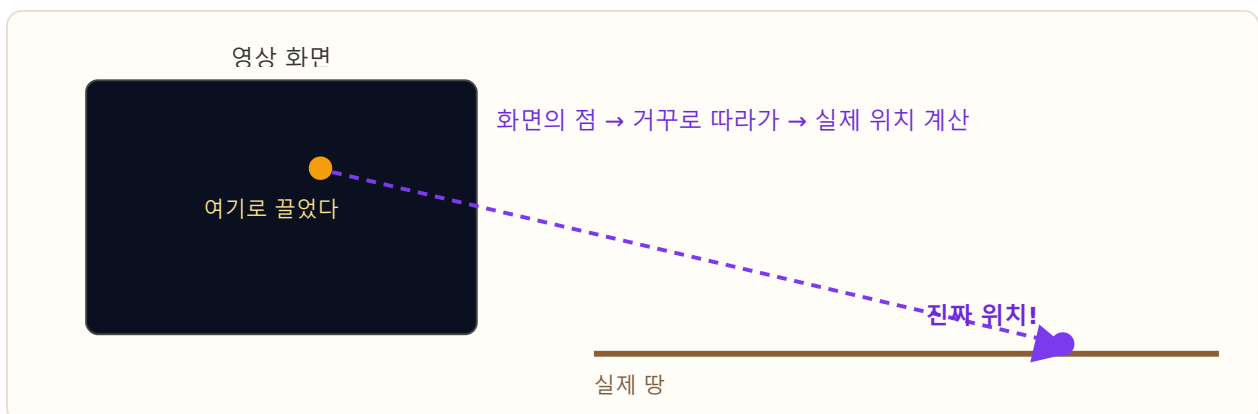


그림 8. 화면에서 끈 점을 거꾸로 따라가 실제 위치를 찾아 자동 보정합니다.

3. 측점(測點) — 우리 영상의 'km 표지판'

3-1. 측점이 뭐예요?

고속도로의 "서울 기점 158km" 표지판처럼, 철도도 **시작점에서 몇 m 떨어졌는지**를 나타내는 표지가 있어요. 이게 **측점(체이니지)** 입니다. 예: 158k200 = 시작점에서 158,200m 지점.

이 프로그램은 영상의 매 순간을 "지금 측점 몇을 보고 있다" 로 바꿔, 시간이 아니라 위치로 영상을 탐색합니다.

3-2. 드론 위치를 측점으로 바꾸기

드론은 철길 바로 위가 아니라 **비스듬히 옆/위**에서 날아요. 그래서 드론 위치를 **철길 선 위로 수직으로 내려찍어** 측점을 읽습니다.



그림 9. 드론 위치를 선로로 수직으로 내려 "지금 보는 측점"을 10m 단위로 읽습니다.

3-3. 왜 같은 측점이 여러 군데 보일까?

주차장·종점에선 드론이 **같은 곳을 앞뒤로 여러 번** 지나가요. 그러면 같은 측점이 여러 번 찍힙니다.

비유: 운동장 트랙을 두 바퀴 돌면 "출발선"을 두 번 지나죠. 잘못된 게 아니라 정말 두 번 지난 거예요.

3-4. 왜 '시간 막대'가 아니라 '거리 막대'일까?

드론은 **가만히 떠 있기(호버)**도 하고 **앞뒤로 왔다 갔다** 해요. 시간 막대면 드론이 멈춰도 커서가 계속 흘러가 어긋납니다. 그래서 **실제 움직인 거리** 기준으로 막대를 만들었어요 — **멈추면 커서도 멈추고, 움직인 만큼만 이동.**

비유: 등산 앱이 "몇 시간째"가 아니라 "몇 km 걸었는지"를 보여주는 것. 쉬면 점이 안 움직이죠.

시간 막대 ✕



드론이 멈춰(호버) 있어도 커서가 계속 흘러감 → 위치와 어긋남

거리 막대 ✓



호버(정지) 중엔 커서도 멈춤 → "지금 보는 위치"와 일치

그림 10. 시간 막대는 멈춰도 커서가 흐르지만, 거리 막대는 움직인 만큼만 이동합니다.

진행 방향(앞으로/뒤로)은 막대의 색(주황=정방향, 청록=역방향)으로 구분합니다.

3-5. 측점을 검색하면 빙글빙글 (순환 이동)

측점값을 입력하고 **Enter**를 치면 그 지점으로 이동, 또 치면 **다음 지점**, 마지막이면 **처음으로** 돌아와요.

비유: 워드의 "다음 찾기"를 반복하면 같은 단어를 차례로 돌다가 처음으로 돌아오죠.

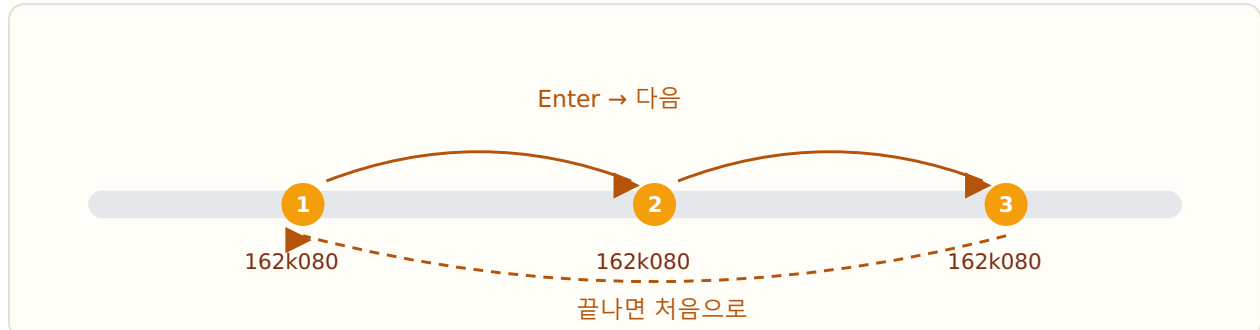


그림 11. 같은 측점이 여러 곳이면, Enter를 누를 때마다 다음으로·끝나면 처음으로 순환합니다.

3-6. '진짜 측점'과 '근처라서 찍힌 것' 구분 (kmExists)

어떤 역 **근처**를 지나가면 동그라미가 찍히는데, 그 자리 실제 측점값이 역의 값과 **다를 수도** 있어요. 그러면 **동그라미는 남기고, 측점값 숫자만 숨깁니다.**

비유: "이 근처에 우체국 있음"은 표시하되, 정확한 주소가 아니면 주소 숫자는 안 붙이는 것.

4. 아주 큰 영상을 끊김 없이 트는 법

2GB 넘는 영상을 통째로 읽으면 컴퓨터가 멈춰요. 그래서 보는 부분만 조금씩(약 10MB) 잘라 받고, 또는 몇 초짜리 조각들로 나눠 이어 붙여 재생합니다(유튜브 방식).



그림 12. 큰 영상을 통째로가 아니라, 보는 만큼 조각으로 잘라 받아 끊김을 막습니다.

오래 보면 데이터가 쌓여 메모리가 터질 수 있어, "지나간 30초치만 남기고 버리도록" 해 뒀습니다. 프레임(사진 한 장) 단위로 정확히 뽑을 땐 FFmpeg라는 전문 도구가 그 장면을 잘라 줍니다.

5. "분명히 고쳤는데 화면이 그대로예요?" (꼭 알아둘 점)

우리가 보는 화면은 "차린 음식(빌드 결과물)" 을 보여줍니다. "날재료(코드)" 를 고쳐도 다시 요리(빌드) 하지 않으면 식탁은 그대로예요.

비유: 레시피를 고쳐도 다시 요리해 차리기 전엔 식탁 음식은 안 바뀝니다.

그래서 코드를 고치면 ① 다시 빌드 → ② 브라우저 새로고침(Ctrl+Shift+R) → ③ 폴더 다시 선택 해야 바뀐 내용이 보입니다.

6. 이게 왜 '특별한(특허감)' 기술일까?

1. 측량 측정 높이로, 비싼 3D 지형 없이 정합 (2-4) + 두 높이 기준 자동 맞춤.
2. 한 번 끌어서 가로·세로·거리 동시 보정 (2-7) — 보통은 깊이 측정 장비가 필요한 일을 장비 없이.
3. 상황 보가며 조절하는 흔들림 잡기 (2-6) — 부드러움과 빠른 반응을 동시에.
4. 시간이 아니라 '위치(측점)'로 영상을 다루기 (3장) — 거리 막대·검색 순환·진짜/근처 구분 등.

△ 단, 특허는 이미 비슷한 게 있는지(선행기술) 전문 검색을 꼭 해봐야 확정됩니다. (아직 미완)

7. 쉬운 용어 사전

용어	쉬운 뜻
측점(체이니지)	노선 시작점에서 몇 m 떨어졌는지 나타내는 '철길 km 표지판'
위도·경도	지구 위 위치를 나타내는 두 숫자
yaw / pitch / roll	카메라가 좌우로 돈 / 아래로 숙인 / 옆으로 기운 정도
투영(projection)	입체(3D)를 납작한 화면(2D)에 그리는 것 (원근법)
역투영	그 반대 — 화면의 한 점이 실제 어디인지 거꾸로 찾는 것
화각(FOV)	카메라가 얼마나 넓게 보는지 (광각=넓게, 줌=좁게)
지오이드	'해발고도'의 기준(평균 바닷물 높이). GPS 높이와 기준이 달라 변환 필요
DEM	땅의 높낮이를 담은 3D 지형 데이터(보통 비쌈)
평활(스무딩)	떨리는 값을 주변과 평균 내어 부드럽게 만드는 것
호버	드론이 한자리에 떠서 멈춰 있는 상태
스테이션바	화면 아래의 측점(위치) 기반 진행 막대
빌드	코드를 컴퓨터가 실행할 형태로 '요리'하는 과정
스트리밍	영상을 통째로 안 받고 보는 만큼 조금씩 받아 트는 것

8. 마치며

"드론이 어디서 어느 쪽을 보는지" 와 "지도에 무엇이 어디 있는지" 를 합쳐서, 영상 위에 정보를 정확히 붙이고, 시간이 아니라 '측점(위치)'으로 영상을 탐색하게 해주는 도구.

복잡해 보여도, 사실은 일상의 생각들 — 친구가 가리키는 곳 알아채기, 원근법, km 표지판, 등산 앱의 걸은 거리, 떨리는 선 다듬기 — 을 컴퓨터로 정밀하게 구현한 것뿐입니다.