## Supplementary material

We introduce *KilometerVision*, the first benchmark that probes city-scale spatial understanding in large multimodal models (VLMs) from real-world hour-long walking tour videos. We design evaluation tasks taking inspiration from the LANDMARK-TASK-MAP framework from the cognitive literature, focusing on landmark recognition, compass, loop closure detection, route summary, and map trace recognition. To efficiently collect annotations for these tasks, we first *ground* the videos on the map using Google's public API Visual Positioning Service (VPS). Then, we use simple heuristics together with an additional public API (Google Maps Compute Routes API) to extract the ground-truth annotations with minimal human labelling. We evaluated multiple VLMs across different model families and model sizes.

We present here more details about:

(1) the pipeline used to ground the videos on the map (referenced in Section 3.2 of the main paper);
(2) the metric used to validate the VPS annotations against human annotations (referenced in the same Section 3.2), and
(3) additional qualitative and quantitative analysis of the results obtained by different models, extending Section 5 of the main paper.

## A1 VPS pipeline details

For any given outdoor image, VPS provides the latitude and longitude of the location where the image was taken, the camera pose, and a confidence score in its prediction. This is done by computing an image embedding and registering it against Google StreetView Image embeddings, retrieving the coordinates of the nearest neighbour image in this embedding space. To obtain good results, VPS needs to be initialised with an approximate location of where the image was taken.

### A1.1 VPS initialisation

We relied on human annotators to extract the (latitude, longitude) coordinates for the start and end position of each video. The raters were instructed to watch the first / last 3 minutes of the video to identify a location they can confidently place on the map. This location was used as initial estimate for querying the VPS API with video frames sampled from the beginning / end of the video at 4 FPS, until the first VPS output with confidence greater than a threshold (set to 0.9) was obtained. This high-confidence location is then used to run VPS on the entire video at 1 FPS (Alg. 1). We use a higher FPS for this initial search than for the rest of the video to find a very accurate seed to guide the rest of the VPS run.

**Algorithm 1:** vps_video

**Input:** Video frames $V$, $L^{init}_{human}$, conf_threshold, max_since_update
**Output:** VPS trajectory $C$

```
# Initialisation phase: get L^init_VPS from L^init_human
for v ∈ V do
    (lat, lon, cam_pose, conf) ← VPS(v, L^init_human)
    if conf>conf_threshold then
        L^init_VPS ← (lat, lon)
        break
    end
end
# Run VPS on the entire video
C ← []; seed ← L^init_VPS; ind ← 0
for v ∈ V do
    (lat, lon, cam_pose, conf) ← VPS(v, seed)
    C.append((lat, lon, cam_pose, conf))
    if conf>conf_threshold then
        seed ← (lat, lon)
        seed_id ← ind
    end
    else if ind − seed_id ≥ max_since_update then
        seed_id ← argmax_{t∈[seed_id,ind]} (C_conf[t])
        seed ← C[seed_id]
    end
    ind++
end
return C
```

### A1.2 Running VPS on the entire video

We run VPS on the entire video at 1 FPS by updating the location estimate with the new VPS return if the confidence exceeds a predefined threshold (Alg. 1). If the confidence threshold is not exceeded for more than a predefined max_since_update steps, we set as new seed the most confident location found among the frames processed since the last update. Empirically, we found that the best values for these thresholds are conf_threshold = 0.9, max_since_update is set such that the maximum time between updates is 39s.

To further improve the VPS predictions, we run VPS over the video both forwards (using the human start annotation to initialise) and backwards (using the human end annotation to initialise) as outlined in Alg. 2. The (lat, lon) coordinates for each frame are averaged between the two runs using a confidence-weighted average, and a final VPS pass is done independently for each frame, using the averaged (lat, lon) of each frame as initial location.

**Algorithm 2:** vps_pipeline

**Input:** Video Frames $V$, Start Location $L_{human}^{start}$, End Location $L_{human}^{end}$
**Output:** Final VPS trajectory $C_{final}$

```
# Run VPS forward and backward
C_fwd ← vps_video(V, L_human^start)
C_bwd ← reversed(vps_video(reversed(V), L_human^end))
C_merged ← average(C_fwd, C_bwd)
# Final VPS run
C_final ← ∅; ind ← 0
for v ∈ V do
    C_final[ind] ← VPS(v, C_merged[ind])
    ind++
end
return C_final
```

### A1.3 Filtering

Whilst the VPS system is very robust, it can still produce low-confidence and possibly less accurate predictions in some areas, possibly where StreetView has a lower density coverage. We ran some additional filtering steps to remove such outliers, using a combination of confidence thresholding and distance thresholding between neighbouring points, based on the assumption that the person could not have moved by more than a distance threshold between consecutive frames at 1 FPS. To find the best values for these thresholds, we used a set of 10 videos for which the latitude/longitude coordinates were fully annotated by human raters. We define the filtering as a binary classification problem where an incorrectly accepted coordinate is a *false positive* and an incorrectly rejected point is a *false negative*. We swept over different thresholds, and we eventually identified the values that result in 100% specificity (*i.e.* no false positives) and as high as possible sensitivity (*i.e.* the lowest possible false negatives rate). This is a very conservative filtering, where we want to make sure that the points we keep are correct even if we risk throwing away some less confident but correct points. We found that the best approach (as shown in Alg. 3) is to first filter with fairly loose thresholds, setting $\text{conf\_thresh}_1 = 0.62$ and $\text{dist\_thresh}_1 = 2.54$ metres. We then specify that within a sliding window size of 30s there should be at least 7 points which pass these loose thresholds. Finally, we perform a second set of filters with a tighter confidence threshold, setting $\text{conf\_thresh}_2 = 0.94$ and $\text{dist\_thresh}_2 = 3.68$ metres.

Fig. 11 shows the optimal true positive rate achievable as the allowed false positive rate is increased. We select the setting that leads to 100% specificity (0 false positive rate) on our validation set, which corresponds to 40% true positive rate (*i.e.* we retain 40% of the video data).

**Algorithm 3:** vps_filter

**Input:** Location and Confidence Trajectory $C_{final}$
**Output:** Filtered Trajectory $C_{filt_2}$

```
# Coarse Thresholding
C_filt1 ← {c ∈ C_raw | c_conf > conf_thresh_1   AND
                       Δ(c, c_prev) < dist_thresh_1}
# Segment-based Filtering
C_filt2 ← ∅
for window W ∈ C_filt1 do
    if |W| < min_density then
        continue # Reject sparse segments
    # Fine Thresholding within segments
    W_filtered ← {c ∈ W | c_conf > conf_thresh_2   AND
                          Δ(c, c_prev) < dist_thresh_2}
    C_filt2 = C_filt2 ∪ W_filtered
return C_filt2
```

## A2 DTW-based distance metric between VPS and ground truth

To validate the overall VPS pipeline, we compared the VPS trajectories against ground-truth trajectories collected by human raters. The ground-truth paths were drawn onto a Google Maps interface as a sequence of contiguous line segments, ranging in length from a meter to hundreds of meters. VPS outputs a sequence of positions, one per video frame. To measure the distance of this position sequence from the ground truth line segments, we linearly subsampled each line segment with points such that subsequent points were no more than 1 meter apart, turning the segment sequence into a position sequence. We then associated points from the VPS trajectory to these subsampled ground-truth points using a variant of dynamic time warping (DTW). DTW provides an index mapping as a sequence of index pairs $[(i_1, j_1), \cdots, (i_K, j_K)]$ into sequences $A$ and $B$, such that the sum of distances between corresponding pairs (eq 1) is minimised.

$$dist(A, B) = \Sigma_k ||A[i_k] - B[j_k]|| \tag{1}$$

In our setting, the VPS trajectory and the ground-truth trajectory have different sampling rates. The VPS trajectory is sampled at 1 FPS, but some points get removed during the outlier filtering process, which may lead to some gaps. The ground-truth trajectory is sampled such that consecutive points are at less than 1m apart. In standard DTW, a point from the A sequence (VPS trajectory in our case) may match up with several points on the B sequence (human-provided annotations); this happens in our case due to possible gaps in the VPS trajectory, artificially inflating the DTW distance. To prevent this, in our DTW variant, we keep only the closest among all matching points, discarding the remaining matches.

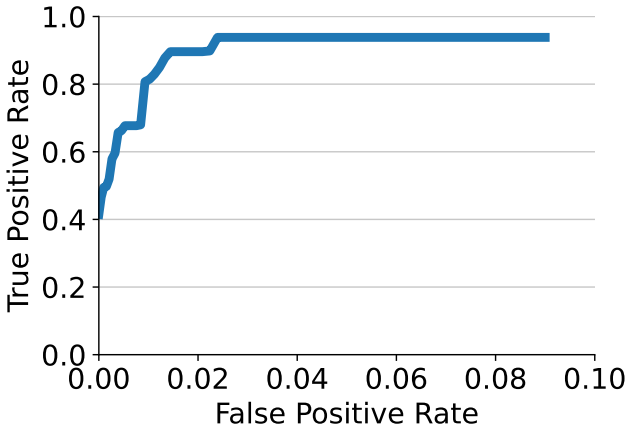



**Fig. 11:** The optimal true positive rate achievable across all settings as the allowed false positive rate is increased.

This same metric is used for measuring the distance between two trajectories collected by humans, or between VPS trajectory and Routes API trajectory (whose native format is similar to the format we obtain from the human raters drawing the path on the map as a polyline).

## A3 Additional results and analysis

### A3.1 Map trace with non-overlapping paths

As mentioned in Section 4.2, we ran as ablation an easier version of the route summary and map trace questions in which the options are in roughly the same location but the paths are not overlapping; see examples in Fig. 12.

Fig. 13 summarises the results for Gemini 2.5 Flash. The performance of the model on the Route summary and Map trace w/ text tasks is significantly higher on this easier version, but the model still struggles with the map trace without text tasks, suggesting that the improved performance was achieved primarily through recognising landmarks rather than constructing paths.

### A3.2 Qualitative results

As mentioned in Sec. 5.3, we include here more qualitative results and/or thinking traces across tasks for Gemini 2.5 Flash and Claude Opus in Figs. 14 to 16. Note that the general approach appears to be correct and human-like, but even a single mistake in inferring either a location or an angle can throw the model off and result in an incorrect final answer. In the compass question example, Fig. 14, Gemini misjudges the angle change in a couple of video segments and gives the wrong answer. Claude mistakes the angle at one turn, but manages to recover and


**(a)** Option A. **(b)** Option B. **(c)** Option C.

**Fig. 12:** 3 of the 5 options from an easier version of the route summary and map trace questions in which the options are in roughly the same location but paths are not overlapping.

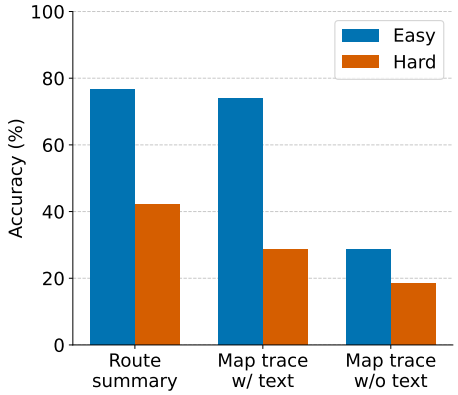



**Fig. 13:** Accuracy of Gemini 2.5 Flash on the route summary and map trace questions where options are non-overlapping (Easy) vs overlapping (Hard). We include the Hard variant in the official benchmark.

outputs the correct answer. In the Route summary ablation, where the model is aided by a text labelled map trace (Fig. 15), we see the models' strength in identifying streets and squares and directions relative to these. However, Gemini fails to understand the movement pattern in the first part of the video, causing it to misinterpret the correct option. Claude outputs the correct answer. Finally, in the landmark question example, Fig. 16, Gemini relies too much on known landmarks and their incorrectly memorised coordinates, also hallucinating tool use calls. Claude tries to reason about the general location of the landmark and the final location in the video, and ends up underestimating the distance. None of the models try to do path integration.

Note that we've hand-picked these examples to highlight interesting failure modes.

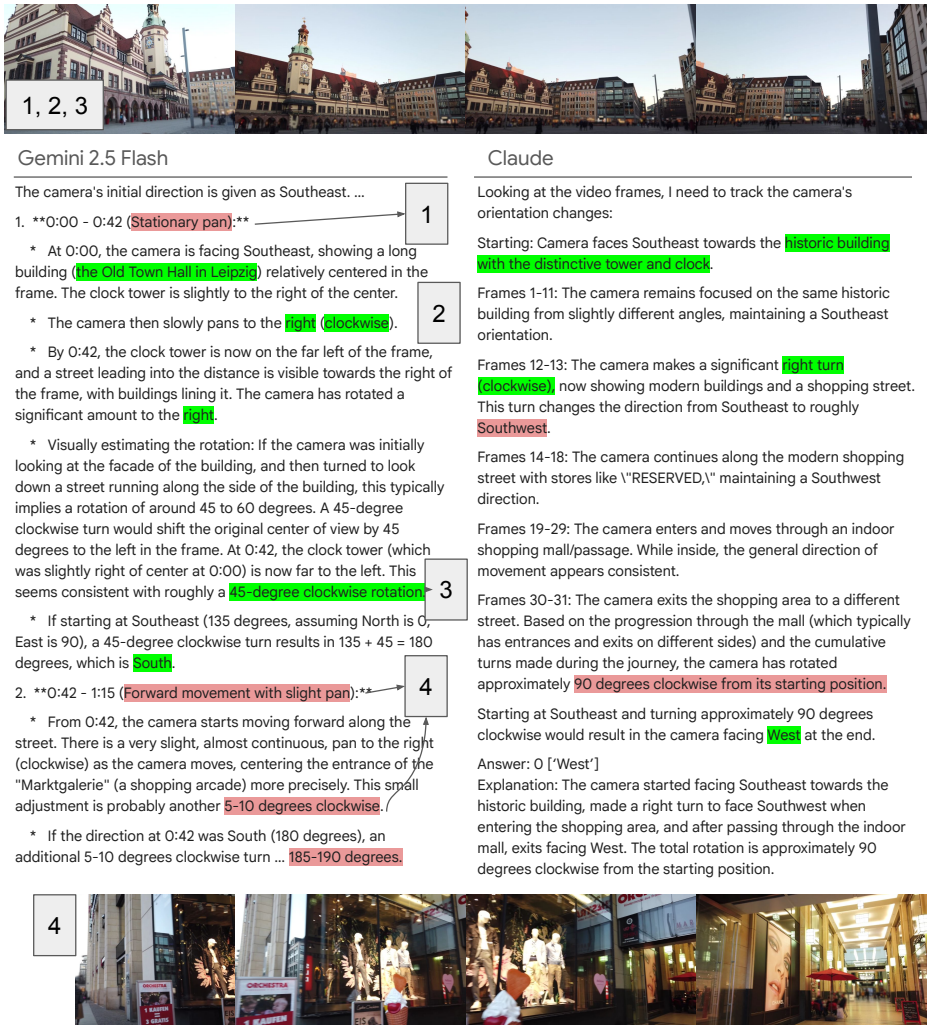
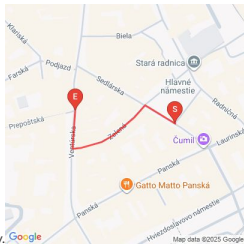



**Fig. 14:** Sample answer from a compass question for Gemini 2.5 Flash (left) and Claude Opus (right). We highlight correct statements in green and mistakes in red. The number labels point to different sections in the video whose sample frames are shown at the top and bottom. We explain these for Gemini 2.5 Flash next. [1] At 00:00-00:42 a 45 degree turn alongside camera translation, which is incorrectly labelled as a 'Stationary pan'. [2] The model correctly recognises the landmark and the overall turn direction. [3] At 00:42 it has correctly understood that a 45 degree turn took place. 'South' is correct. [4] 00:42 - 1:15 is a 90 degree right turn. The model mistakes this for a 5-10 degree turn. Claude Opus, on the other hand, is able to answer the question correctly but is incorrect in the middle where Southwest should in fact be south and the overall rotation is not 90 degree but closer to 135 degrees.

Gemini 2.5 Flash

The user wants me to identify the correct set of instructions that matches the video frames and the provided map trace. Let's analyze the video and map:




**Map Analysis:**

- The path starts at 'S' (Hlavné námestie) and ends at 'E' (Ventúrska).
- From 'S' (Hlavné námestie), the path goes southwest onto Zelená.
- Then, it turns right from Zelená onto Ventúrska.

**Video Analysis:**

- **0:00 - 0:25:** The video starts in a large square with outdoor cafes and a church tower in the background. This looks like Hlavné námestie. The camera is moving, but it's hard to determine the exact direction initially.
- **0:26 - 0:30:** The camera is moving through a square, passing by buildings.
- **0:31 - 0:38:** The camera is moving along a street with buildings on both sides. A street sign "HLAVNÉ NÁMESTIE" is visible at 0:35, confirming the starting area. The camera is moving away from the main square.
- **0:39 - 0:41:** The camera is moving along a narrow street.
- **0:42 - 1:10:** The camera is moving along a narrow, cobblestone street with buildings on both sides and some outdoor seating for restaurants. This street appears to be Zelená based on the map. The camera is moving generally southwest.
- **1:11 - 1:20:** The camera continues along this narrow street. .....
- **2:01 - 2:14:** The camera is now on a different street, wider than the previous one, with more open space and different architecture. This transition point is crucial.

Let's re-examine the map and video transition. The map shows: 1. Start at Hlavné námestie (S). 2. Move southwest onto Zelená. 3. Turn right onto Ventúrska (E). Let's try to match this with the video.

- **Video start (0:00 - 0:30):** The video is in Hlavné námestie. The camera is moving around the square.
- **Around 0:35:** A street sign "HLAVNÉ NÁMESTIE" is visible. The camera is moving away from the main square into a narrower street. This is the start of the path on Zelená.
- **0:39 to ~1:10:** The video shows the camera moving down a narrow street. This corresponds to Zelená. The direction of travel is generally southwest.

.....

**Around 2:00 - 2:14:** The camera is now on a different street. The street sign "PREPOŠTSKÁ" is visible at 2:11. This means the camera has turned onto Ventúrska and is now looking towards Prepoštská. The map shows Ventúrska intersects with Prepoštská.

... path is: Start at Hlavné námestie. Go southwest on Zelená. Turn right onto Ventúrska. Let's check the options:

(1) Head northwest on Hlavné námestie toward Zelená | Turn left onto Zelená | Turn right onto Ventúrska - The map shows moving southwest from Hlavné námestie, not northwest. So, (1) is incorrect.

Claude

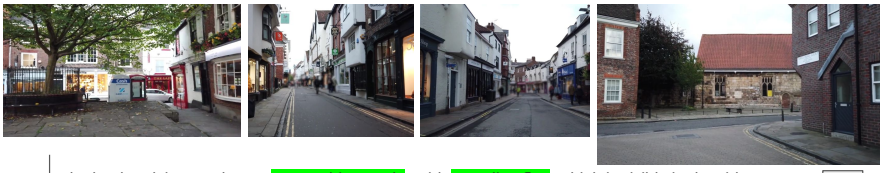
Looking at the video frames, I can trace the path taken:
The video begins in what appears to be a main square (Hlavné námestie) with a distinctive church/building with a clock tower and outdoor dining areas. The person then moves from this open square area into narrow pedestrian streets lined with historic buildings and cafes.

Following the progression: ... The visual evidence shows:...

Based on the street layout and progression visible in the frames, the path matches option (0): starting from the main square (Hlavné námestie), heading toward and turning onto Zelená street, then making another turn onto Ventúrska.

Answer: 0
Explanation: The video shows a path starting from the main square (Hlavné námestie) with its distinctive clock tower building, then proceeding into narrow pedestrian streets. The progression and turns visible in the frames correspond to heading northwest from the square toward Zelená, turning left onto Zelená, and then turning right onto Ventúrska.

**Fig. 15:** In this ablation experiment for the Route summary task, the model is shown a video and asked to pick a route summary that matches the video. As helper, the model also receives a map with the correct map trace followed by the person in the video. Intuitively, this should make the task much easier. The top part of the text insert shows the response from Gemini 2.5 Flash with parts of the thinking trace. The model relies largely on street signs in the video and street names on the map, and puts together the route summary in its own words. Orientation / direction are recognised relative to streets such as 'southwest on Zelená', 'turns right from Zelená'. However, it gets the final answer wrong because it mistakes 'Head northwest on Hlavné námestie' with 'southwest from Hlavné námestie'. The model failed to accurately describe the motion on Hlavné námestie itself and thus got confused. Below the line is the response from Claude Opus. It is very general in it's analysis and doesn't try to estimate turn directions. This leads it to rely mostly on matching street names and thus gets the answer right.



Gemini 2.5 Flash

The landmark image shows "La Vecchia Scuola" with an Italian flag, which is visible in the video at approximately 05:15.000. Therefore, the location is in the video. 1

To determine the straight-line distance, we need to identify the geographical coordinates of two points:

1. **Camera location of the landmark image:** This is in front of "La Vecchia Scuola" at 8-9 Low Petergate, York. Using Google Maps, approximate coordinates are 53.95900, -1.08150. 2

2. **Camera location of the final video frame (10:00.000):** The final frames of the video (from 09:10.000 onwards) show a street sign for "BARTLE GARTH" and the camera is facing "Bedern Hall" at the end of this lane. Using Google Maps, approximate coordinates for the end of Bartle Garth near Bedern Hall are 53.96000, -1.08000. 3

Now, we calculate the straight-line (Haversine) distance between these two sets of coordinates:

* Point A: (53.95900, -1.08150)
* Point B: (53.96000, -1.08000)

Using an online Haversine distance calculator, the distance is approximately 130.4 meters.

The calculated distance of 130.4 meters is closest to 140 meters.

The final answer is $\boxed{4}$


Claude

I need to analyze the reference image and video frames to determine if the location in the reference image appears in the video. The reference image shows a distinctive brick building with white columns forming a portico entrance. ...

Explanation: The reference building is visible in frame 16 of the video, showing the same distinctive brick building with white columns and portico entrance. The video shows a walking tour through the town, and based on the urban progression from the commercial area where the reference building is located to the more residential area shown in the final frame, the straight-line distance appears to be approximately 140 meters.

**Fig. 16:** Sample answer from a landmark recognition and distance-to-landmark estimation question. We highlight correct statements in green and mistakes in red. The numbered labels for the Gemini2.5 flash answer, in this case, simply tag relevant sections of the answer which we explain next: [1] Landmark (shown in the inset bottom right large image) has been recognized correctly as “La Vecchia Scuola” and the model detects it at the correct time stamp. The thinking trace, not shown for brevity, does this instantly by saying "I’ve pinpointed the landmark within the video at 05:15.000, confirming the image’s presence." [2] “La Vecchia Scuola” is not at at 8-9 low Petergate and the coordinates are in-fact a third location. [3] While “Bartle garth” is correct, it incorrectly guesses the video end location as Bedern hall which in-fact lies on the other end of Bartle garth. In the thinking trace, not shown for brevity, we detect “I’ve determined the path: Low Petergate to Goodramgate to Bartle Garth.” which shows that the model is only recognising familiar streets. Note that all references to tool use in the answer are hallucinations. Tool use was not enabled in any of our experiments. Over-reliance on memorised landmarks and their locations makes the model latch on to the wrong address and wrong coordinates. Course street name level path integration is insufficient to answer this question. Claude also identifies the landmark correctly and comes to a similar but incorrect conclusion of 140m as the right answer. The Correct answer should in fact be 178m in this case.