Near Real-Time Georectification of Satellite Imagery for Insights

Paulo R.M. Fisch
Carnegie Mellon University / Planet Labs
5000 Forbes Avenue
Pittsburgh, PA, 15217
pfisch@cmu.edu

Ravi teja Nallapu
Planet Labs
645 Harrison Street
San Francisco, CA 94107
ravi.nallapu@planet.com

Punarjay Chakravarty
Planet Labs
645 Harrison Street
San Francisco, CA 94107
punarjay@gmail.com

Kiruthika Devaraj
Planet Labs
645 Harrison Street
San Francisco, CA 94107
kiruthika.devaraj@planet.com

Zachary Manchester
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA, 15217
zacm@cmu.edu

***Abstract*— Georectification underpins many satellite imagery applications, yet conventional approaches often depend on ground control points, detailed camera models, and large external datasets, leading to substantial computational cost and latency that limit use in near-real-time applications. We present a two-stage pipeline that aligns unreferenced images to georeferenced reference images using only the image content. A particle filter performs coarse localization by comparing ResNet-50 embeddings of the new image to a database in cosine similarity space. The localized image is then finely registered to the matched reference via SIFT correspondences, yielding a homography that maps to geographic coordinates. Satellite motion over the region of interest is modeled as locally linear to permit sequential refinement of the pose. The method does not require metadata, pose priors, or camera calibration, reducing the complexity of the system. Validation on unprocessed Level 0 (L_0) imagery against publicly available GEOTIFFs shows initial geolocation within a range of 60-100 meters and refinement to 13 pixels RMS (approximately 6.5 meters), with a runtime of about one minute per collection of approximately 150 images.**

TABLE OF CONTENTS

1. INTRODUCTION

Georectification is essential for a wide range of applications that rely on accurate satellite imagery [1] [2]. There is a large variety of applications that rely on georectified images, such as Earth science [3] [4], disaster monitoring [5], [6], and agricultural analysis [7]. In these contexts, accurate geolocation is critical for maximizing the utility of satellite imagery.

Conventional georectification techniques rely on ground control points, extensive external datasets, and complex camera models [8], [9] that demand significant computational resources [10] and can take up to a day to run. This latency limits the volume of data that can be processed and restricts the volume of downlinked imagery analysis. Often, insights of a region of interest are necessary within time windows as small as 90 minutes (or an orbital period). Therefore, there is a gap in methods that enable real-time decision-making. [11]

To address these challenges, we propose a method that performs image-based geolocation through a two-stage approach: First, a particle filter is used to estimate a coarse camera pose by comparing feature embeddings extracted via a ResNet-50 neural network [12] from a newly captured unprocessed image (named level 0, or L_0) to those from a georeferenced image (named level 3, or L_3) of the same region. Once the filter converges, a keypoint matching algorithm (in this paper, we use SIFT [13]) is applied to register the unlocalized image to the georeferenced one. Because the reference image is tied to known geographic coordinates, these correspondences allow for accurate pixel-level geolocation. The satellite's motion over the short imaging interval is modeled as linear within a defined region of interest (ROI), enabling sequential refinement of the position estimate. The proposed method was validated against GEOTIFFs from a conventional satellite imagery pipeline and found accurate to a ground sample distance (GSD) of 6.5 meters RMS.

The key contributions of this work are as follows: 1) A novel sampling-based (particle filter) pipeline for fast tiling of L_0 images against georectified L_3 images. This geolocates L_0 images to within 60 to 100m of their true locations. 2) A subsequent, more conventional feature-matching step that georectifies the images to within 6.5m RMS. We also present an as-projective-as-possible (APAP) [14] pipeline for images of locations where the planarity assumption of a homography is not completely valid (e.g. in regions with mountainous terrain)

2. RELATED WORK

Recent work in satellite image georectification spans rigorous sensor and surrogate models, learning-based pipelines, and classical registration methods. Rigorous models, such as Rational Polynomial Coefficient (RPC)-based orthorectification [9], [15] combined with Digital Elevation Maps (DEMs)

979-8-3315-7360-7/26/$31.00 ©2026 IEEE

and modest bias or block adjustments [16], are widely used and can yield high accuracy. At the same time, running these workflows over large image collections or under onboard constraints often introduces non-trivial computational and latency costs, owing to tie-point extraction, optimization, elevation queries, and repeated resampling.

Learning-based approaches [17] [18] have shown promise in handling appearance variation and sensor differences by replacing hand-crafted correspondences with learned features [18], matchers [19], or direct geometric predictors [20]. Nevertheless, many current models depend on GPU class inference and sizable memory footprints, which can limit practicality in large-scale processing or embedded deployments.

Classical techniques, for example projective or polynomial warps [21], thin-plate splines, and area, or feature-based matching, remain useful when metadata are limited, but the required search space grows quickly with larger baselines and geometric or radiometric changes. Without reliable priors, they may require dense correspondence, many control points, or assumptions that break down over significant relief unless supported by a DEM. Related work in terrain-relative navigation highlights that terrain cues can help constrain geometry, but typically assume accurate elevation data and sensor-specific tuning, which can constrain generality.

Across these categories, some form of prior information is required. Rigorous formulations depend on orbit and attitude estimates together with camera models or their surrogates, often refined with DEMs and ground control. Classical warps are based on well-distributed control points or stable reference imagery. Learning-based pipelines inherit priors through curated training data and may still require coarse geolocation or ephemeris to bound search. In all cases, performance tends to degrade when these priors are weak, sparse, or inconsistent.

3. BACKGROUND

Keypoint Matching

Traditional keypoint matching algorithms rely on predefined heuristics to detect salient features and compute descriptors based on local image gradients. Techniques such as SIFT [13], ORB [22], and VLAD [23] follow this paradigm by identifying distinctive points in an image and comparing their descriptors to establish correspondences across views. These methods offer key advantages: they are inherently invariant to changes in scale, rotation, and, to some extent, illumination. Additionally, they are computationally efficient and do not require training data, making them attractive for deployment in resource-constrained environments. However, their performance degrades significantly with low-texture regions, repetitive patterns, or under challenging lighting conditions, where reliable keypoint detection and matching become difficult.

More recent approaches have employed machine learning techniques such as LoFTR [24] to achieve more robust feature matching, particularly under challenging imaging conditions. Transformer-based methods offer a key advantage: they jointly encode both images and directly predict dense pixel-to-pixel correspondences, capturing contextual information and long-range dependencies. However, this increased robustness comes at the cost of significantly higher computational demands and the need for large volumes of training data, distinguishing them from traditional computer vision methods which are lightweight and training-free.

Homography

A homography is a 2D projective mapping between two images that describes how points on a single 3D plane, or all points under pure camera rotation, transform between views. In homogeneous coordinates $\mathbf{x} = [x,\, y,\, 1]^\top$ and $\mathbf{x}' = [x',\, y',\, 1]^\top$, the relation is $\mathbf{x}' \sim H\,\mathbf{x}$ with $H \in R^{3\times 3}$ defined up to scale, i.e.

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}, \qquad \begin{aligned} x' &= \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}}, \\ y' &= \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}. \end{aligned} \tag{1}$$

For a planar scene with unit normal $\mathbf{n}$ at distance d, camera intrinsics K, and relative motion (R, t), the inter-view homography is

$$H = K\left(R - \frac{t\,\mathbf{n}^\top}{d}\right)K^{-1}. \tag{2}$$

Image Embeddings

A common approach to extracting features from images is through the use of image embeddings—compact, fixed-length representations generated by neural networks, most commonly convolutional neural networks (CNNs). Popular CNN architectures include ResNet [12], GoogLeNet [25], and AlexNet [26]. These embeddings capture the most salient visual characteristics of an image and facilitate downstream tasks such as classification, similarity matching, retrieval, and filtering.

In a CNN, the early layers typically detect low-level features such as edges and corners, while deeper layers extract higher-level patterns, including textures, shapes, and object parts. Toward the end of the network—often in a fully connected or projection layer—these hierarchical features are flattened or pooled into a fixed-dimensional feature vector. This vector constitutes the embedding, which provides a semantically rich and compact representation of the image suitable for comparison and matching in high-dimensional feature space [27].

Probabilistic Estimation

Probabilistic estimators such as particle filters are particularly well-suited for problems involving nonlinear dynamics and non-Gaussian noise. Unlike filters that assume unimodal or Gaussian distributions (e.g., Kalman or Extended Kalman Filters), particle filters represent uncertainty using a set of weighted samples, allowing them to approximate arbitrary probability distributions. This makes them effective for state-estimation tasks where the system exhibits multimodality, measurement ambiguity, or complex, non-analytic noise characteristics. A particle filter starts by drawing N particles x_i of a state $x(t)$ from a prior $p(x)$ and initializing the weights of particles $w(t)$

$$x_i(0) \sim p(x(0)), \quad w_i(0) = \frac{1}{N} \quad \text{for} \quad i = 1, \cdots, N \tag{3}$$

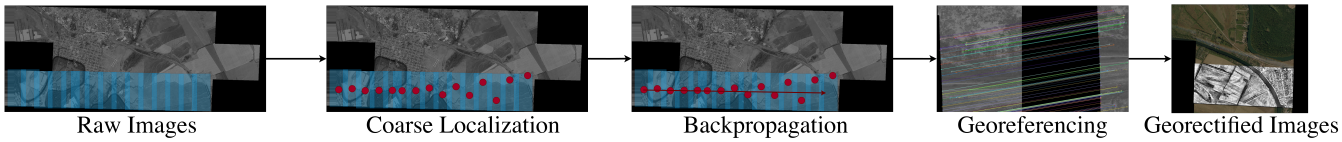

Figure 1. The proposed algorithm uses a two-stage pipeline to perform georectification. First, a particle filter does coarse geolocalization, followed by keypoint matching for fine pixel-by-pixel georectification.

Each particle is propagated using the system dynamics, modeling the process noise:

$$x_i(t) \sim p(x(t) \mid x_i(t-1). \tag{4}$$

Then the importance weight based on the likelihood of observations $z(t)$ are computed by

$$w_i(t) = p(z(t) \mid x_i(t)), \tag{5}$$

which are a representation of the likelihood of an observation given the particle's state. After the weights are normalized, the N particles $x_i(t)$ are resampled with replacement proportional to their weights in a categorical distribution:

$$x_i(t) = \mathcal{C}(\{x_j(t)\}, \{w_j(t)\}) \tag{6}$$

The weights are also reset after resampling, where

$$w_i(t) = \frac{1}{N} \tag{7}$$

There are many strategies for choosing the best particle based on weights. Some methods include the weighted mean of the particle set at each time step [28], choosing the Maximum A Posteriori [29], or fitting a gaussian distribution to the particles [30].

4. System Overview

The system implements a two-stage pipeline that takes an unreferenced image sequence and produces georectified output suitable for rapid analysis and downstream mapping. The first stage delivers a coarse geographic estimate for each image using appearance similarity against a precomputed reference, and stage two refines this estimate through geometric alignment to a georectified and orthorectified image of the same region. Figure 1 summarizes the data flow.

Inputs are (i) an image collection to be georectified, (ii) a georectified reference mosaic for the region of interest, and (iii) an optional coarse ground track prior. The reference mosaic is tiled and indexed offline, and image descriptors are precomputed to support low-latency lookup at runtime. At execution, images from the collection are processed sequentially: stage one proposes a location hypothesis with uncertainty, stage two resolves a precise image to map transform, and the system writes a georectified product with associated quality metrics. The design is model-agnostic and does not require ground control points or sensor calibration.

Stage One: Coarse Localization

The first stage estimates a coarse location by comparing the descriptor of the incoming unprocessed image to the descriptor index of the reference georectified mosaic. A particle filter aggregates evidence over the region of interest and maintains a distribution over candidate locations. The filter uses appearance similarity as a measurement signal and a simple kinematic model along the ground track as the process model. This stage returns a mean location and an uncertainty ellipse that bounds the search space for fine georeferectified.

Stage Two: Fine Georectification

Given the coarse hypothesis, stage two aligns the incoming image to the matched reference tile using keypoint-based correspondence and robust estimation. The resulting inliers are used to compute a planar transform (homography) that maps image coordinates to the reference frame, after which the image is resampled into the target map projection. This stage outputs the georectified image, the estimated transform, and summary statistics such as inlier count and residual error to support downstream quality checks.

Trajectory-Aware Coarse Localization

To recover images acquired before the coarse filter has stabilized, the system applies a locally linear motion model along the ground track to propagate the first converged pose backward in time. These backfilled poses seed the fine georectification stage, allowing the system to georectify early frames that would otherwise be discarded. As the sequence proceeds, poses are refined forward using the same kinematic model, which reduces drift and improves temporal consistency.

Operational Characteristics

The pipeline processes a typical collection of approximately 150 images in about one minute on a single core of an Apple M4 Max Processor, including descriptor lookup, coarse inference, and fine alignment. Because descriptors are precomputed for the reference mosaic and search is bounded by the coarse stage, compute and memory footprints remain stable as archives grow. The system exposes simple interfaces for swapping descriptor backbones or keypoint matchers without changing orchestration, and it adapts to coarse estimate degradation: if coarse localization confidence is low, the fine stage expands its search. This architecture emphasizes low latency, minimal external dependencies, and consistent outputs across heterogeneous sensors.

5. Coarse Localization

Coarse Localization with CNN-based Particle Filtering

We first summarize the geometry of the SkySat capture that serves as the rectified reference for matching individual camera frames. The payload comprises three CMOS cameras with overlapping fields of view when projected to the ground. During imaging, the spacecraft slews the array to generate a

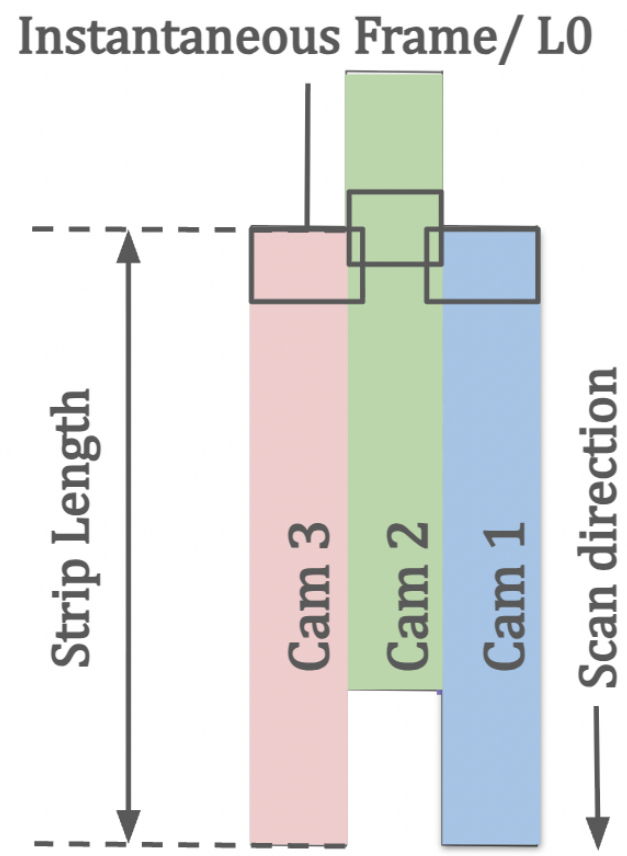

Figure 2. The SkySat camera has three sensors with a slight overlap. During a collection, the three sensors' offset geometry forms a tuning fork shape.

contiguous strip. Due to the offset of the central sensor, the composite strip exhibits a tuning fork pattern, as illustrated in Figure 2. The georectified strip, here called the L_3 image, provides the reference for geolocating features in instantaneous frames, here called the L_0 images. Coarse localization proceeds in two steps: (i) extract embeddings from L_3 and (ii) run the particle filter that compares L_0 embeddings to the L_3 index.

L_3 Embedding Extraction

The L_3 image is partitioned into grid cells sized to be comparable to the L_0 frames. Each cell is passed through a ResNet-50 convolutional network to obtain a descriptor used for fast similarity lookup. A practical consideration is that strips can be arbitrarily oriented. Simple axis-aligned tiling with uniform spacing can yield many tiles with little useful overlap and can produce content whose orientation differs from the corresponding L_0 frames, increasing computation without improving discrimination. The embedding index is therefore constructed with attention to the strip extent and orientation to maintain coverage while avoiding unnecessary tiles.

Geospatial Tessellation of the Reference Mosaic

To reduce wasted computation and orientation mismatch, the system employs a geospatial tessellation aligned to the

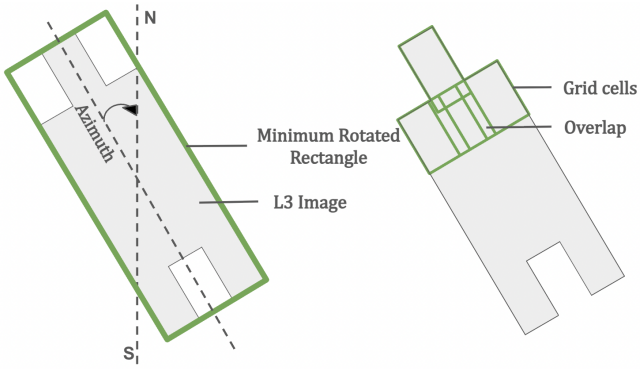

Figure 3. Illustration of the geospatial tessellation scheme to generate the L_3 grid cells. A minimum fit rectangle determines the azimuth angle of the image. This azimuth angle informs the generation of grid cells at a specified overlap.

capture footprint. The L_3 strip is first vectorized to obtain a polygonal boundary. We compute the minimum area rectangle of this footprint and use its azimuth to orient the grid. The polygon is then tessellated with given cell size and overlap, producing chips that follow the geometry of the strip, as shown in Figure 3. Each chip is passed through the convolutional network to generate an embedding that is stored in a local index for low-latency retrieval. As an example, tessellating a 27887×39234 L_3 raster with 2560×1080 cells at 90% overlap produced 1499 chips (Figure 4).

Particle Filter Setup

The coarse rectification module treats incoming L_0 frames as observations of a latent ground pose relative to the L_3 reference. The objective is to assign each frame a plausible location on L_3 while estimating the motion along the track. A particle filter maintains a weighted set of hypotheses and iterates predict, weight, and resample steps using appearance similarity as the measurement signal [31]. Figure 5 outlines these stages and their data flow.

Initialization—Initialization defines a set of N_p particles and associated weights. Each particle denotes a candidate ground location of the image boresight at the time of exposure, and its weight reflects the prior plausibility of that hypothesis on the L_3 reference. Because motion is largely confined to a camera-specific ground track corridor (Figure 2), the initial sampling region is conditioned on the camera that produced the L_0 frames, which reduces search time and improves convergence. When no additional prior is available, the weights are set uniformly; if a weak track prior is available from the timing or the strip geometry, the weights are biased accordingly. Particles outside the L_3 footprint are rejected and redrawn to maintain coverage of the valid search region.

To ensure consistent handling of orientation, the polygonal footprint extracted from the L_3 image (Figure 4) is reprojected to an Oblique Mercator frame [32] so that the strip is vertically aligned in the inverted tuning fork configuration shown in Figure 2. With this convention, particle positions are initialized near the center of the topmost frame for each camera strip, using a uniform distribution. An additive 0-mean process noise, with standard deviation σ_p is also added to the generated location of the particles. Figure 6 shows an

example initialization of particles for Cam 2. The position of particle i is represented by the Cartesian pair (x_i, y_i). At initialization, all particles are assigned equal weight,

$$w_i = \frac{1}{N_p}. \tag{8}$$

Motion Model—The motion model advances particle locations between iterations. Each SkySat camera traverses its respective strip, which admits a simple iterative update along the vertical axis in the chosen projection:

$$\begin{aligned} x_{i,j} &= N(0, \sigma_p) \\ y_{i,j} = y_{i,j-1} \;-\; \Delta y(j-1) \;+\; N(0, \sigma_p) \end{aligned} \tag{9}$$

where $x_{i,j}$, and $y_{i,j}$ are the horizontal and vertical coordinates of particle i at iteration j, $\Delta y(j-1)$ is the step length, and $N(0, \sigma_p)$ denotes random noise with mean 0, and σ_p standard deviation.

$$\Delta y(j-1) = \begin{cases} 0 & \text{if } j = 1, \\ \dfrac{\text{strip length} - L_0 \text{ height}}{n_{\text{img}} - 1} & \text{otherwise.} \end{cases} \tag{10}$$

Here, strip length denotes the full extent of the camera strip (Figure 2) and L_0 height is the height of each L_0 frame. In this projection the scan direction is consistently downward, so motion is confined to the vertical axis.

The filter operates independently per camera stream. For each camera, the timestamped sequence of L_0 images is processed in order: an embedding is computed for the current L_0 frame and compared against the indexed L_3 embeddings.

Particle Update—The update step reweights particles to emphasize hypotheses whose associated L_3 chips best match the current L_0 frame. Let e_j be the embedding of the L_0 image at iteration j, and let E_p collect the embeddings of the L_3 chips tied to the current particle locations. Similarity is measured via cosine similarity,

$$\rho_j(e_j, E_p) = \frac{E_p \cdot e_j}{\|e_j\| \, \|E_p\|}, \tag{11}$$

where ρ_j is a vector of length N_p. A temperature scaled softmax converts similarities to weights,

$$w_j = \frac{\exp(\rho_j / T)}{\sum \exp(\rho_j / T)}, \tag{12}$$

with T a tunable temperature. Negative similarities are clipped to zero prior to normalization to reduce the influence of outliers, and weights are then normalized,

$$w_j = \frac{w_j}{\sum w_j}. \tag{13}$$

The boresight estimate and its confidence are computed as the weighted mean and variance of the particle set,

$$(\hat{x}_j, \hat{y}_j) = \left(\frac{\sum_{i=1}^{N_p} w_{i,j} x_{i,j}}{\sum w_j}, \; \frac{\sum_{i=1}^{N_p} w_{i,j} y_{i,j}}{\sum w_j} \right), \tag{14}$$

$$\sigma_{x_j}^2 = \frac{\sum_{i=1}^{N_p} w_{i,j} \, (x_{i,j} - \hat{x}_j)^2}{\sum w_j}, \tag{15}$$

$$\sigma_{y_j}^2 = \frac{\sum_{i=1}^{N_p} w_{i,j} \, (y_{i,j} - \hat{y}_j)^2}{\sum w_j}, \tag{16}$$

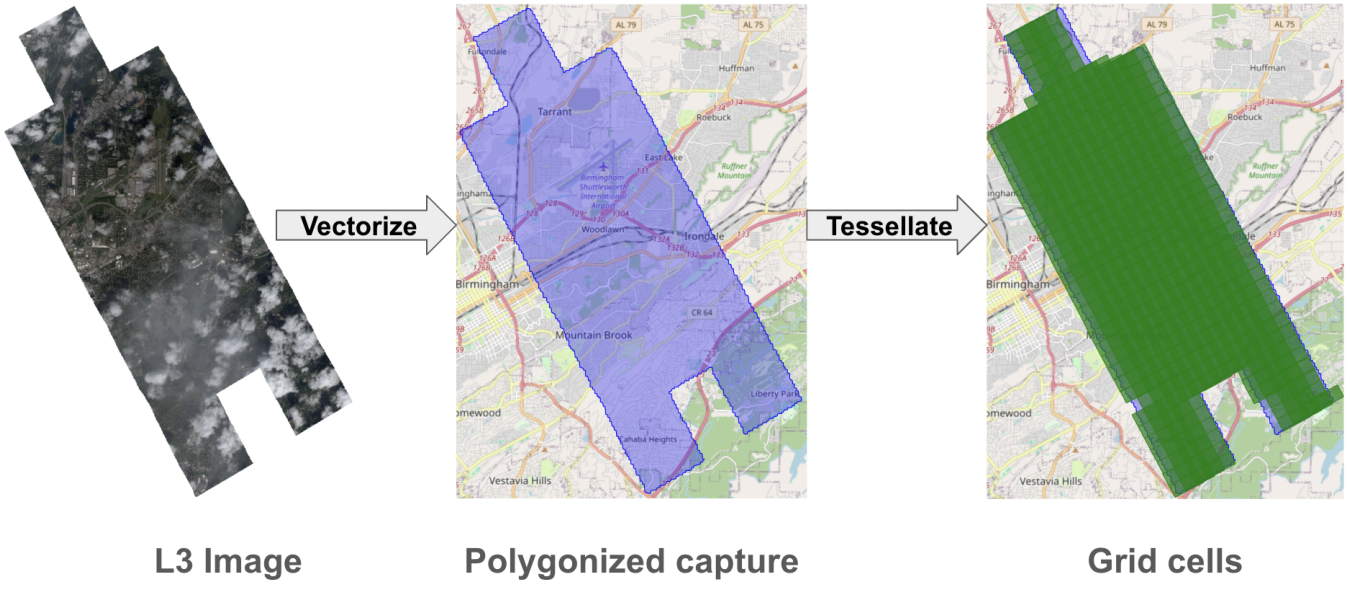

Figure 4. Example grid cell generation used for generating the embeddings of an L_3 image. The L_3 image is vectorized into a polygonized capture and then tessellated into grid cells. These grid cells are then used to generate embeddings for the particle filter.

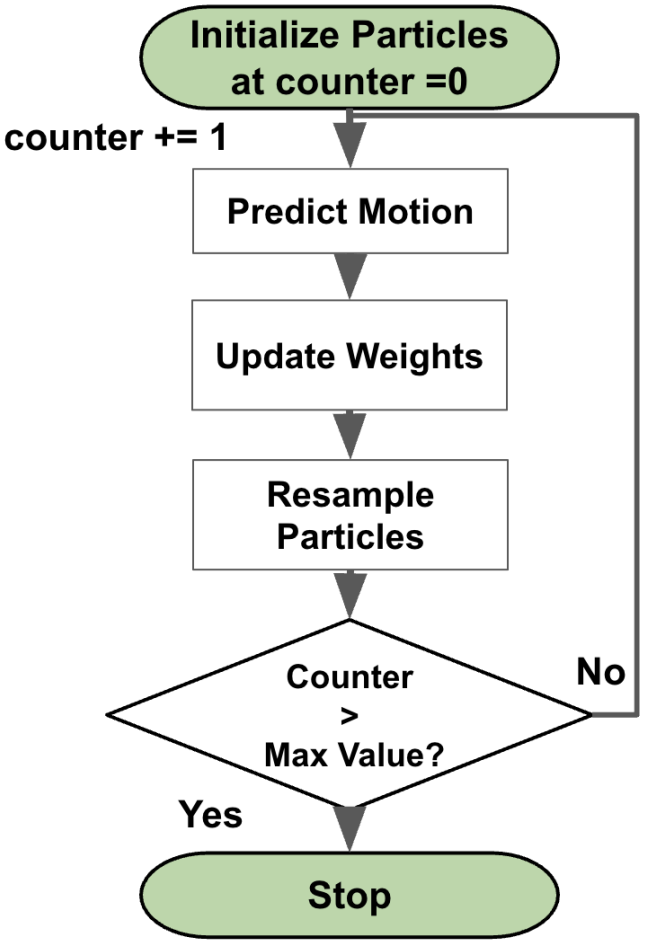

Figure 5. Particle filter workflow for coarse localization. After initializing particles each iteration performs motion prediction, weight update based on image–reference similarity, and resampling proportional to the updated weights. The loop repeats until the iteration counter exceeds a specified maximum, at which point processing stops.

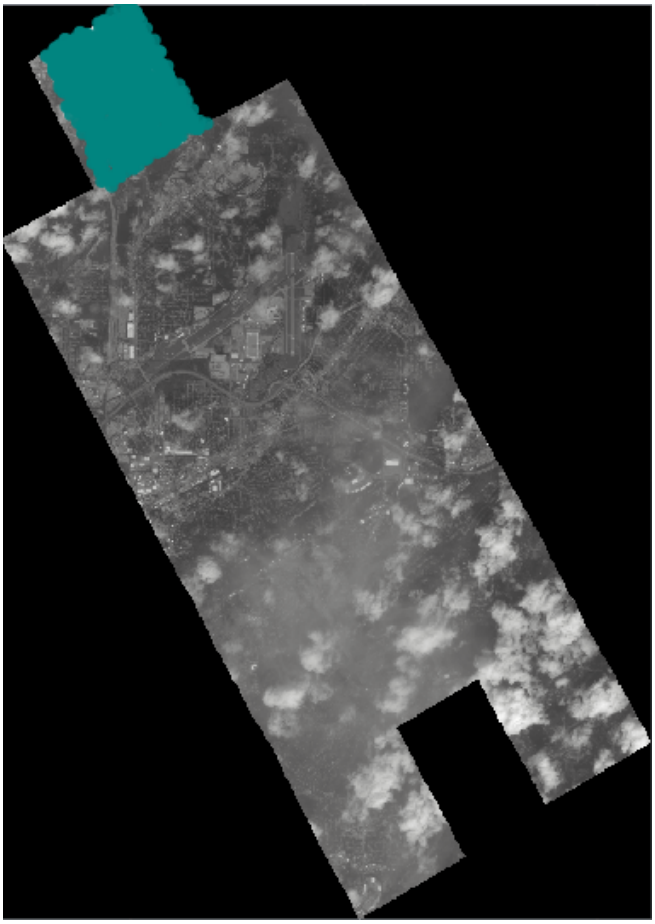

Figure 6. An example initialization of 1000 particles on a Cam 2 strip of an L_3 image. All particles (in light green) initialize around the region with highest similarity in the L_3 image.

$$\sigma_j = 2\sqrt{\sigma_{x_j}^2 + \sigma_{y_j}^2}, \tag{17}$$

where $(\hat{x}_j, \hat{y}_j)$ is the estimated boresight position, $\sigma_{x_j}^2$ and $\sigma_{y_j}^2$ are the coordinate variances, and σ_j is the approximate 95th percentile confidence radius.

Particle Resampling— Resampling generates a new particle set for the next iteration. A cumulative sum (inverse CDF) scheme draws N_p samples proportional to the current weights, yielding a population biased toward high weight hypotheses. After resampling, weights are reset uniformly as in Equation 8. The predict–update–resample cycle then continues for the remaining L_0 frames in the camera sequence (Figure 5).

6. Fine Rectification

After the particle filter provides a coarse localization of the incoming image, a fine georectification stage assigns geographic coordinates to individual pixels. This stage aligns the unprocessed L_0 image to a reference L_3 product for which a map projection and geotransform are known.

The alignment proceeds by extracting keypoints and descriptors from the L_0 image with SIFT [13]. Other detectors and descriptors such as FAST [33], BRIEF [34], or FREAK [35] are compatible. SIFT was chosen for this example due to superior accuracy compared to other methods. Descriptors are matched against a preprocessed database of keypoints computed once from the L_3 reference. The coarse pose prior restricts the search region and expected scale range, which reduces run time and improves match quality. Tentative correspondences are filtered with a ratio test and mutual checks, then verified geometrically.

The L_3 keypoint descriptors are stored with explicit geographic coordinates, forming a library of thousands of geo referenced descriptors. The coarse localization defines a search space within this library to limit candidate matches, reduce memory traffic, and lower the chance of false positives. This search space is derived from an experimental characterization of the coarse estimator, where an average covariance of the pose errors sets its extent. In practice, the spatial window is computed at the L_3 ground resolution and

padded by three standard deviations of the coarse error in each dimension, which yields a compact candidate set while preserving high recall.

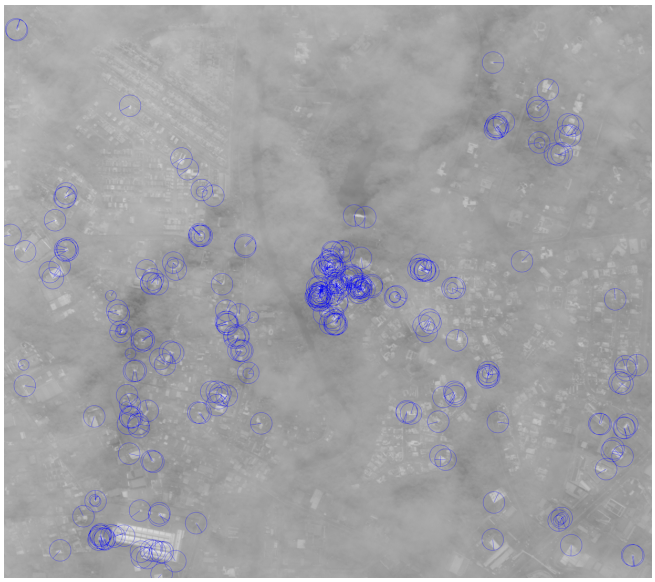

Figure 7. Keypoints (in this image, blue ORB keypoints) and descriptors store information of salient image features from an L_3 image and also detect the same features in a different L_0 image

Verified matches support the estimation of a homography that maps pixel coordinates in L_0 to the L_3 image. Because the L_3 product already carries a mapping from pixels to geographic coordinates through its geotransform, composing the homography with that mapping yields geographic coordinates for each L_0 pixel. Where the L_3 mapping is locally affine in the projected space, this composition is straightforward and can be applied at raster or feature locations.

Several parameters govern accuracy and throughput. Useful controls include the number of octaves and the contrast and edge thresholds, the descriptor distance metric, the ratio test threshold, and the inlier threshold and maximum iterations in the robust homography estimator, for example RANSAC. Minimum inlier counts, keypoint density, non maximum suppression, and limits on match ambiguity help in low texture or repetitive areas. The coarse localization prior can also bound candidate orientations and scales to further reduce run time.

Feature-based methods (e.g., SIFT) remain effective under modest cloud cover because they anchor on repeatable local structure—terrain edges, road networks, and building corners. Heavy cloud or haze occludes texture and predictably degrades matching. To improve robustness to seasonal change (vegetation state, snow, shadow length), the reference index should include multi-temporal imagery, and keypoints should be drawn across seasons rather than from a single date.

The approach assumes that the scene is approximately planar over the region of interest, so relief and off nadir viewing can introduce residual error. On relatively flat terrain or at high altitude, this assumption is often adequate. In areas with significant topography, better results were obtained with an as-projective-as-possible [14] piecewise warp model. In such cases, the piecewise model compensates for terrain warp at the expense of longer computation times.

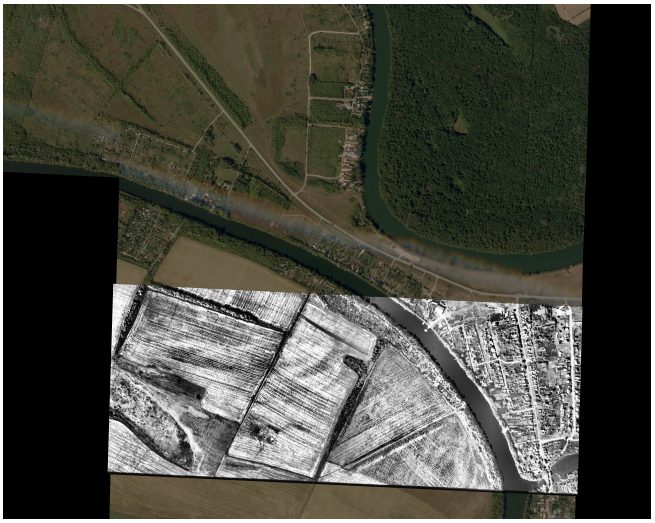

Figure 8. The homography overlays the unprocessed L_0 image (in grayscale) onto a georectified L_3 (in color), making use of the L_3's geographical mapping of each pixel to rectify the L_0.

7. Experiments

We evaluated the approach on an archival set of publicly available SkySat scenes with a nominal ground sampling distance of 0.5 m. For a trial pass, we selected five geographically distinct locations and processed each as a sequence of approximately 150 consecutive images. The pipeline was run without location-specific tuning beyond the coarse localization stage described earlier.

Coarse Localization

The parameters used to setup the particle filter are presented in Table 1. We used 4 datasets to evaluate the performance of the pipeline, resulting in a total of 10 cloud-free with all the three cameras. It was found that datasets with images that had clouds, and poor-brightness would result in poor performance of the the particle filet. Processing the camera datasets separately allowed us to filter-out these performance losses. Each dataset had $180 - 684$ L_0 images per camera. The coarse localization ran on a MacBook Pro with an M4 Max processor with 36 GB memory, where the total processing time was about 130 ms per L_0 image, and about 63 ms of this was spent on the particle filter loop.

Figures 9-11 present the results of coarse localization on each of the three cameras respectively. In of these cases, the predicted camera boresight was confined to the camera lanes as seen here. As noted here, the particles tend to converge to tightly packed clusters, due to the resampling step of the particle filter. It must be noted here that the estimated boresight track is essentially formed from the particles, where the ResNet embeddings of the L_0 and L_3 match. Figures 10, and 11 also show some effects of running coarse localization on datasets with clouds: We see particles avoiding the clouds and trying to cluster around brighter areas nearby, resulting

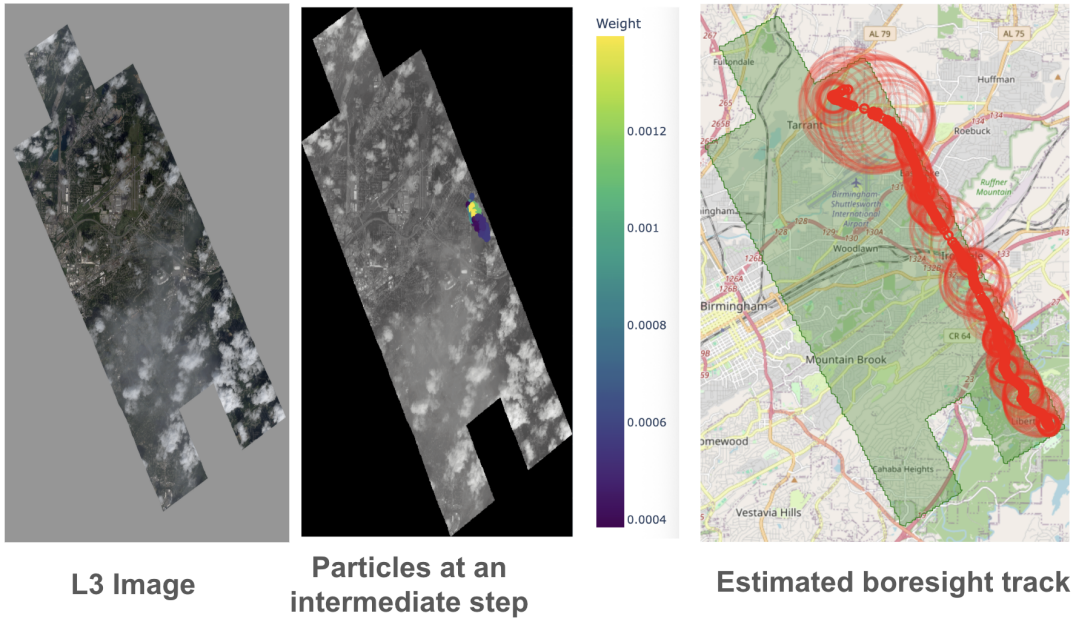

Figure 9. Example Coarse localization on a Cam 1 dataset showing the L_3 image (left), particle filter at an intermittent timeframe (center), and the estimated boresight trajectory (right)

Table 1. **Parameters used to setup the Particle Filter.**

Parameter	Value
L_0 Height	1080 px
L_0 Width	2560 px
Noise, σ_p	50 px
Number of particles, N_P	1000
% Overlap	90%
Temperature, T	0.1

Table 2. **Parameters used to setup SIFT matching**

Parameter	Value
nFeatures	0
nOctaveLayers	3
σ_s	1.6
contrastThreshold	0.04
edgeThreshold	10
kNN inlier threshold	0.6

in an apparent drift. However, this effect is then compensated during the Fine rectification step provided the image contains sufficient features.

Fine Georectification

After the particle filter converges, fine georectification uses the estimate of all boresight tracks to calculate the search space of keypoints for each image. For qualitative visualization, we overlay the L_0 frames on the L_3 reference used in the fine stage. The L_0 images are shown in grayscale while the L_3 image remains in color to make residual misalignments apparent. Figure 2 displays one full pass as a swath of L_0 footprints over the L_3 background. In this example we use SIFT [13] for highest match quality; Table 2 lists the parameter settings. After estimating a homography from SIFT inliers between each L_0 frame and its matched L_3 tile, we warp the L_0 image into the L_3 frame and assess alignment using the same SIFT configuration. Correspondences against the georeferencing reference provide an independent planimetric error estimate. Over a dataset of 169 images, the mean root-mean-square error is 6.5 meters, which corresponds on average to 13 pixels at 0.5 m GSD.

Quality control follows a consistent procedure. For each frame, we use the estimated homography to warp the L_0 image into the coordinate system of its L_3 reference. We then perform SIFT matching between the warped L_0 and the L_3 image. For every verified correspondence, we compare pixel coordinates and compute the residual. The average root mean square residual in pixels is converted to geographic distance using the known ground sampling distance, which yields an error estimate in meters for that frame. Aggregating these per frame values over a pass provides summary statistics for each site.

On a MacBook with an M4 Max processor, processing an entire scene of roughly 150 images completed in about one minute with the parameter settings used here in a single core CPU workload. For context, a workflow based on traditional rational function models and full orthorectification, including bias refinement and resampling over a digital elevation model, required hours to days in our environment. While absolute timings depend on implementation details and hardware, the relative difference was consistent across the five sites.

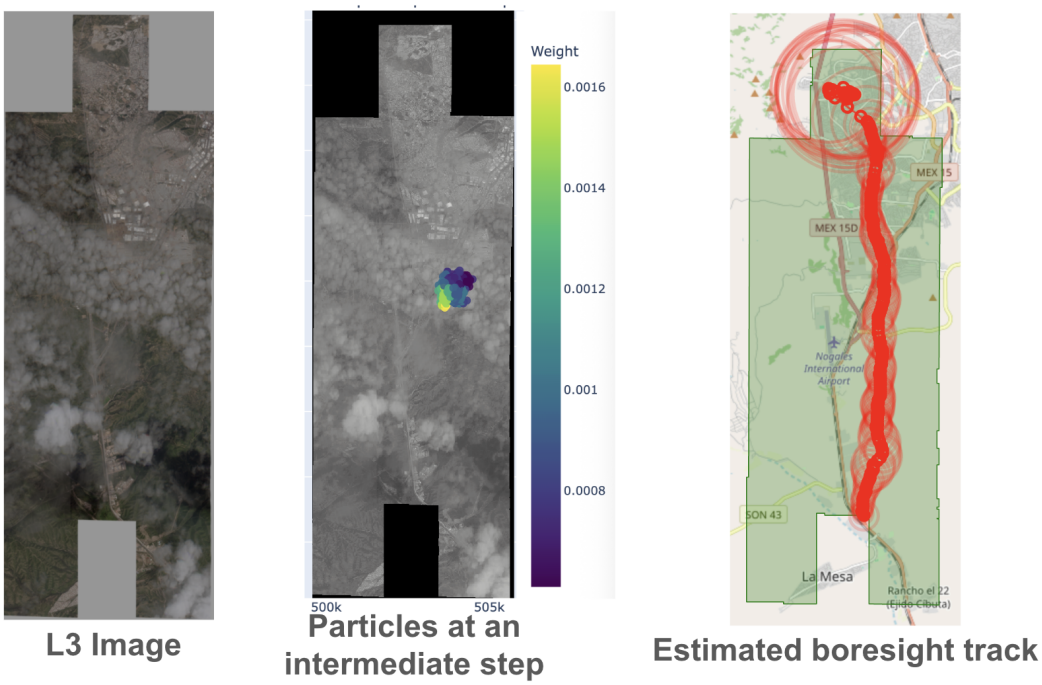

Figure 10. Example Coarse localization on a Cam 2 dataset showing the L_3 image (left), particle filter at an intermittent timeframe (center), and the estimated boresight trajectory (right)

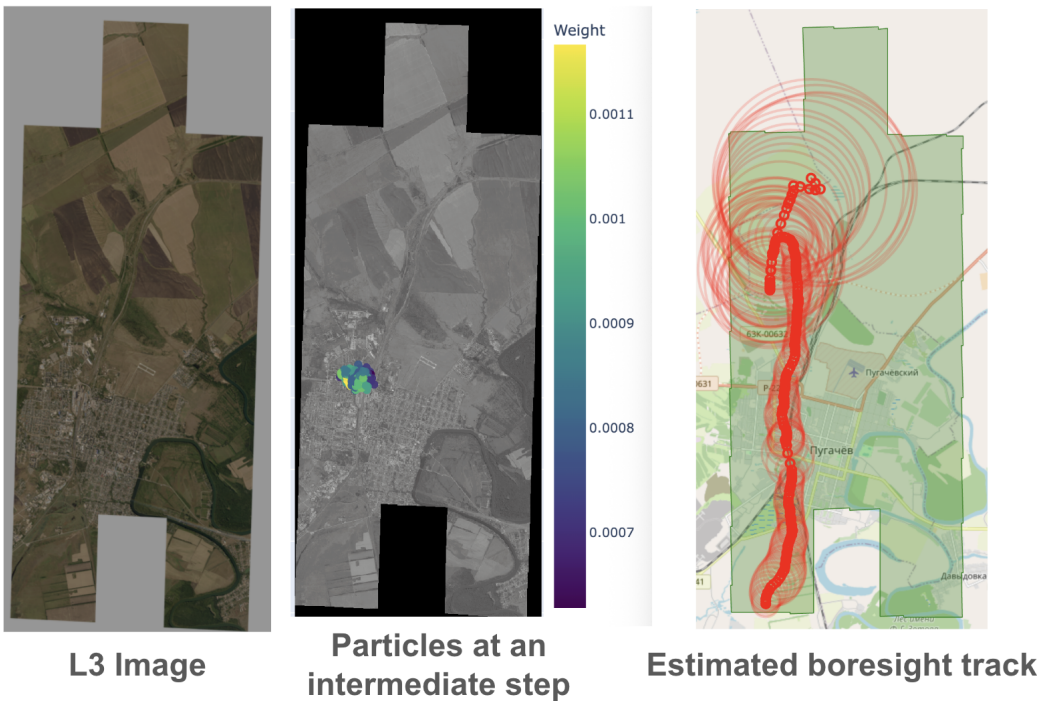

Figure 11. Example Coarse localization on a Cam 3 dataset showing the L_3 image (left), particle filter at an intermittent timeframe (center), and the estimated boresight trajectory (right)

8. Conclusions

The proposed method offers substantially lower runtime than classical approaches while avoiding reliance on ground control points and remaining model agnostic with respect to sensor- and vendor-specific camera models. In practice, it processes a collection in approximately one minute, enabling fast decision making in time-sensitive settings such as rapid mapping, event monitoring, and tasking feedback. By elimi-

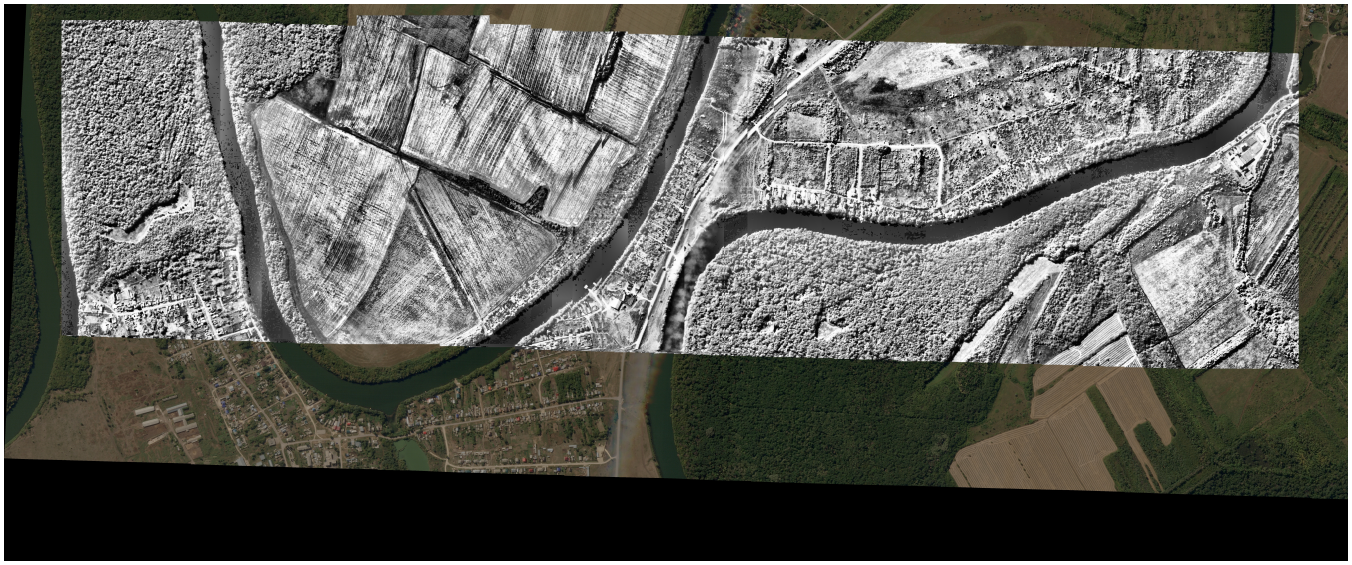

Figure 12. Example of a georectified L_0 image (in grayscale) on top of a L_3 product (in color). Notice how the features around the edges align between the two images. An RMS of 6.5m was achieved.

nating site-specific control and detailed model calibration, the pipeline reduces pre-deployment overhead and simplifies integration across heterogeneous sensors and archives. The low computational cost further enables deployment on resource-constrained platforms and supports large-scale production without extensive infrastructure.

These properties make the method well suited for scenarios where timeliness and operational simplicity are primary constraints. However, several aspects merit further study. Additional evaluation under noisier reference satellite data is needed to fully characterize robustness. While the particle filter is expected to tolerate partial cloud coverage, a systematic analysis of varying cloud conditions and their effect on localization accuracy remains future work. Seasonality effects, including changes in vegetation and appearance, may also impact embedding consistency and should be explicitly evaluated. Finally, although strong performance is demonstrated with the current embedding architecture, the framework is not architecture dependent, and exploring alternative embedding models, such as transformer-based approaches, represents a promising direction for improving robustness.

REFERENCES

[1] F. Eugenio and F. Marqués, "Automatic satellite image georeferencing using a contour-matching approach," *IEEE transactions on geoscience and remote sensing*, vol. 41, no. 12, pp. 2869–2880, 2004.

[2] A. Hackeloeer, K. Klasing, J. M. Krisp, and L. Meng, "Georeferencing: a review of methods and applications," *Annals of GIS*, vol. 20, no. 1, pp. 61–69, 2014.

[3] D. Phiri, M. Simwanda, S. Salekin, V. R. Nyirenda, Y. Murayama, and M. Ranagalage, "Sentinel-2 data for land cover/use mapping: A review," *Remote sensing*, vol. 12, no. 14, p. 2291, 2020.

[4] J.-N. Thépaut, D. Dee, R. Engelen, and B. Pinty, "The copernicus programme and its climate change service," in *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2018, pp. 1591–1593.

[5] J. Blumenfeld, "Wildfires can't hide from earth observing satellites," *NASA EOSDIS*, 2019.

[6] G. Ifimov, T. Naprstek, J. M. Johnston, J. P. Arroyo-Mora, G. Leblanc, and M. D. Lee, "Geocorrection of airborne mid-wave infrared imagery for mapping wildfires without gps or imu," *Sensors*, vol. 21, no. 9, p. 3047, 2021.

[7] M. Weiss, F. Jacob, and G. Duveiller, "Remote sensing for agricultural applications: A meta-review," *Remote sensing of environment*, vol. 236, p. 111402, 2020.

[8] D. Poli and T. Toutin, "Review of developments in geometric modelling for high resolution satellite pushbroom sensors," *The Photogrammetric Record*, vol. 27, no. 137, pp. 58–73, 2012.

[9] C. V. Tao and Y. Hu, "A comprehensive study of the rational function model for photogrammetric processing," *Photogrammetric engineering and remote sensing*, vol. 67, no. 12, pp. 1347–1358, 2001.

[10] S. Aati and J.-P. Avouac, "Optimization of optical image geometric modeling, application to topography extraction and topographic change measurements using planetscope and skysat imagery," *Remote Sensing*, vol. 12, no. 20, p. 3418, 2020.

[11] P. d'Angelo, G. Kuschk, and P. Reinartz, "Evaluation of skybox video and still image products," *ISPRS Archives*, vol. 40, pp. 95–99, 2014.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.

[13] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2. Ieee, 1999, pp. 1150–1157.

[14] J. Zaragoza, T.-J. Chin, M. S. Brown, and D. Suter, "As-projective-as-possible image stitching with moving dlt," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 2339–2346.

[15] J. Grodecki and G. Dial, "Block adjustment of high-resolution satellite images described by rational polynomials," *PERS*, 2003. [Online]. Available: https://www.asprs.org/wp-content/uploads/pers/2003journal/january/2003_jan_59-68.pdf

[16] C. Fraser and H. Hanley, "Bias-compensated rpcs for sensor orientation of high-resolution satellite imagery," *PERS*, 2005. [Online]. Available: https://www.asprs.org/wp-content/uploads/pers/2005journal/aug/2005_aug_909-915.pdf

[17] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *CVPR Workshops*, 2018. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2018_workshops/papers/w9/DeTone_SuperPoint_Self-Supervised_Interest_CVPR_2018_paper.pdf

[18] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *CVPR*, 2020. [Online]. Available: https://openaccess.thecvf.com/content_CVPR_2020/papers/Sarlin_SuperGlue_Learning_Feature_Matching_With_Graph_Neural_Networks_CVPR_2020_paper.pdf

[19] L. Li *et al.*, "Deep learning in remote sensing image matching: A survey," *ISPRS J. Photogramm. Remote Sens.*, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0924271625001376

[20] G. Berton, C. Masone, and B. Caputo, "Rethinking visual geo-localization for large-scale applications," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 4878–4888.

[21] "gdalwarp — gdal documentation," 2025. [Online]. Available: https://gdal.org/en/stable/programs/gdalwarp.html

[22] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*. Ieee, 2011, pp. 2564–2571.

[23] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3304–3311.

[24] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, "Loftr: Detector-free local feature matching with transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 8922–8931.

[25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[27] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: concepts, cnn architectures, challenges, applications, future directions," *Journal of big Data*, vol. 8, pp. 1–74, 2021.

[28] A. Johansen, "A tutorial on particle filtering and smoothing: Fifteen years later," 2009.

[29] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on signal processing*, vol. 50, no. 2, pp. 174–188, 2002.

[30] G. Terejanu, P. Singla, T. Singh, and P. D. Scott, "A novel gaussian sum filter method for accurate solution to the nonlinear filtering problem," in *2008 11th International Conference on Information Fusion*. IEEE, 2008, pp. 1–8.

[31] S. Thrun, "Particle filters in robotics." in *UAI*, vol. 2, 2002, pp. 511–518.

[32] J. P. Snyder, "The space oblique mercator projection," *Photogramm. Eng. Remote Sensing*, vol. 44, no. 585-596, p. 140, 1978.

[33] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European conference on computer vision*. Springer, 2006, pp. 430–443.

[34] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *European conference on computer vision*. Springer, 2010, pp. 778–792.

[35] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint," in *2012 IEEE conference on computer vision and pattern recognition*. Ieee, 2012, pp. 510–517.

BIOGRAPHY

Paulo R. M. Fisch is a PhD candidate in the Robotics Institute at Carnegie Mellon University. He has previous experience working at the German Aerospace Center (DLR) and got his Mechanical Engineering degree from the University of São Paulo in 2020. His interests include satellite Guidance and Navigation Systems, computer vision, optimal control and state estimation for space systems, with recent work on satellite orbit determination.

Ravi teja Nallapu is a Senior Orbits R&D Engineer at Planet, where he works on mission design for Planet's upcoming constellations. He was the mission design engineer for the Tanager and Pelican missions. Prior to this, he worked as a Flight Simulator Engineer with American Airlines where he worked on the maintenance of commercial aircraft simulators. He received his Ph.D. in aerospace engineering from the University of Arizona in 2020., and an M.S. in aerospace engineering from the University of Houston in 2012. His research interests include constellation design, on-orbit demonstrations, and trajectory optimization.

Punarjay Chakravarty (Jay) is a Staff Engineer at Planet Labs applying CV/ML expertise to develop real-time insights for Earth observation satellites. He holds a PhD in robotics from Monash University, where he specialized in collaborative surveillance between fixed cameras and mobile robots. Previously, he served as a Machine Learning and Robotics lead at Ford Autonomous Vehicles LLC, leading a perception team that demonstrated fully autonomous vehicle operation in large parking garages and filing 55 patents in 5 years. He also worked as a post-doctoral research fellow at KU Leuven on the Cametron project for autonomous film-making with drones. His research has produced 40+ peer-reviewed papers in top computer vision and robotics conferences with over 3000 citations.

Kiruthika Devaraj is a VP of Engineering at Planet where she leads the spacecraft team that designs and builds Planet's satellites. Over the last ten years, she has led Planet's dove and smallsat avionics platforms, radio and communication systems, advanced payloads including Real Time Insights using next-gen GPUs and onboard AI, as well as as real time connectivity onboard satellites using intersatellite links. Prior to Planet, Kiruthika was a radio astronomer at Stanford University and the National Radio Astronomy Observatory where she built instruments that were deployed at multiple radio telescopes. She has a M.S. and Ph.D. in EE from Georgia Tech where she worked on NASA mission Juno supporting the microwave radiometer instrument to study the deep atmosphere of Jupiter.

Zachary Manchester is an assistant professor in the Robotics Institute at Carnegie Mellon University and founder of the Robotic Exploration Lab. He received a PhD in aerospace engineering in 2015 and a BS in applied physics in 2009, both from Cornell University. His research interests include control and optimization with application to aerospace and robotic systems with challenging nonlinear dynamics.