Traditional techniques for feature detection include:

- **Harris Corner Detection —** Uses a Gaussian window function to detect corners.
- **Shi-Tomasi Corner Detector —** The authors modified the scoring function used in Harris Corner Detection to achieve a better corner detection technique.
- **Scale-Invariant Feature Transform (SIFT) —** This technique is scale invariant unlike the previous two.
- **Speeded-Up Robust Features (SURF) —** This is a faster version of SIFT as the name says.
- **Features from Accelerated Segment Test (FAST) —** This is a much more faster corner detection technique compared to SURF.
- **Binary Robust Independent Elementary Features (BRIEF) —** This is only a feature descriptor that can be used with any other feature detector. This technique reduces the memory usage by converting descriptors in floating point numbers to binary strings.
- **Oriented FAST and Rotated BRIEF (ORB)** — SIFT and SURF are patented and this algorithm from OpenCV labs is a free alternative to them, that uses FAST KeyPoint detector and BRIEF descriptor.

Glimpse of Deep Learning feature extraction techniques

Traditional feature extractors can be replaced by a convolutional neural network (CNN), since CNN's have a strong ability to extract complex features that express the image in much more detail, learn the task specific features and are much more efficient. Multiple works have been done on this. Few of them are listed below:

- **SuperPoint: Self-Supervised Interest Point Detection and Description** — The authors suggest a fully convolutional neural network that computes SIFT like interest point locations and descriptors in a single forward pass. It uses an VGG-style encode for extracting features and then two decoders, one for point detection and the other for point description.

- **D2-Net: A Trainable CNN for Joint Description and Detection of Local Features —** The authors suggest a single convolutional neural network that is both a dense feature descriptor and a feature detector.
- **LF-Net: Learning Local Features from Images —** The authors suggest using a sparse-matching deep architecture and use an end-to-end training approach on image pairs having relative pose and depth maps. They run their detector on the first image, find the maxima and then optimize the weights so that when run on the second image, produces a clean response map with sharp maxima at the right locations.
- **Image Feature Matching Based on Deep Learning** — They adopt a deep Convolutional neural network (CNN) model, which attention on image patch, in image feature points matching.
- **Deep Graphical Feature Learning for the Feature Matching Problem** — They suggest using a graph neural network to transform coordinates of feature points into local features, which would then make it easy to use a simple inference algorithm for feature matching

Data Preprocessing

1. Data Cleaning/Cleansing

Data Cleaning/Cleansing routines attempt to fill in missing values, smooth out noise while identifying outliers, and correct inconsistencies in the data. Data can be noisy, having incorrect attribute values. “Dirty” data can cause confusion for the mining procedure. Although most mining routines have some procedures, they deal incomplete or noisy data, which are not always robust. Therefore, a useful Data Preprocessing step is to run the data through some Data Cleaning/Cleansing routines.

2. Data Integration

Data Integration is involved in data analysis task which combines data from multiple sources into a coherent data store, as in data warehousing. These sources may include multiple databases, data cubes, or flat files. Databases and data warehouses typically have metadata.

Metadata is used to help avoiding errors in schema integration. Another important issue is redundancy. An attribute may be redundant, if it is derived from another table. Inconsistencies in attribute or dimension naming can also cause redundancies in the resulting data set.

3. Data Transformation

Data are transformed into appropriate forms of mining.

4. Data Reduction

Complex data analysis and mining on huge amounts of data may take a very long time, making such analysis impractical or infeasible. Data Reduction techniques are helpful in analyzing the reduced representation of the data set without compromising the integrity of the original data and yet producing the qualitative knowledge.

Image Feature Extraction

Each drone image has a collection of unique features which differentiate it from other images. These are known as key points. Key points from each image are extracted using automatic computer vision algorithms (SIFT, BRISK, etc.) These features consist of the building's corner, roads, edges, etc.

Generally, images with good texture variation have 40,000+ features. It can be easily understood why photogrammetry performs poor in areas of low texture variation like water bodies, dense forest, sand, sky etc. KeyPoint's extraction becomes difficult in texture less surfaces.

Feature Matching

Extracted features are then searched (in the nearby images) and matching is performed. Using GPS data to search relevant images makes the matching process much faster and accurate. From matched features, fundamental matrix is derived and the relative position between two cameras is estimated. Techniques like Flann is often used to conduct search and match.

Image stitching

Photo stitching is the process of combining multiple photographic images with overlapping fields of view to produce a segmented panorama or high-resolution image.

References:

https://medium.com/@mehulved1503/feature-selection-and-feature-extraction-in-machine-learning-an-overview-57891c595e96

https://www.thepythoncode.com/article/sift-feature-extraction-using-opencv-in-python#:~:text=SIFT%20stands%20for%20Scale%20Invariant,scale%20and%20other%20image%20transformations.

https://www.geospatialworld.net/blogs/what-happens-when-we-process-drone-data-under-the-hood/#:~:text=Drone%20data%20is%20processed%20using,volume%20etc%20from%20an%20image.