Seamless Oblique Image Mosaics for Aerial Visualization (P2 Submission)

Xiaoluo Gong
student #5923476

1st supervisor: Martijn Meijers
2nd supervisor: Edward Verbree
External supervisor: Annemieke Verbraeck (Geodelta B.V.)

June 19, 2025

1 Introduction

Oblique aerial images are captured with an inclined camera angle, offering multiple perspectives of the areas of interest (Verykokou and Ioannidis (2024)). Oblique aerial images provide more intuitive and stereoscopic views of the structures and landscapes than traditional nadir images, revealing details that are often obscured from the vertical view. In addition, compared to panoramas such as Google street view, which can also offer a 3D-like visualization experience and is usually collected by cars or handheld devices, oblique aerial images can cover a larger range with better time efficiency and lower financial cost. Meanwhile, with a larger view field than the panorama, oblique aerial images have less chance to cause dizziness after viewing(Armstrong et al. (2016)). As moving through panoramas offer users the view of walking or driving in the street, the continuous exploring of oblique images offers users the view of flying across the city. With these advantages, oblique aerial images are particularly valuable in fields such as urban planning, heritage protection, and disaster management (Zhang et al. (2023); Höhle (2013); Tang et al. (2024)).

Geodelta has developed a comprehensive platform, Omnibase, which offers combined visualization and easy measurements of point cloud, panorama, nadir images, oblique images, and 3D meshes (Geodelta (2024)). Omnibase presents the oblique images by assigning them separately to the accurate locations and range on the map (see Figure 1). To give users a smoother and continuous experience when navigating through consecutive oblique aerial images, seamless stitching of the images is essential to improve image visualization.

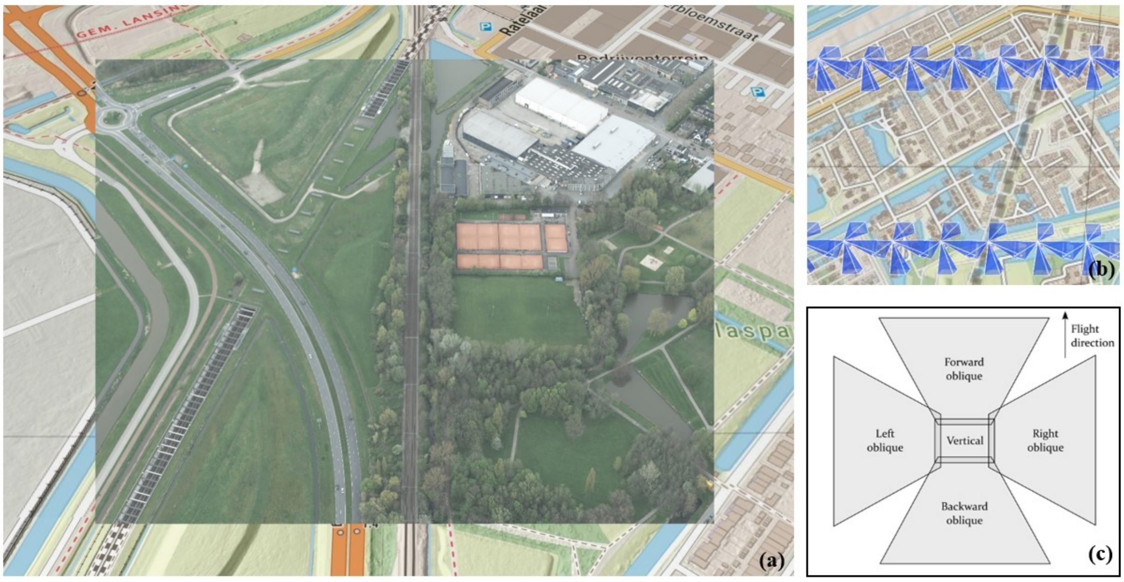

Figure 1: (a):an oblique aerial image accurately superimposed on the base map, source: Omnibase; (b):overview of camera frustums of both oblique and nadir aerial images, source: Omnibase; (c) the Maltese cross ground coverage pattern for the camera frustums (Verykokou and Ioannidis (2024))

For general applications, image stitching is the process that can combine different images by their overlapped areas to obtain an integrated image with a wide view (Wang and Yang (2020)). However, in the case of aerial oblique images, each individual image already covers a large area and has a significant file size. Stitching all oblique images in a region into a single integrated map is impractical due to both computational and visualization challenges. Computationally, such a map would be difficult to process because of its massive size. From a visualization perspective, warping all images to fit a single reference frame can result in significant distortions.

This M.Sc. thesis focuses on a different approach: stitching aerial images to provide smooth and continuous transitions when switching between adjacent oblique images. This dynamic process ensures seamless navigation between images rather than creating a static large-scale integrated map. At least two images with the same view direction must be stitched together to ensure a smooth transition between images. More images in the same column or row may also be stitched together to create a wider, belt-like view. Users can stop at any intermediate stage of their moving actions to do the measurement, especially measuring those objects that are divided into two parts by different image frames. Pre-calculations of some parameters like the transformation matrix between two images will be done to achieve a fast real-time performance.

Stitching aerial oblique images is a challenge in several aspects. First, oblique images have perspective distortion depending on the varying distance from the camera to the ground objects (see Figure 2). Second, parallax can cause ghosting and blurring when stitching pictures (LYU et al. (2019)), especially when the parallax is very large. Practical issues like real-time processing of morphing huge aerial photos and finding key points from monotonous landscape like water can also be potential challenges. Furthermore, considering the coverage pattern for the camera frustums (see Figure 1,c), stitching images across the flight line can also be challenging, because images with the same direction of the camera frustum may have small overlap area, while images with a large overlap may have the opposite direction of the camera frustums.

Figure 2: Oblique aerial images of the same area in Utrecht with different perspective distortion, source: Omnibase

To solve these problems, image processing technologies including tie-point matching and morphing are needed. This research also incorporates the use of 3D models, especially 3D coordinates of control ground points and point cloud, in the stitching process. By aligning the images to a 3D reference and using 3D control points, this approach ensures more accurate tie-point matching and improved stitching results. The process only applies perspective distortion to the aerial images ,thus the final results will preserve the feature of straight lines, which can help to apply precise measurement on the images. With a combination of those technologies, the main goal of this research is to create seamless mosaics from oblique aerial images, allowing for a continuous and immersive aerial view experience, as well as minimum errors in measurements through the oblique aerial images.

2 Related work

2.1 Image Stitching Techniques

Multiple image stitching techniques are applied to different scenarios, including merging images taken by daily mobile devices(Laraqui et al. (2017)), generating panoramic images(Brown and Lowe (2007)), stitching large-scale nadir aerial images(Pham et al. (2021)), etc. In addition to the specific techniques used in different scenarios, there are some common basic steps in image stitching, including detecting key points, matching the corresponding points, and image warping.

2.1.1 Key Points Detecting

Distinctive invariant features are considered the key factors that can be used to decide the correspondence between different images. Many algorithms have attempted to detect and match key points that are invariant in scale and rotation. Some widely used algorithms are presented and discussed in this section.

- **Scale Invariant Feature Transform (SIFT) by Lowe (2004):** First, the Gaussian filter is applied to the input image to find the scale space of the image. The local maxima and minima are calculated based on the difference of Gaussians (DoG) and are considered as the potential key points that are invariant to the scale. After selecting the potential key points. Finally, a localization algorithm refines the key point selection by fitting them more precisely to nearby data. Every remaining high-contrast key point is assigned one orientation based on local image gradient directions. Finally, each output key point would have a highly distinctive descriptor. The descriptors are used to match the corresponding key points in different images.

- **Speeded-Up Robust Features (SURF) by Bay et al. (2008):** Partly inspired by SIFT, SURF has steps similar to SIFT, and some adjustments are applied in each step of feature detecting. First, it uses box Gaussian filters to create the scale space and a Hessian matrix as a measure of local change around a single point. The point will be chosen when its Hessian determinant is maximal. To make sure the descriptor of the key point is rotation invariant, SURF calculates the Haar wavelet responses in the x and y directions within a circular neighborhood around the point of interest. SURF will construct a square region around the detected key point and divide it into 4×4 subregions. Then, the Haar wavelet responses are summed up over each subregion and form a feature vector as descriptor. According to the author of SURF, the SURF detector has a faster speed, which is the main improvement and makes it suitable for real-time computation.

- **Oriented FAST and rotated BRIEF (ORB) by Rublee et al. (2011):** This method is built on the FAST detector(Rosten and Drummond (2006)) and the BRIEF descriptor(Calonder et al. (2010)). The original FAST detector used machine learning to find key points, but did not include the orientation component. The ORB measured the orientation by intensity centroid and added this attribute to the key points detected by FAST. BRIEF, as a binary string key point descriptor, is computationally efficient, but performs badly with rotation. The ORB steered the BRIEF according to the orientation of key points, thus improving the BRIEF performance in rotated images. ORB is much faster than SURF and SIFT, but it still has problems with scale invariance.

- **DIScrete Keypoints (DISK) by Tyszkiewicz et al. (2020):** DISK used reinforcement learning and achieved end-to-end feature matching for a large amount of points. It first used a U-Net (Ronneberger et al. (2015)) based architecture to extract a detection heatmap and then divided this heatmap into grid cells. The key points are sampled within each cell using softmax normalization and a sigmoid-based quality filter. Using reinforcement learning, DISK is trainable and flexible for specific tasks, compared to constructing hand-crafted features such as SIFT and SURF.

2.1.2 Feature Matching

After detecting the key points, matching them with the corresponding images is the next step. The descriptors generated when detecting the features are essential for the matching. For example, a simple use case is to calculate the pairwise distance of all descriptors and find the nearest neighbor as the best match. However, in practice, the real-world images are very noisy, leading to false matches if only the distance of the descriptors is used as a matching reference. Furthermore, comparing all descriptor distances from all key points has a computational time complexity of $O(n^2)$ with a quadratic in the expected number of features. Thus, some more sophisticated techniques are used for feature matching.

- **Random sample consensus (RANSAC):** RANSAC is an iterative algorithm used to estimate parameters of a mathematical model from a dataset that may contain significant outliers (Fischler and Bolles (1981)) and is widely applied in computer vision (Stewart (1999)). It starts by entering key points using a similarity measure, such as the distance for the SIFT descriptors. Then RANSAC will randomly select a minimal subset of correspondences and compute a linear estimate that has the physical meaning of the camera motion. RANSAC will compute the residuals between the real points locations and the estimated locations and will consider those with residuals smaller than the predefined threshold as inliers. The random selection, model fitting, and inlier counting process will repeat until the estimate gets the maximum number of inliers. RANSAC is efficient when processing noisy data and can produce a high-accuracy alighment.

- **Transformer used in feature matching:** The transformer, a well-known deep learning architecture, has been widely applied to feature matching, due to its self and cross attention mechanisms to consider the relationship between all input key points (Carion et al. (2020)). Two of the most famous models for feature matching are SuperGlue (Sarlin et al. (2020)) and LightGlue (Lindenberger et al. (2023)). The transformer as the backbone of these models consists of self-attention layers to aggregate information from other features within the same image, as well as cross-attention layers to aggregate the information from features in the other image. However, built upon SuperGlue, LightGlue also has some differences and improvements compared to SuperGlue. First, LightGlue uses relative positional encoding, which can better capture long-range dependencies. For the scoring system, SuperGlue uses the Sinkhorn algorithm, an iterative method for optimal transport, to compute the correspondence points between two points. In LightGlue, a simpler matchability and similarity scoring system is designed to compute a pairwise score matrix between the points of both images. Additionally, LightGlue also introduces a pruning mechanism. At the end of each layer, a confidence classifier determines if the matching predictions are reliable. If it is confident, the program will stop early to save computation time. Points that are classified as unmatchable are pruned early to focus computation on relevant features. In the experiments by the authors of LightGlue, DISK + LightGlue is proved to be the combination of feature detector and matcher with the highest accuracy. In their experiments, LightGlue shrinks 35% of the running time compared to SuperGlue.

2.1.3 Image Morphing

Image morphing is a technique in computer vision used to achieve a smooth transformation between two images. This process ensures continuity by generating an intermediate representation that blends the two images. It involves two primary steps: warping, which aligns the shapes of the images, and blending, which mix the intensities or colors of the pixels to create a seamless transition (Efros (2016)).

- **Image Warping:** Global warping based on 2D motions can be achieved through global linear transformations of the image. These transformations are represented by a transformation matrix, which can have degrees of freedom (DOF) ranging from 2 to 8. This corresponds to the minimum need for only 1 to 4 pairs of corresponding points to define the transformation. The DOF determines the type of transformation, enabling various operations such as translation, scaling, rotation, shearing, or a combination of these. Figure 3 illustrates the basic types of 2D transformations, including rigid transformations (translation and rotation), similarity transformations (scaling and rotation), affine transformations, and projective transformations. Each type provides progressively more flexibility in warping the image, with projective transformations allowing the highest degree of non-linear adjustment. These global transformations are particularly useful for scenarios where the deformation is uniform across the entire image.

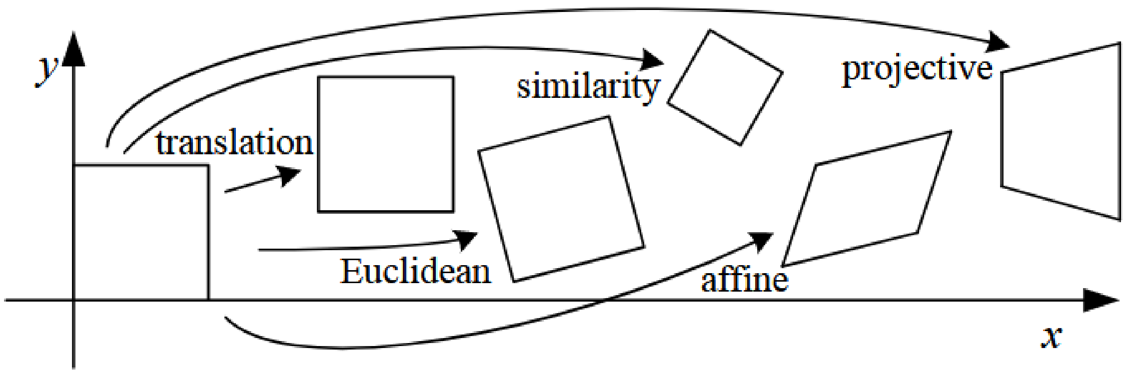

Figure 3: Basic types of 2D transformations(Szeliski et al. (2007))

However, global warping may not suffice when the process involves significant non-linear deformations or complex geometries (Efros (2016)). In such cases, local image warping using mesh-based techniques offers a more effective solution by defining the warp independently for each small mesh. Delaunay triangulations are used to create these meshes, as they maximize minimum angles, prevent edge intersections, and ensure high computational efficiency (Lee and Schachter (1980)). The corresponding triangle meshes are generated based on the feature-matching algorithm's identified points, and a linear transformation is applied to each pair of corresponding triangles.

- **Image Blending:** After aligning the position of two images by warping them correctly, blending the pixel values is the next step to seamlessly make the color transitions. Linear blending, also known as cross-dissolve, is a simple and widely applied method in video editing (Adobe). It generates the new image by adjusting the opacity of one image over the other, which can be presented as the formula below:

$$I_{\text{blend}} = \alpha \cdot I_1 + (1 - \alpha) \cdot I_2$$

I_1 and I_2 are the pixel values of the two images and I_{blend} represents the blended image. α is the weight factor and its value is between 0 and 1, which controls the contribution of each image and produces a gradual transition between the two images.

When the two images to blend have significant differences in color, texture, or brightness, an obvious seam or ghosting can occur. The Laplacian pyramid blend decomposes the images into multiple frequency levels, creates Laplacian pyramids for each image based on the smoothing levels, and blends the pyramids at each level (Burt and Adelson (1983)). This will fill the invalid and edge pixels with neighboring values and minimizes abrupt transitions. The initial pyramids blending operates in the intensity domain, which can struggle with global inconsistencies such as differences in illumination. Pyramids that blend in the gradient domain minimize the seam by optimizing the blending of image gradients, and have a significantly improved performance(Levin et al. (2004)). Gradient domain blending can also be applied to other classic blending methods such as feathering and optimal seam.

2.2 Oblique Aerial Image Visualization Platforms

With an increasing number of oblique images made today and growing needs to view and analyze those data, several platforms have emerged to visualize and interact with these images. Here are the features and functions of some existing platforms.

- **USGS Oblique Aerial Photography Viewer (U.S. Geological Survey (2018)):** It is a web-based platform designed to provide access to oblique aerial imagery captured across the United States. As an interactive viewer, it allows users to explore oblique aerial images within a map interface as a reference to the location where the images are taken, and it is equipped with basic functions for the user to explore with panning and zooming (see figure 4).

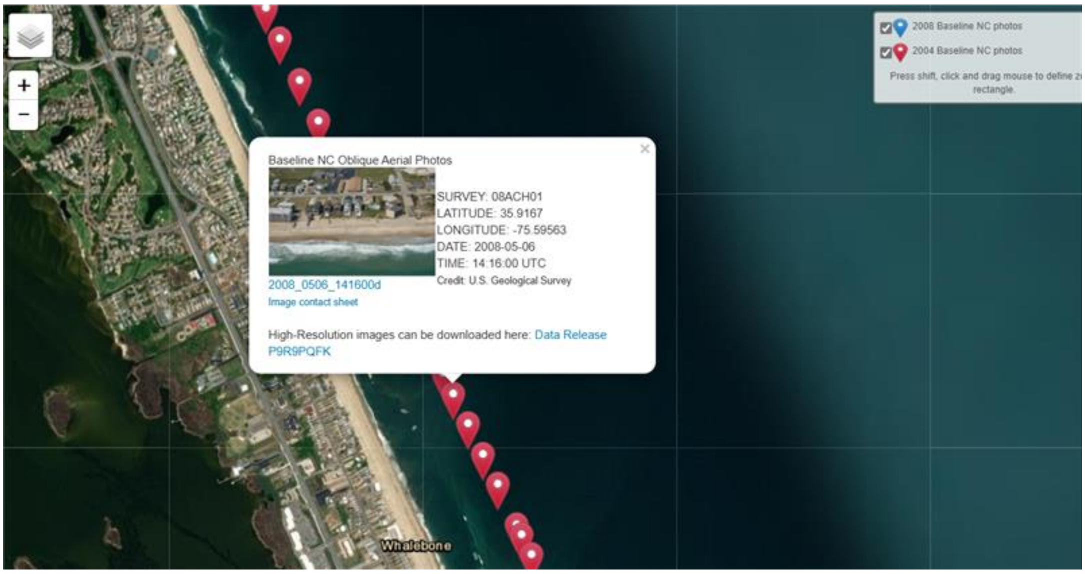

Figure 4: User Interface of USGS Oblique Aerial Photography Viewer

In addition, as a research platform, it offers users access to the meta data and easy download of the images. However, this platform stopped updating after 2018 and it does not stitch or perform any other process on the images for a smoother visualization experience.

- **Bing Maps Bird's Eye (Bing Maps Team (2022)):** Bing Maps released high-resolution oblique aerial images as well as nadir satellite images on its online map platform (see figure 5). They get oblique aerial images georeferred to the basic map and present oblique images when the mouse is moved over the nadir images of the corresponding areas. This platform offers a very continuous experience when switching from a nadir image to its corresponding oblique image. However, it does not create a smooth transition between oblique images. Each time the user can only browse one oblique image. Also, as a platform for the general public, it does not have functions to make precise measurements from the images.

Figure 5: Nadir and oblique images of the same area around Holland Spoor train station in Den Haag, source: Bing Maps

- **XMAP (GeoXphere):** XMAP is a cloud-based GIS platform and the visualization of oblique aerial images is one of its modules. XMAP has built-in cascade and height measurement tools, which can apply accurate measurements for multiple scenarios. In addition, it can change the orientation of the oblique images taken in the same area. Like most of oblique aerial image visualization platforms, it aligns every oblique image with the base map. Although with a few seconds of delay, XMAP still realizes a relatively smooth transition from one oblique image to another. As a commercial software and platform, XMAP is not open source, so we cannot quantitatively assess its image transition or stitch quality.

2.3 Other 3D-like Visualization Platforms

Comparing nadir and oblique images (see Figure 5), we can see that the oblique images can provide users with a 3D-like experience without really building a 3D city model. In the nadir image, it is difficult to perceive and measure the height of the buildings. However, oblique images can show the height and volume of ground objects more intuitively, which also makes measurement directly from images possible. Thus, this section introduces some 3D-like visualization platforms for building environments. Although not directly presenting aerial oblique images, many of these platforms use oblique images as input data and process them with similar techniques as image stitching.

- **Street-level true panoramas:** As a well-known public map platform, Google street view offers panoramas with wide coverage ranging from the main roads of cities to historical sites. The image data are collected by Google and its verified contributors, who are required to buy Google's proprietary 360° cameras. These professional cameras capture true panoramic images 360° coverage and the images are pre-stitched using custiom hardware and software during the capture process (Google). Apple look around also offers similar panorama services like Google street view and the data is collected by their survey vehicles (Apple).

- **Crowdsourcing of street-level imagery:** Although Google and Apple build high-quality panoramas that cover many cities, it is very expensive to collect data with their special survey devices. Mapillary built a global network of contributors in which anyone can join and collect street-level images with their own tools such as smartphones or action cameras. With low device requirements and a large number of contributors around the world, Mapillary offers extensive coverage of its panorama service, reaching even remote or poor areas. Meanwhile, considering the varying quality of the volunteer collected data, Mapillary introduced many machine learning and computer vision algorithms for blurring privacy, image segmentation, 3D reconstruction, and traffic sign recognition (Mapillary).

3 Research Scope

The research questions addressed in this M.Sc. thesis, along with the expected outcomes, are focused on processing existing oblique aerial image data of city-scale in Omnibase to achieve seamless stitching and smooth transitions. The main research question is as follows:

How can seamless oblique image mosaics be created dynamically from aerial photographs to enhance continuous visualization and minimize measurement errors?

The subquestions under this main questions include:

1. To what extent can the stitching achieve a continuous transition between oblique aerial images? And how is continuity measured?

2. How does the area of the intersection or the number of detected tie-points between two images influence the stitching quality?

3. What is the robust method for oblique aerial image stitching when the intersection area is small?

4. How does the stitching process impact the usability of images for measurement tasks?

5. How is stitching quality assessed from both visualization and measurement aspects?

Additional interesting questions are beyond the scope of this MSc thesis, but encouraged to explore include:

1. How can auxiliary data such as 3D point clouds and nadir images be used to enhance the accuracy and visualization for oblique aerial image stitching?
2. How does lens distortion influence stitching result?
3. How to interpolate to get an in-between image when it is needed?
4. How much do deep learning models improve tie-point detection compared to the traditional method like SIFT?
5. Which warping method is better for oblique aerial images considering both time efficiency and stitching quality?
6. How to integrate the image stitching algorithm with the current Omnibase platform?

Due to factors such as time, equipment and data accessibility, this research would **NOT** include the following questions:

1. How to design user-friendly tools for visualization and measurement on stitched oblique aerial images with Omnibase?
2. How to use generative AI to create in-between images?
3. How to generate large-extensive aerial images that cover a whole city?
4. How to apply oblique aerial image stitching techniques on global scale dataset like Google Earth?

4 Methodology

Figure 6 shows the working flow and main steps of this thesis project, including image stitching implementation and the assessment of the result. The general process of stitching images consists of two main necessary steps, matching with tie points and morphing. And interpolation is an optional step that may be applied when the stitching generated directly by morphing the existing images is not ideal. The evaluation of the stitching results will be done based on visual effects, geometric alignment accuracy, and algorithm efficiency in time and memory.

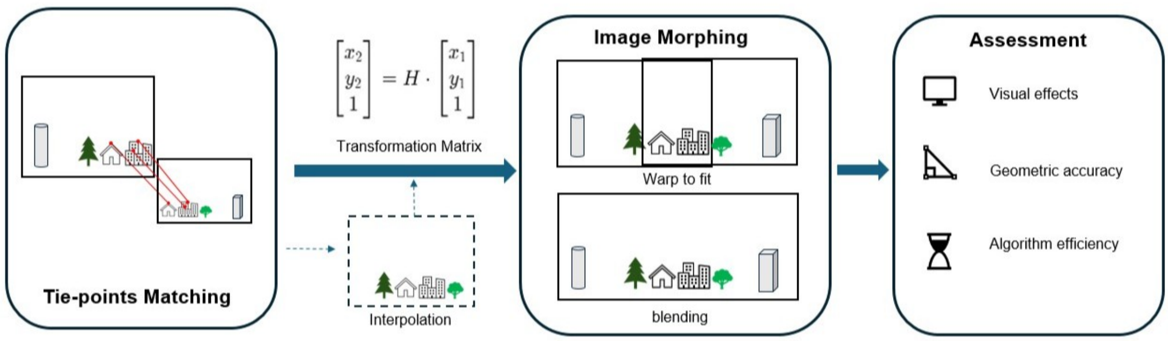

Figure 6: Working flow of this MSc thesis project

4.1 Tie-points matching

Geodelta has already built the database for aerial images of Utrecht with matched tie points by applying a deep neural network, LightGlue (Lindenberger et al. (2023)). LightGlue matches local sparse features with high efficiency and robustness in accuracy. LightGlue performs best for adjacent images with the same view direction although it can also find a few corresponding points between those images that have little intersection areas or with different view directions. The thesis project will focus on the adjacent images with the same view direction for image stitching. The two images on the left of figure 7 shows the overview of the matching result, there are 588 pairs of tie-points in total. The two images on the right of figure 7 show some details of the matching results of the two images on the right which are mostly correct but still have some errors. In practice, the key points detected in more than two images will be selected as reliable tie points. Most of the matching results for adjacent image pairs have numbers of tie-points between 200 and 600, which is sufficient for building the warping transformation.

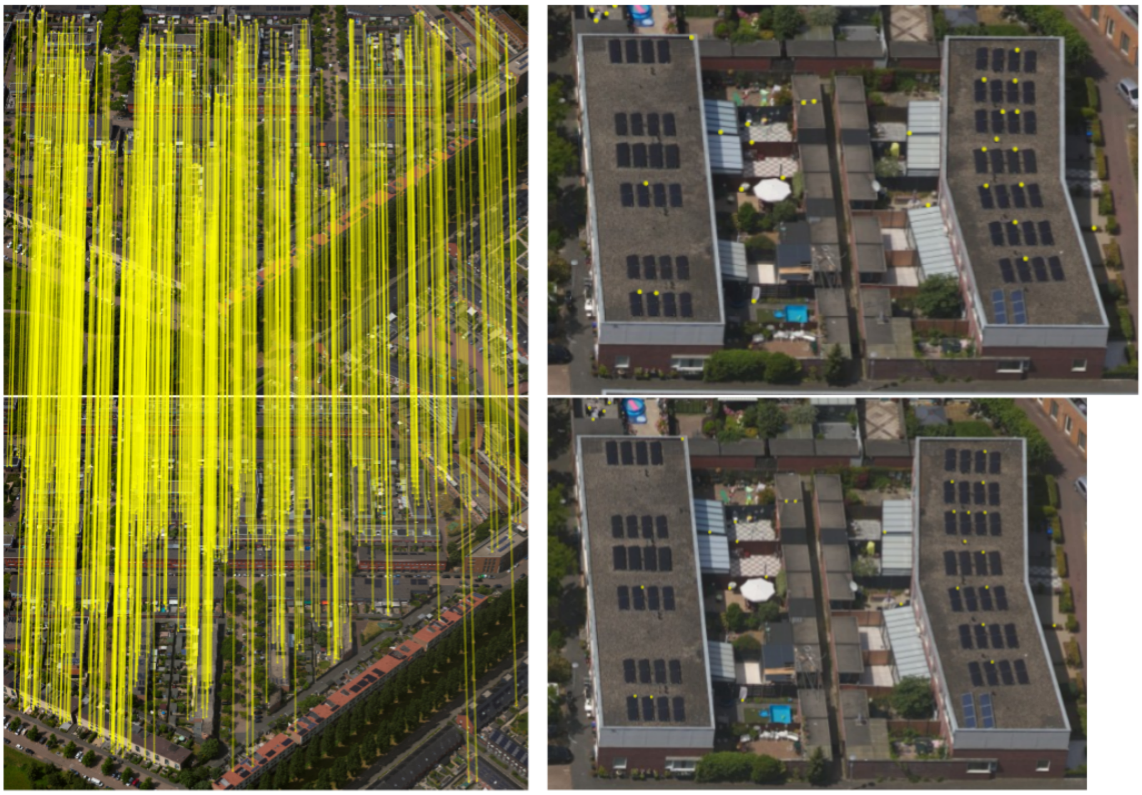

Figure 7: Tie-points matching results for two adjacent oblique aerial images with the same view direction, source: Phoxy

4.2 Morphing Implementation

Image distortion in aerial oblique images is mainly caused by perspective changes due to camera motion. Figure 8 illustrates the perspective distortion resulting from the capture of oblique aerial images. The lower side of the image, which is closer to the camera, appears to be larger and more stretched than the upper side. When the camera moves, the upper side of the previous image will be placed in the lower side in the next. The distortion is usually linear and global. So, the global projective 2D transformation can be useful for most cases.

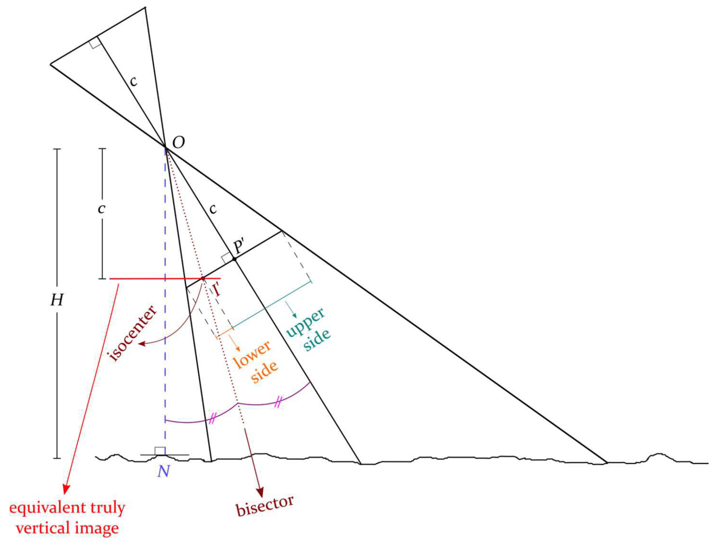

Figure 8: Perspective distortion caused by oblique view (Verykokou and Ioannidis (2024))

Projective transformation, also known as homography, is a type of 2D transformation that maintains perspective relationships between points. The straight lines remain, but the angles, distance and parallel relationship are not preserved. The perspective transformation can map any quadrilateral to another quadrilateral, making it very flexible for warping and alignment. Thus, perspective transformation is widely used in image rectification, stitching, and projection. A projective transformation can be represented mathematically using a **3×3 homography matrix**:

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$$

The matrix H is defined up to scale (i.e., the matrix is normalized such that $h_{33} = 1$ or another value), there are **8 independent unknowns** to solve for: $h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32}$.

The relationship between a point (x, y) in the source image and the transformed point (x', y') in the target image is given by:

$$\begin{bmatrix} x' \\ y' \\ w \end{bmatrix} = H \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Where:

$$x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}}$$

$$y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}$$

Cross-multiplying to eliminate the denominators gives two linear equations for each correspondence:

$$h_{11}x + h_{12}y + h_{13} - h_{31}x'x - h_{32}x'y - h_{33}x' = 0$$

$$h_{21}x + h_{22}y + h_{23} - h_{31}y'x - h_{32}y'y - h_{33}y' = 0$$

Thus, each pair of points contributes 2 equations to the linear system. Since there are 8 unknowns in H, at least 4 pairs of points are needed to solve them. The system will be solved with the singular value decomposition **(SVD)**.
The intersected part of two images can be warped with the solved perspective transformation matrix. Since the oblique aerial images in Omnibase are usually taken under similar lighting conditions, there are not supposed to be abrupt changes in colors between two adjacent images. Linear image blending would be chosen as a simple and effective method for the last step of morphing.

4.3 Interpolation

Although most of the oblique aerial images have sufficient overlap areas for stitching, there is still the possibility that the gap between two adjacent images is so large that it can cause ghosting and blurring if morphing directly. To solve this potential problem, we can perform the interpolation to create the virtual in-between images. There are two possible methods for interpolation demonstrated in Figure 9 and Figure 10. After interpolation, morphing will be applied to continuously switch between two adjacent images.

- **2D interpolation:** Figure 9 shows the 2D interpolation method that used the tie points in the picture as a reference to generate an image in between. As how one point moves from the one image to the next can be tracked and the precise positions where those images were taken are known.It can be linearly interpolated to obtain the position between the two images.

Figure 9: The in-between image(b) is generated by interpolating the left image(a) and the right image(c), using the red tie-points.

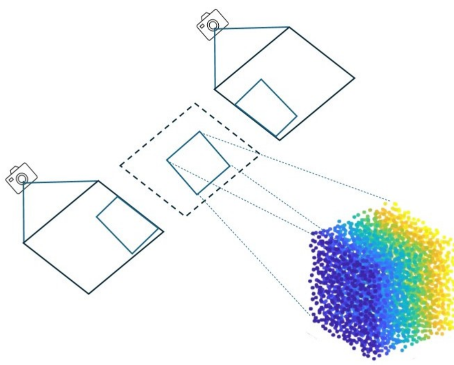

Figure 10: The in-between image is generated by projecting the 3D point cloud cube into the camera plane.

- **3D interpolation:** Figure 10 illustrated using the 3D locations of the points in point cloud that the tie-points are corresponding to. The location of the in-between virtual image frame can be linearly interpolated. The points in point cloud can be projected onto the virtual image frame, and afterwards the in-between image can be used for morphing.

4.4 Assessment

- **Visual effects:** Achieving good visual effects is one of the most important goals of this project, as well as the Omnibase platform. However, visual effects are very subjective and difficult to quantify. Generally, to achieve good visual effects, this project aims to minimize seams, duplicate objects, and distortions in the stitched images. Assessments of visual effects will be presented directly by screenshots of the image details.

- **Geometric accuracy:** The geometric accuracy of the stitched image can be measured from two aspects. First, the alignment accuracy can be measured by the root mean square error (RMSE) between the transformed coordinates of the tie points and the coordinates of their corresponding points on the target image to stitch. See Figure 11 for the visualization of the errors, and the RMSE between the red and blue points in Figure 11 is 15.82 pixels per point. The reprojection errors between the warped tie-points and their aligning Ground Control Points can also be calculated as a geometric accuracy reference. Second, the measurement accuracy will be measured by the measurement result of distances, areas, and angles.

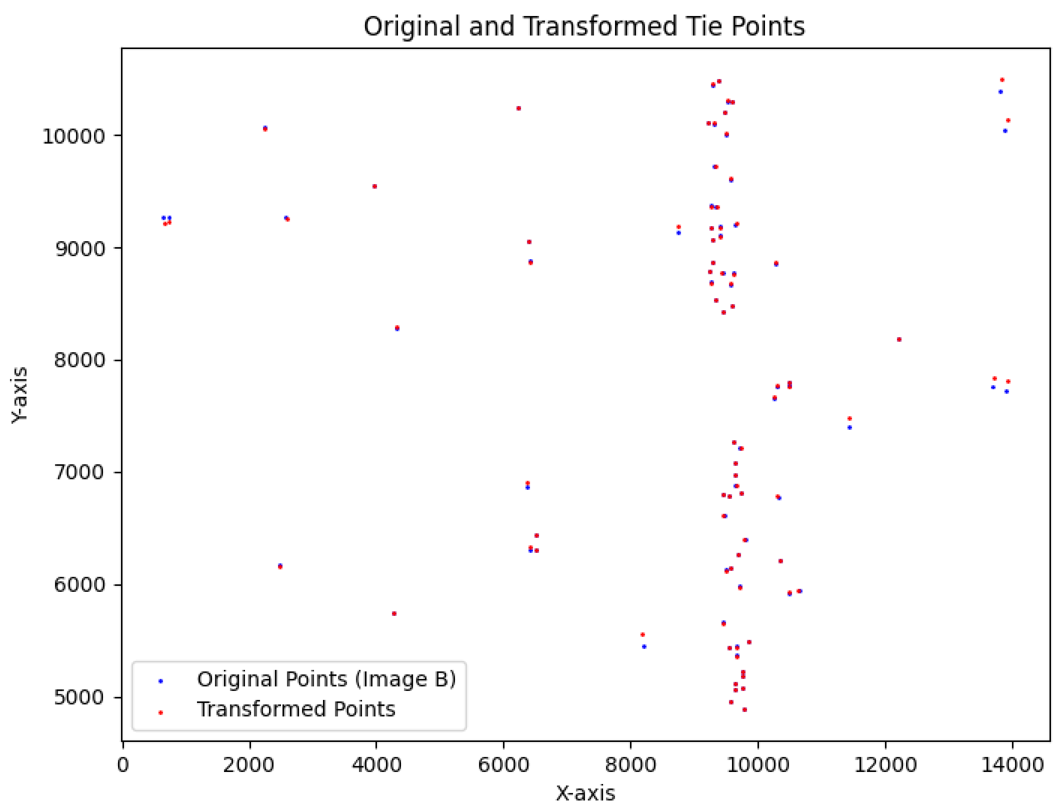

Figure 11: Scatter of transformed tie points and the corresponding points in the target image to stitch (Image B)

- **Algorithm efficiency:** Time efficiency of the stitching algorithm will influence the continuity of image transition process. Both the theoretical time complexity of the algorithm and the practical time cost of the transition between two images will be considered to evaluate the algorithm. Also, since the algorithm will process images with large file sizes and calculate complex matrix, memory usage will also be included in the evaluation of the efficiency of the algorithm.

5 Time planning

Figure 12 shows the proposed timeline for this MSc thesis project. This planning is based on the graduate schedule of the Geomatics program in TUDelft, the content of this project, and the author's personal learning progress and ability.

The tasks of this project are divided into three parts, learning new knowledge, writing code to realize the function, and finally presenting it both in paper and in presentation. The blue blocks in P3 include the implementation for the main questions described in Section 3 and the Optimize block in P4 could include some of the additional questions. The required deliverables are highlighted with black outlines.

Taking into account possible adjustments or unexpected situations in the project process, this timeline serves only as a guide.

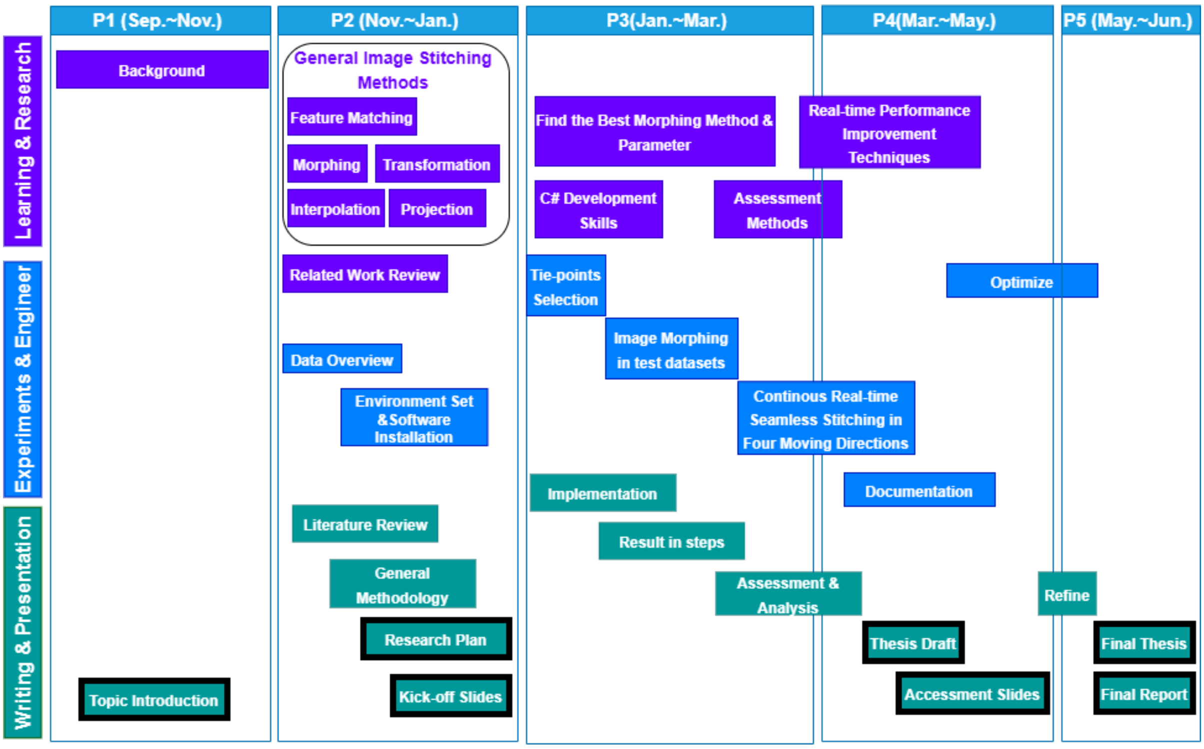

Figure 12: Gantt chart for time planning

6 Tools and datasets used

6.1 Softwares and Platforms

- Coding language and platform: **C#** will mainly be used to achieve fast real-time computation and integrate with Geodelta's development environment. **Python** is also used for some quick algorithm experiments and tests.

- Data View: **Omnibase** is a released platform developed by Geodelta that integrates visualization for nadir and oblique aerial images, panoramas, and point clouds. **Phoxy** is a platform now under development by Geodelta to show the tie points between two images.Both platforms are used in this thesis project for viewing data and quick analysis.

- Data Management: **SQLite** is used for storing the camera parameters and information about tie points. In this program, accessing the information from the database and processing them is required. Writing and storing information to the database may also be required for storing some calculated parameters to get better real-time performance.

6.2 Datasets

- **Oblique Aerial Images:** The current dataset of this thesis project covers the municipality of Utrecht (see Figure 13). The images were taken in 2024 with 45 degrees down to the earth's surface by a survey company, Kavel 10. The exact positions of the images are recorded with the GNSS system. Some images that largely consist of water are labeled as unsuitable mapping due to the difficulty of finding the connection ground points for accurate adjustment. A similar dataset taken in Rotterdam may also be used in this thesis project.

a

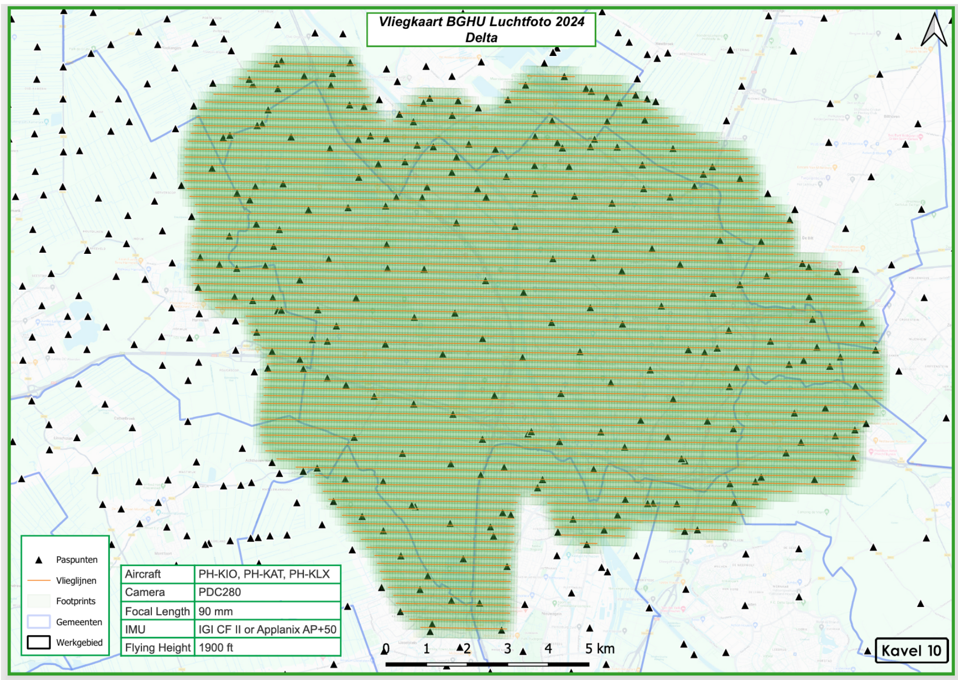

Figure 13: Flight map of oblique aerial images taken in Utrecht

- **Phoxy Database:** As mentioned in the tool part, Phoxy is still in development for matching images. Although still having many bugs for its user interface, its database has already built with abundant information about those oblique aerial images including parameters of the camera and raw images, record of matched image pairs, information about the features. Figure 14 shows the simple version of this database structure that contains the fields that will be used in this thesis project.

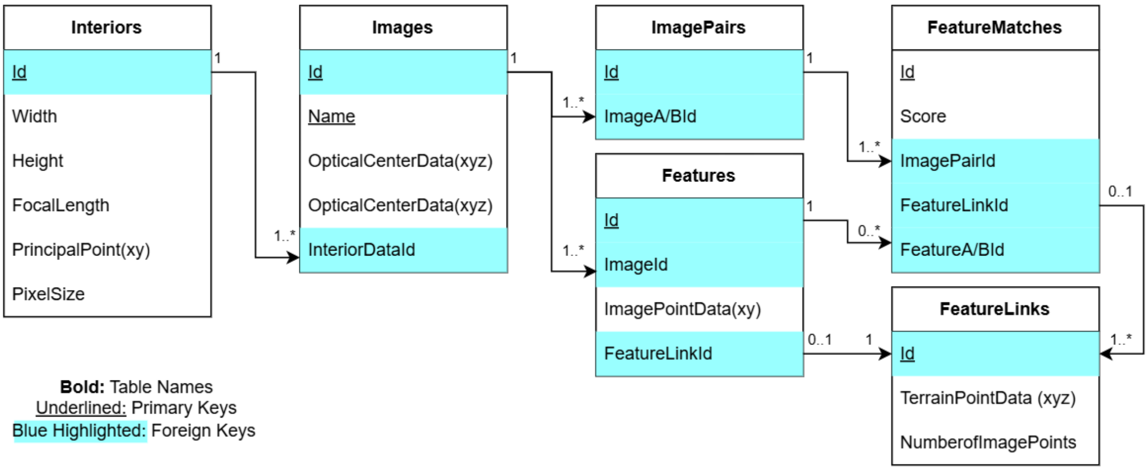

Figure 14: Database Structure for Phoxy

References

Adobe. What is a cross dissolve transition? https://www.adobe.com/creativecloud/video/discover/video-cross-dissolve.html. Accessed: 2025-01-02.

Apple. Apple maps image collection. https://maps.apple.com/imagecollection/us. Accessed: 2025-01-06.

D. Armstrong, E. Charlesworth, A. J. Alderson, and D. B. Elliott. Is there a link between dizziness and vision? a systematic review. *Ophthalmic and Physiological Optics*, 36(4):477–486, 2016.

H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.

Bing Maps Team. Get bird's eye views in your next great maps app. https://blogs.bing.com/maps/2022-07/Get-Bird-s-Eye-Views-in-your-next-great-maps-app, 2022. Accessed: 2025-01-02.

M. Brown and D. G. Lowe. Automatic panoramic image stitching using invariant features. *International journal of computer vision*, 74:59–73, 2007.

P. J. Burt and E. H. Adelson. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics (TOG)*, 2(4):217–236, 1983.

M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*, pages 778–792. Springer, 2010.

N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

A. A. Efros. Image morphing. https://people.eecs.berkeley.edu/~job/Classes/CS184/Fall-2016-Slides/efros-morphing-cs184.pdf, 2016. Lecture slides, accessed: 2025-01-02.

M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

Geodelta. OmniBase Software. https://geodelta.com/software/omnibase, 2024. Accessed: 2024-11-27.

GeoXphere. Xmap oblique viewer. https://xmap.geoxphere.com/apps/oblique. Accessed: 2025-01-02.

Google. Google street view: Contribute. https://www.google.com/streetview/contribute/. Accessed: 2025-01-06.

J. Höhle. Oblique aerial images and their use in cultural heritage documentation. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40:349–354, 2013.

A. Laraqui, A. Baataoui, A. Saaidi, A. Jarrar, M. Masrar, and K. Satori. Image mosaicing using voronoi diagram. *Multimedia Tools and Applications*, 76:8803–8829, 2017.

D.-T. Lee and B. J. Schachter. Two algorithms for constructing a delaunay triangulation. *International Journal of Computer & Information Sciences*, 9(3):219–242, 1980.

A. Levin, A. Zomet, S. Peleg, and Y. Weiss. Seamless image stitching in the gradient domain. In *Computer Vision-ECCV 2004: 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part IV 8*, pages 377–389. Springer, 2004.

P. Lindenberger, P.-E. Sarlin, and M. Pollefeys. Lightglue: Local feature matching at light speed. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17627–17638, 2023.

D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004.

W. LYU, Z. ZHOU, L. CHEN, and Y. ZHOU. A survey on image and video stitching. *Virtual Reality amp; Intelligent Hardware*, 1(1):55–83, 2019. doi: 10.3724/sp.j.2096-5796.2018.0008.

Mapillary. Mapillary technology. `https://www.mapillary.com/technology?locale=en_US`. Accessed: 2025-01-06.

N. T. Pham, S. Park, and C.-S. Park. Fast and efficient method for large-scale aerial image stitching. *IEEE Access*, 9:127852–127865, 2021.

O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9*, pages 430–443. Springer, 2006.

E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011.

P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020.

C. V. Stewart. Robust parameter estimation in computer vision. *SIAM review*, 41(3):513–537, 1999.

R. Szeliski et al. Image alignment and stitching: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 2(1):1–104, 2007.

M. Tang, X. Mei, Y. Li, C. Chen, X. Liu, and H. Lu. Application of oblique photogrammetry technique in geological hazard identification and decision management. *Earthquake Research Advances*, 4(3):100269, 2024.

M. Tyszkiewicz, P. Fua, and E. Trulls. Disk: Learning local features with policy gradient. *Advances in Neural Information Processing Systems*, 33:14254–14265, 2020.

U.S. Geological Survey. Oblique aerial photography viewer. `https://www.usgs.gov/tools/oblique-aerial-photography-viewer`, 2018. Accessed: 2025-01-02.

S. Verykokou and C. Ioannidis. Oblique aerial images: Geometric principles, relationships and definitions. *Encyclopedia*, 4(1):234–255, 2024.

Z. Wang and Z. Yang. Review on image-stitching techniques. *Multimedia Systems*, 26(4):413–430, 2020.

Z. Zhang, L. Cheng, X. Feng, X. Wang, G. Zhai, K. Tang, and S. Sun. Application of oblique photography technology in airport site selection. In *Civil Engineering and Disaster Prevention*, pages 413–419. CRC Press, 2023.