# Image matching and registration

## Introduction

There are multiple possible directions the project can involve in. Since other tasks were already taken, I decided to go with “image matching” which is related to template matching, one technique of object detection. It relies heavily on detecting points of interest or “edges” in images.

In the context of the project, image and template matching could possibly be used to construct a map from multiple 2D flat images or from the drone footage, we’ll need to apply edge detection ( point of interest detection) on multiple consecutive images that we got from a video of the sea in order to construct a final map.

## Key Points and Feature Detection in Computer Vision

Identifying key points or features in images is essential for many tasks in computer vision, like matching, aligning, and transforming images. Key points are distinct and unique parts of an image that can be detected and used to align or stitch multiple images together.

## Harris Corner Detection Algorithm

The Harris corner detection algorithm is a common method for finding corners in images, which are important features. This algorithm uses a structure matrix to summarize gradient information around a pixel. By looking at the eigenvalues of this matrix, it can detect corners where there are significant changes in gradient directions. This helps in finding unique points in an image that remain consistent under different transformations, making it easier to match corresponding points in different images.

## Affine and Projective Transformations

Two basic types of image transformations used in computer vision are affine and projective transformations:

* **Affine Transformations**: These transformations preserve parallel lines and include operations like scaling, rotation, translation, and shearing. They are represented by a 2x3 matrix and are crucial for applications that require seamless blending of images.
* **Projective Transformations**: These transformations can handle more complex distortions, such as warping, and are represented by a 3x3 matrix. They are especially useful in creating panoramas by warping images back to a single plane, allowing multiple images to be combined into one cohesive image.

These transformation will represent the most common and possible modifications that a camera picture can apply on any given scene.

## RANSAC Algorithm

The RANSAC (Random Sample Consensus) algorithm is a method for dealing with outliers in data when performing image transformations. RANSAC iteratively selects random subsets of data points to estimate a model and identifies inliers that fit the model well. This way, RANSAC can effectively manage noisy data and find the best fit for image transformations, making it a valuable tool in scenarios where data may contain significant outliers.

## Encoding and Solving Linear Systems

Encoding and solving linear systems of equations using matrices is a key technique in finding affine transformations between matched points. This involves forming matrices from the known values and solving for the unknowns using methods like least squares minimization. This approach ensures accurate transformations even in overdetermined systems, where there are more equations than unknowns.

## Homography and Image Stitching

Homography is a type of projective transformation used to align images of the same planar surface. It involves finding a matrix that projects points from one image to another while maintaining the geometric relationship between them. Homography is crucial for applications like panorama stitching, where multiple images need to be combined into a single, seamless image.

## Feature Descriptors: HOG and SIFT

Feature descriptors are used to describe and match key points between images:

* **Histogram of Oriented Gradients (HOG)**: HOG descriptors help reduce dependence on lighting conditions by normalizing gradient magnitudes and focusing on gradient directions. This method provides some invariance to lighting changes but is not rotation invariant.
* **Scale-Invariant Feature Transform (SIFT)**: SIFT addresses the limitations of HOG by providing both rotation and scale invariance. SIFT detects key points in images and calculates descriptors that are robust to changes in scale, rotation, and illumination. This makes SIFT particularly effective for matching features across different images, even under significant transformations.

## Advanced Image Stitching Techniques (Possible improvements)

To overcome the limitations of straightforward projective transformations, advanced techniques are used:

* **Cylindrical Projection**: Modeling the world as a cylinder for image stitching reduces distortion and allows for the creation of wide-field-of-view images. This involves transforming image points into cylindrical coordinates and then projecting them back onto a flat plane.
* **Alpha Blending and Multiband Blending**: These techniques enhance the visual quality of stitched images by smoothly blending overlapping regions, reducing noticeable seams and discontinuities.

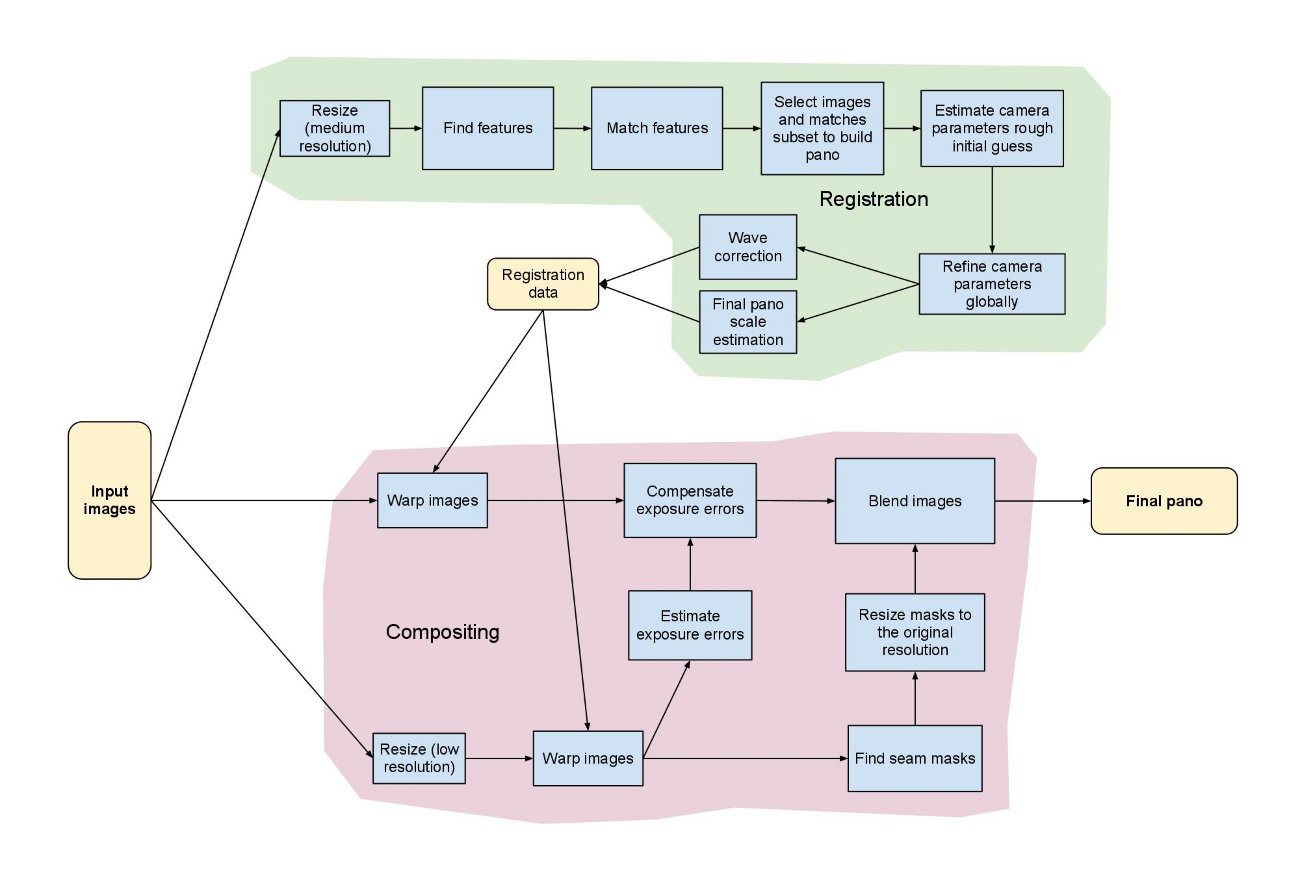
## Transition to Neural Networks

While traditional methods like SIFT were groundbreaking in feature detection and matching, modern computer vision applications increasingly rely on neural networks. Neural networks offer superior performance and flexibility in handling complex image transformations and feature extraction tasks, making them the preferred choice in contemporary computer vision systems.

## Implementation

I’ll build a working program first with data that I got elsewhere, then I’ll work on data of the project if there is any given.

OpenCV has a built in stitching method that is usually used to combine multiple images together into one big panoramic picture. We can use this feature without necessarily understanding the exact details of the stitching pipeline, but here is a diagram showing the steps:



Stitching pipeline from OpenCV documentation

We can start with 2 random frames to test out the result of this feature before applying it to a complete video sample.

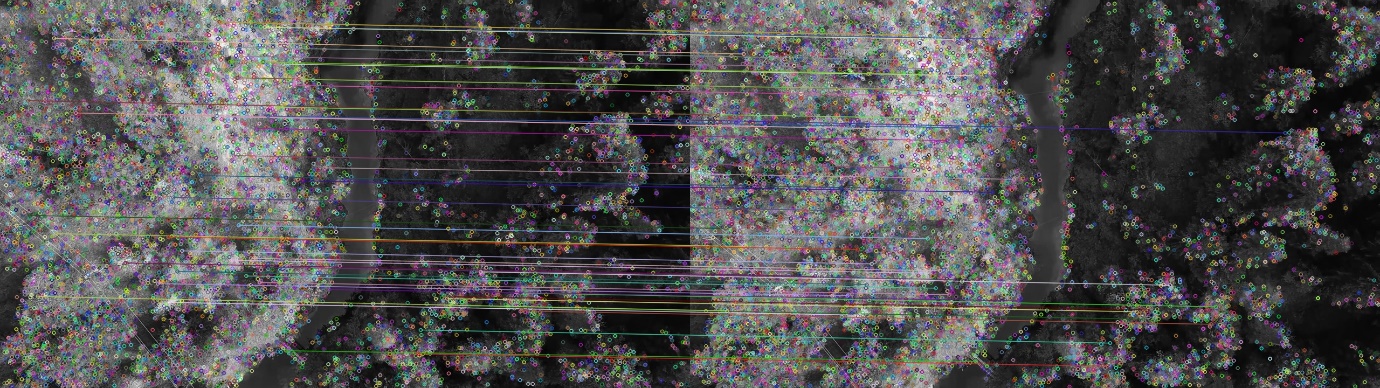
By using the 1st and 4th pictures of the sample video, we get the following result:   
 

Blended picture of 1st and 4th frames

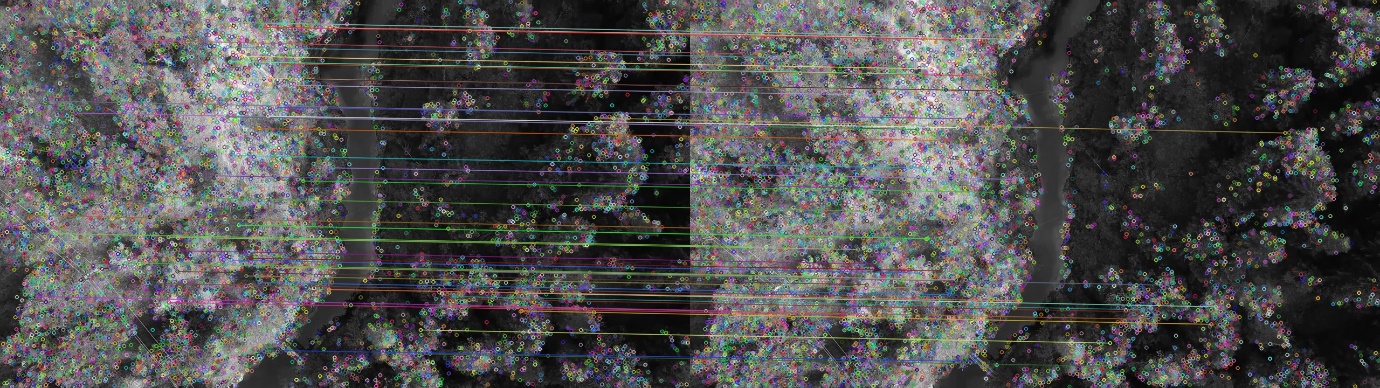
## Keypoints extraction



## Image matching



Brute-Force matching



FLANN matching

Having mostly (or entirely) horizontal lines with no intersections could be a sign that the matching is correct, meaning the keypoints represent the same feature in both pictures.

The rest of the results are in the project repository: <https://github.com/fayssalElAnsari/Posidonea_Oceanica>

## References and sources used

<https://github.com/mikeroyal/Photogrammetry-Guide>

<https://docs.opencv.org/4.x/d1/d46/group__stitching.html>

<https://en.wikipedia.org/wiki/Template_matching>

<https://www.youtube.com/watch?v=taty6lPVcmA&list=PLjMXczUzEYcHvw5YYSU92WrY8IwhTuq7p&t=3026s>

<https://chat.openai.com/>