# Ex1_NewSpace – Parts 1 - 4 Summary

## Colab link:
## https://colab.research.google.com/drive/1YOgURT71km9IIrDwBpmQRx5mSEwVudY-?usp=sharing

## Students:
Abedallah Zoabi 211407424

Saleh Sawaed 214074676

Natan Partouche 342533106 (in meluaim)

## Part 1: Star Detection and Matching Algorithm Design

### Algorithm: Star Detection and Matching

***Input:***

Two grayscale images containing stars.

***Output:***

A list of matched star pairs between the images.

---

### 1. Star Detection

- Convert image to grayscale and apply Gaussian blur.
- Apply binary thresholding to isolate bright spots.
- Detect contours and compute centroids using image moments.
- Output: list of star coordinates `(x, y)`.

---

### 2. Triangle Descriptor Construction

- For each image, generate all triangles from 3-star combinations.
- For each triangle, compute side lengths and derive a scale- and rotation-invariant descriptor based on side ratios.

---

### 3. Triangle Matching

- Compare triangle descriptors between the two images.
- If descriptors are similar (within a threshold), consider them a match.

---

### 4. Voting-Based Star Matching

- For each matched triangle pair, vote for matching star pairs.
- Select the best one-to-one matches based on highest votes.

## Part 2: Star Detection with Blob Analysis

In Part 2, a more precise method was implemented to detect stars and return detailed information for each detection. The algorithm uses **Laplacian of Gaussian (LoG)** blob detection to identify star-like features in grayscale images.

Each detected star is described as a **4-tuple**:
**(x, y, r, b)** where:

- `x`, `y` — star coordinates in the image,
- `r` — estimated radius of the star (from the blob size),
- `b` — average brightness around the star.

To improve accuracy, stars are filtered by **brightness and radius**, and the results are visualized and saved in a text log. This method is robust to noise and captures star properties for later matching.

## Part 3: Star Matching with Triangle Descriptors

In Part 3, we implemented a geometric matching algorithm to identify corresponding stars between two images. Each image's detected stars are used to construct triangles, and each triangle is described using **ratios of side lengths**, which are invariant to **rotation and scale**.

The algorithm performs the following steps:

- Build triangle descriptors from all star triplets in both images.
- Compare descriptors between images and find similar triangles.
- Use a **voting system** to identify the most likely star-to-star matches based on how often they appear in matched triangles.

This method is **robust to shifts, rotation, and scale differences** between images.

## Part 4: Large-Scale Matching and Evaluation

We evaluated the triangle-based star matching algorithm using two images. The output shows a **successful match of 30 stars**, meaning the same star was correctly identified in both images.

The visualization (see figure) displays **lines connecting matched stars**, demonstrating the algorithm's ability to handle **translation and rotation** between the images. Each line connects the position of a star in Image 1 to its corresponding position in Image 2.

This proves the geometric triangle-based method is **accurate and robust**, especially in real-world noisy data, and confirms that using triangle descriptors yields **consistent star matches across images**.