# Mini-project: A visual odometry pipeline!

**Juan Bermeo**[1] **and Tom**[2]

[1]**juandiego.bermeoortiz@uzh.ch, LEGI: 20-754-951**
[2]**Address of second author**

## ABSTRACT

Please provide an abstract of no more than 300 words. Your abstract should explain the main contributions of your article, and should not contain any material that is not included in the main text.

Keywords:    Keyword1, Keyword2, Keyword3

## MILESTONES

1. **Intialization Module**

   (a) Define/Specify the logic or argument to pick specific initialization frames for each dataset.

   (b) Identify features in both frames with one of the following techniques:

      - Corner detection: Harris, Shi-Tomasi, FastCorner
      - Sift Features
      - **Bonus**: Lift

   (c) Match the keypoints between frames to find keypoint correspondences between different views (between frames or between multiple cameras).

      - Choose a type of distance/similarity metric
      - Choose to do matching with either pairwise comparison or KLT

   (d) Estimate the point cloud of the landmarks.

   (e) Estimate relative pose of the next frame by calculating the fundamental matrix.

      - Use 8 point ransac (estimateFundamentalMatrix)
      - **Bonus**: 5 point ransac
      - **Bonus**: 1 point ransac + 5 point ransac

   (f) Refine the estimated pose with Bundle Correction.

2. **Continous VO Module**

   (a) Create functions that iterates over all subsequent states, always using the keypoints in frame $i$ and compare/match them to those of frame $i+1$ to estimate the pose of frame $i+1$. (practically following the same pipeline as initialization).

   (b) Persist the the estimated poses at each step.

   (c) Choose error metrics and create function that calculates global and local error in pose estimation:

      - Figure out how to obtain the ground truth of poses at the local and global level
      - Create function that computes the metric for both