In-Pipe Bot

Abstract: A bot that moves inside the pipe to inspect Cracks, Block and Rust and plots/tags the respective point of issue in a Live Map of the Bot.

Approach

1) Record inner-view of the pipe using a camera that rotates on the axis of the bot.
   - **Method1:** Record a video and convert it into panaromic image using keypoint recognition and image stiching.
   - **Method2:** Take photographs at a certain delays such that all parts are covered avoiding over-lapping in images.
2) Check the images resulting from previous steps for cracks.
   - **Method1:** Detecting edges and contours of the cracks
   - **Method2:** CNN classifier
3) **Mechanics**: A manually-pushed capsule with a probe at front that will turn into bends of the pipe.

Creating Panaromic Images

1) Extract frames from the video
2) Sample the video frames to reduce number of stiches required
3) Detect keypoints and descriptors
4) Detect a set of matching points that is present in both images (overlapping area)
5) Apply the RANSAC method to improve the matching process detection
6) Apply perspective transformation on one image using the other image as a reference frame
7) Stitch images together
8) Step 3 to Step 7 are repeated over consecutive frames.

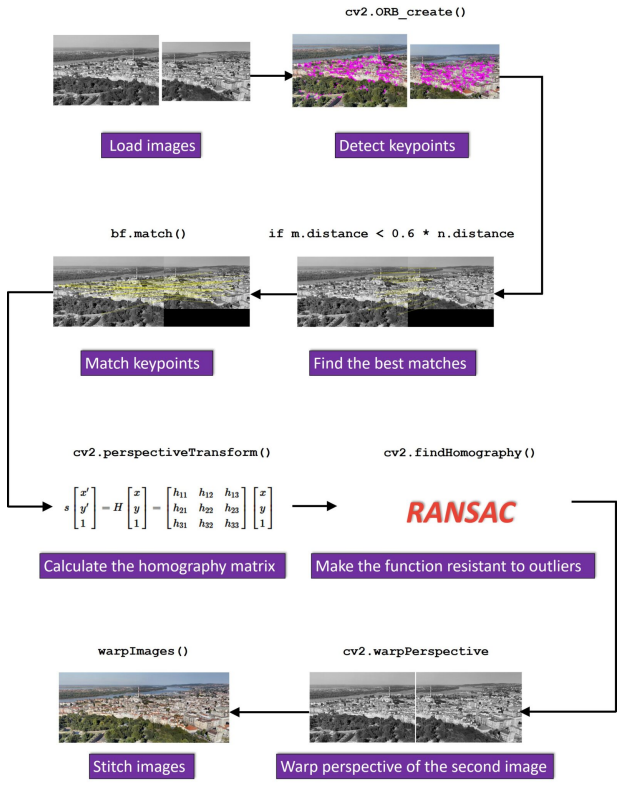

FIGURE 1. Stiching two Images (frames)

Model 1

This model identifies a crack present in an image using built-in OpenCV functions.
Follwing is the procedure followed to obtain the crack in the image.

1. PreProcessing :

Conversion of Image from BGR to Gray scale using `cv2.cvtcolor()`
Denoising the Image using `cv2.fastNlMeansDenoising()` and applying Blur on the resultant Image using `cv2.GaussianBlur()`

`cv2.fastNlMeansDenoising()`
Based on Non-Local Means Algorithm to DeNoise an gray scale image.
`cv2.GaussianBlur()`
Convolving the image with Gaussian Kernel with specified size and Standard Deviations in the X and Y directions.

2 . Edge Detection :

Canny edge detection is performed using `cv2.Canny()`

Algorithm of Canny Edge Detection
Following are few of the other ED :
1. `cv2.Laplacian()`
2. `cv2.Sober()`
3. `cv2.Scharr()`
4. `cv2.HoughLines()`

Model 2

An Binary Classification using CNNs with 40000 Images (227 x 227 x 3) Dataset. Which contains 20000 Images with crack and 20000 Images without crack.
The architecture of CNN contains 3 Convolutional Layers and 3 Fully Connected Layers, Where Adam is used as Optimizer and Cross Entropy Loss is used as Loss Function.

References:

1) Image Stiching: LINK