Localization of Unmanned Aerial Vehicles in Corridor Environments using Deep Learning

Ram Prasad Padhy, Shahzad Ahmad, Sachin Verma, Sambit Bakshi, and Pankaj Kumar Sa*

Department of CSE, NIT Rourkela, India

25th International Conference on Pattern Recognition (ICPR), 2020. Submission# 2600

Abstract

We propose a monocular vision assisted localization algorithm, that will help a UAV navigate safely in indoor corridor environments. Always, the aim is to navigate the UAV through a corridor in the forward direction by keeping it at the center with no orientation either to the left or right side. The algorithm makes use of the RGB image, captured from the UAV front camera, and passes it through a trained Deep Neural Network (DNN) to predict the position of the UAV as either on the left or center or right side of the corridor. Depending upon the divergence of the UAV with respect to an imaginary central line, known as the central bisector line (CBL) of the corridor, a suitable command is generated to bring the UAV to the center. When the UAV is at the center of the corridor, a new image is passed through another trained DNN to predict the orientation of the UAV with respect to the CBL of the corridor. If the UAV is either left or right tilted, an appropriate command is generated to rectify the orientation. We also propose a new corridor dataset, named UAVCorV1, which contains images as captured by the UAV front camera when the UAV is at all possible locations of a variety of corridors. An exhaustive set of experiments in different corridors reveal the efficacy of the proposed algorithm.

Introduction

- The only sensor used is a forward facing static camera due to its light weight, low power consumption.
- Although more challenging, monocular vision processing is more efficient than stereo vision for real-time navigation tasks
- A DNN based model is proposed for safe localization a UAV in indoor corridor environments.
- Our goal is to autonomously navigate a UAV in indoor corridors without any collision either with the side walls or with the front wall.
- Unlike previous methods, where the DNN models were designed to predict flight commands directly, our proposed method makes use of an important characteristic of a corridor, the Central Bisector Line (CBL), to generate commands.

Contributions

1. We propose a method, which uses two different DNN models for safe localization of a UAV in corridor environments. The first DNN is responsible for predicting the deviation of the UAV in terms of translation whereas the second DNN predicts the orientation.
2. The algorithm generates necessary control commands to keep the UAV along the CBL of the corridor and continuously monitors the position to rectify any deviation.
3. We propose a new corridor dataset, UAVCorV1 [1], which contains images as captured from the UAV front camera from different positions of a number of corridors, having varying dimensions and intensity exposure.
4. The dataset is trained on several state-of-the-art DNN models for predicting both translational and rotational deviation. The best model among them based on few accuracy metrics is chosen for real-world navigation flights.

Central Bisector Line

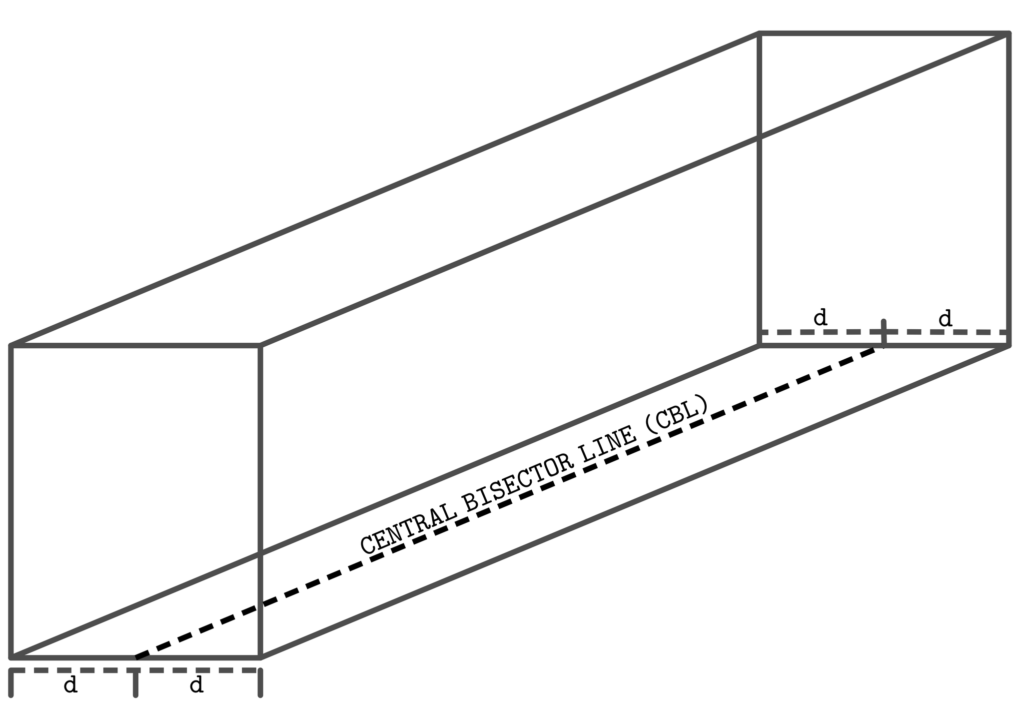


Figure 1: Central bisector line of a corridor. It's an imaginary line used as reference.

Dataset Creation

Figure 2: Images as captured by the UAV front camera from 9 different possible alignments over a horizontal line perpendicular to the CBL.

- At a certain place inside the corridor, three different locations on a horizontal line, which is perpendicular to the CBL; the center and two extreme sides of the corridor, are selected. At each location on the horizontal line, the images are captured while the UAV is tilted in three different directions: center, left, and right tilt. Hence, we will have 9 different images corresponding to a particular horizontal line.
- For each image, a similar image is captured by placing two markers on the CBL of the corridor. These images are known as bisector images.

CBL creation with Marker Image

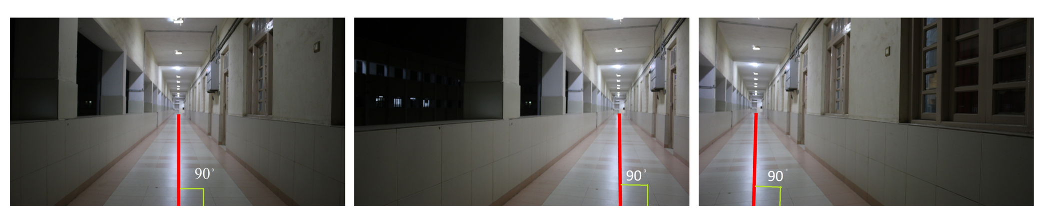
(a) Actual image (b) Image with markers (c) CBL on image plane

Figure 3: Process of obtaining the CBL on the image plane using markers. These are connected by a red colored line, which forms the CBL on the image plane.

Target Data Generation

(a) Left side of CBL (b) Right side of CBL (c) On the CBL

Figure 4: Translational Shift: Three different positions of the UAV over a horizontal line perpendicular to the CBL.



(a) Aligned with the CBL (b) Left tilted (c) Right tilted

Figure 5: Rotational shift: Three different orientations of the UAV, when it is situated on the CBL.

- For each image, the dataset now contains two target values:
  1. Angle of the CBL (for rectifying the translational deviation)
  2. Distance of the CBL (for rectifying the rotational deviation).

UAV Navigation

- The process of UAV navigation is a two-step process occurring consecutively
  1. Rectifying the translational deviation (side-wise variation) for bringing the UAV to center over the CBL
  2. Rectifying the rotational deviation (change in orientation) to align the UAV with the CBL, when the UAV is already at the center

Network Structure

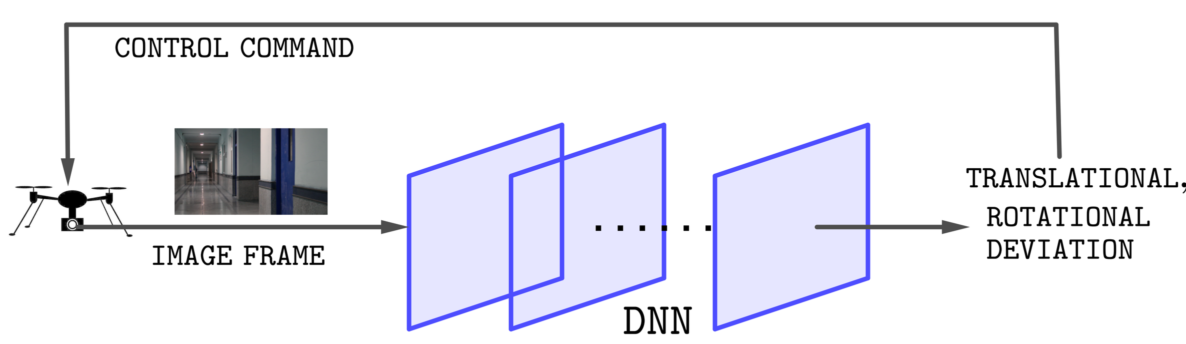


Figure 6: Architectural flow of the proposed DNN based corridor navigation model. Both processes are achieved by processing the images from the UAV front camera through pre-trained DNNs, that predict the deviations.

Table 1: Network structure

Input	Pre-trained Models	Augmented Convolution Layers	Augmented last Fully connected layer	Output layer
320×180	AlexNet	no augmentation of convolution layer	4096×1	1×1
	VGG-16	Conv2d(512, 1024, 1×1) Conv2d(1024, 128, 5×5) Conv2d(128, 16, 1×1)	96×1	
	InceptionV3	**Main:** Conv2d(2048, 1024, 1×1) Conv2d(1024, 512, 2×2) Conv2d(512, 128, 3×3) **Aux:** Conv2d(768, 128, 4×4) Conv2d(128, 32, 2×2)	**Main:** 256×1 **Aux** :640×1	
	ResNet-50	Conv2d(2048, 1024, 1×1) Conv2d(1024, 128, 5×5) Conv2d(128, 8, 1×1)	96×1	
	ResNet-101			
	ResNet-152			
	DenseNet-201	Conv2d(1920, 1024, 1×1) Conv2d(1024, 128, 5×5) Conv2d(128, 16, 1×1)	96×1	
	DenseNet-161	Conv2d(2208, 1024, 1×1) Conv2d(1024, 128, 5×5) Conv2d(128, 16, 1×1)		

Loss Function: Mean Absolute Error

$$\text{MAE}(\hat{Y}, Y) = \frac{1}{n}\sum_{i=1}^{n} |\hat{y}_i - y_i|,$$

where $\hat{Y} = \{\hat{y}_i\}_{i=1}^{n}$ and $Y = \{y_i\}_{i=1}^{n}$ denote the predicted and target values for a mini batch of size n, respectively.

Control Command generation

Algorithm 1: Control command generation

Input: Image From UAV front camera: img
Output: UAV direction: $[pitch, roll, yaw]$

$angle$ = TrainedModelForAngle(img)
if $angle\ out\ of\ bound\ for\ continously\ 1\ second$ **then**
    Land the UAV
if $angle \approx 90^\circ$ **then**
    dist = TrainedModelForDistance(img)
    if $dist \approx 0.5$ **then**
        Actuate UAV in Pitch Forward
    else if $dist < 0.5 - \delta$ **then**
        Actuate UAV in Yaw Left until $dist \approx 0.5$
    else
        Actuate UAV in Yaw Right until $dist \approx 0.5$
else if $angle < 90^\circ - \delta$ **then**
    Actuate UAV in Roll Right until $angle \approx 90^\circ$
else
    Actuate UAV in Roll Left until $angle \approx 90^\circ$
return $[pitch, roll, yaw]$

Evaluation Metrics

$$\text{Mean Squared Error}: \ \text{MSE}(\hat{Y}, Y) = \frac{1}{n}\sum_{i=1}^{n} (\hat{y}_i - y_i)^2,$$

$$\text{Mean Absolute Error}: \ \text{MAE}(\hat{Y}, Y) = \frac{1}{n}\sum_{i=1}^{n} |\hat{y}_i - y_i|,$$

$$\text{Mean Relative Error}: \ \text{MRE}(\hat{Y}, Y) = \frac{1}{n}\sum_{i=1}^{n} \frac{|\hat{y}_i - y_i|}{y_i}.$$

where $\hat{Y} = \{\hat{y}_i\}_{i=1}^{n}$ and $Y = \{y_i\}_{i=1}^{n}$ denote the predicted and target values for a mini batch of size n, respectively.

Experimental Results: Quantitative Comparison

Table 2: Evaluation metrics for the prediction of translational and rotational shift.

Pretrained Model	Translational Deviation			Rotational Deviation		
	MSE	MAE	MRE	MSE	MAE	MRE
AlexNet	0.21997	1.72831	1.39280	5.5677	27.006	54.147
VGG-16	0.47318	2.84597	2.41795	1.1928	4.6213	13.533
InceptionV3	0.11929	1.44906	1.20959	0.0687	3.1364	10.485
ResNet-50	0.11253	1.50321	1.18916	0.0473	2.7485	9.0729
ResNet-101	0.11103	1.46875	1.17946	0.1186	4.2163	14.681
ResNet-152	0.11032	1.41558	1.16529	0.0662	3.4258	10.823
DenseNet-201	0.12383	1.79144	1.42709	0.0828	3.6442	12.142
DenseNet-161	**0.05791**	**1.32693**	**1.08712**	**0.0326**	**2.5060**	**1.5570**

Experimental Results: Qualitative Comparison

TIIR Building	Physics Department	Life Science Department	Computer Science Department

GT: 90.720°	**GT:** 47.055°	**GT:** 90.000°	**GT:** 136.507°
PR: 90.303°	**PR:** 46.707°	**PR:** 91.142°	**PR:** 135.253°

Figure 7: Qualitative performance evaluation of translational deviation for different corridor locations of National Institute of Technology, Rourkela, India. Ground truth and predicted values are given in degree. GT: Ground truth, PR: Predicted.

TIIR Building	Physics Department	Life Science Department	Computer Science Department

GT: 0.488	**GT: 0.805**	**GT: 0.547**	**GT: 0.167**
PR: 0.491	**PR: 0.808**	**PR: 0.528**	**PR: 0.163**

Figure 8: Qualitative performance evaluation of rotational deviation for different corridor locations of National Institute of Technology, Rourkela, India. Ground truth and predicted values are in the range $[0, 1]$. GT: Ground truth, PR: Predicted.

Real World Navigation Experiments

- Parrot A.R.Drone quadcopter is used for validation purpose.
- Image transmission delay through ROS is about 0.21s. Our DNN model prediction takes about 0.08s for measuring both translational and rotational deviation simultaneously. Also, the time required for communicating a control command through ROS is about 0.21s.
- Hence, our algorithm can process at most two frames in one second, which is sufficient to generate safe control commands and navigate without collision.
- It may be noted that apart from our network prediction, different factors, such as control and state estimation affect the actual UAV flight in real-world scenarios.
- We tested our algorithm across 50 trials in 10 different corridors, out of which 43 trails were found to be successful.

References

[1] ``Corridor Dataset for UAV Navigation,'' http://www.nitrkl.ac.in/docs/CS/Database/Windows/NitrUAVCorridorV1.