Hierarchical Structured Learning for Indoor Autonomous Navigation of Quadcopter

Vishakh Duggal
RRC, KCIS, IIIT
Hyderabad
vishakh.duggal@research.iiit.ac.in

Kumar Bipin
linux.kbp@gmail.com

Utsav Shah
RRC, KCIS, IIIT
Hyderabad
shah.utsav@research.iiit.ac.in

K. Madhava Krishna
RRC, KCIS, IIIT
Hyderabad
mkrishna@iiit.ac.in

ABSTRACT

Autonomous navigation of generic monocular quadcopter in the indoor environment requires sophisticated approaches for perception, planning and control. This paper presents a system which enables a miniature quadcopter with a frontal monocular camera to autonomously navigate and explore the unknown indoor environment. Initially, the system estimates dense depth map of the environment from a single video frame using our proposed novel supervised Hierarchical Structured Learning (HSL) technique, which yields both high accuracy levels and better generalization. The proposed HSL approach discretizes the overall depth range into multiple sets. It structures these sets hierarchically and recursively through partitioning the set of classes into two subsets with subsets representing apportioned depth range of the parent set, forming a binary tree. The binary classification method is applied to each internal node of binary tree separately using Support Vector Machine (SVM). Whereas, the depth estimation of each pixel of the image starts from the root node in *top-down* approach, classifying repetitively till it reaches any of the leaf node representing its estimated depth. The generated depth map is provided as an input to Convolutional Neural Network (CNN), which generates flight planning commands. Finally, trajectory planning and control module employs a *convex programming* technique to generate collision-free minimum time trajectory which follows these flight planning commands and produces appropriate control inputs for the quadcopter. The results convey unequivocally the advantages of depth perception by HSL, while repeatable flights of successful nature in typical indoor corridors confirm the efficacy of the pipeline.

CCS Concepts

•**Computing methodologies** → **Vision for robotics;** *Evolutionary robotics;*

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ICVGIP, December 18-22, 2016, Guwahati, India

© 2016 ACM. ISBN 978-1-4503-4753-2/16/12...$15.00

DOI: http://dx.doi.org/10.1145/3009977.3009990

Figure 1: Quadcopter autonomously navigating in indoor enviornment, bottom left sub-image depicting depth map of environment estimated using HSL and bottom right sub image show flight planning command generated by CNN

Keywords

Autonomous Navigation; Vision ;Quadcopter

1. INTRODUCTION

Recently, the miniature quadcopters have become an intriguing platform for contemporary new applications in indoor environment. Low cost, high maneuverability, hover capabilities have made them a versatile platform for research. Quadcopters could be used in security and surveillance tasks, where the capacity of flying above ground obstacles give them a great advantage over ground robots [1, 2]. However, miniature quadcopters are not able to carry power consuming sensors such as range finders and with unavailability of GPS in the indoor environment makes autonomous navigation a challenging task.

The solution presented in this work enables a miniature quadcopter with limited payload capability and frontal camera as primary sensor to navigate autonomously in an unknown GPS-denied indoor environment. The capabilities of the proposed system could be described as a twofold framework: (1) Dense depth map estimation: dense depth map is estimated from each individual video frame in real time, captured from the frontal monocular camera of the quadcopter using our novel supervised Hierarchical Structured Learning (HSL) approach. (2) Autonomous Navigation: estimated depth map is provided as an input to Convolutional Neural Network (CNN), which generates flight planning commands. These flight

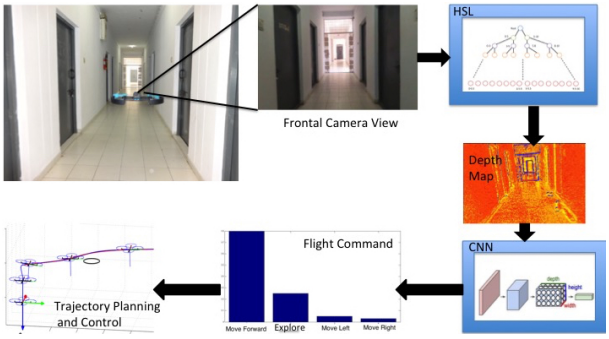

Figure 2: Navigation Framework Architecture: Image captured from frontal camera of quadcopter is provided as input to HSL. Depth map generated by HSL is provided as an input to CNN which generates flight planning command. Trajectory planning and control module generates the trajectory and control input for the quadcopter.

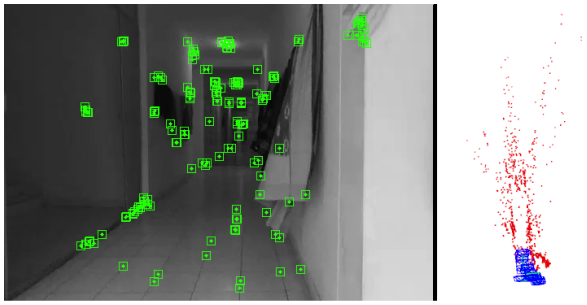

Figure 3: (a) Sparse ORB features (green boxes) detected in the indoor environment, not suitable for 3D navigation. (b) Overall features (red dots) detected and map created by ORB slam with blue box representing state of drone.

commands are provided to trajectory planning and control framework, which employs convex optimization for generating minimum time trajectory and controls for quadcopter, shown in Fig 2.

First, the system is primarily concerned with real-time estimation of depth map of environment from a single video frame. The RGB video captured from frontal camera of quadcopter is provided as an input to our novel HSL approach which generates dense depth map of the environment. The proposed supervised HSL approach, discretizes the overall depth range (0-10 meters) into multiple sets. It structures these sets hierarchically and recursively through partitioning the set of classes into two subsets with subsets representing apportioned depth range of the parent set, forming a binary tree. Binary classification is applied at each internal set separately using Support Vector Machine (SVM) classifier. The depth estimation of each pixel of video frame starts from root node in *top-down* approach, classifying repetitively till it reaches any of the leaf node representing its estimated depth. This allows transforming complex regression based depth estimation into multi-class classification problem utilizing multiple simple supervised learned binary classifiers, which is discussed in detail in Section 3.1. This approach is applicable for low altitude autonomous navigation where near-by object depth is more important than far off objects and also depth range is more important than exact depth of object. The depth estimation module of the framework appropriately utilizes the image descriptors-color, texture and orientation as the feature vector. Real-time dense depth map of the environment is estimated for each video frame, circumventing the need for VSLAM [3,4] based sparse reconstruction, which is often not suitable for navigation application.

Second, dense depth map estimated by Hierarchical Structured Learning (HSL) technique is provided as an input to Convolutional Neural Network (CNN), which predicts the flight planning commands (Go-Straight, Explore, MoveLeft, MoveRight etc), shown in Fig. 2. The CNN is trained using supervised learning approach with depth images being input and corresponding legitimate flight planning commands being the output. This proposed approach decouples depth perception from flight planning. Furthermore, as CNN accepts depth map as input thus allowing integrating either 3D LIDAR or Kinect based depth map etc in future. Our proposed approach makes CNN module less complex and more generic due to reduced dimensionality of input (depth map) $\mathbb{R}^3 \rightarrow \mathbb{R}^1$ and learning invariant depth map structures instead of RGB images, discussed in detail in Section 3.2. Estimated flight planning commands are provided as an input to the trajectory planning and control module, which employs convex optimization for generating minimum time trajectory and control inputs for quadcopter.

Current literature focuses only on estimating dense depth from a single frame without considering its applicability. Whereas, this paper presents novel HSL approach from the standpoint of robotics vision along with its application in autonomous navigation of quadcopter. Real-time performance, accuracy, adaptability and processing requirements were considered while designing the framework. HSL allows non-uniform quantization of overall depth range, providing flexibility to adapt the depth prediction to suite application requirements.

The quintessential contribution of this paper according to the authors lies in dovetailing very current, state of the art robotic vision and machine learning techniques to achieve real-time autonomous navigation of a monocular quadcopter, which continues to be a key and visible area of aerial vehicle research. To the best of our knowledge, such a system which seamlessly integrates dense depth estimation using Hierarchical Structured Learning (HSL) technique and flight planning using CNN for indoor autonomous navigation of miniature quadcopter with monocular camera as its primary sensor, as shown in Fig. 1 has not been presented in literature before.

2. RELATED WORK

The perception of the depth map of an environment is crucial to obstacle avoidance and autonomous navigation of quadcopters. The principal existing techniques for depth map estimation include monocular cues, structure-from-motion [4] and motion parallax. Moreover, structure-from-motion and motion parallax approaches require stable tracking between subsequent frames. The dense depth map estimation from video stream obtained from the frontal monocular camera of the quadcopter, in our proposed work, utilizes feature vector which combines monocular cues: texture, texture gradient and color information [5].

The depth estimation is normally considered to be a regression problem and considered to be more complex than multi-class classification. However, Hierarchical Structured Learning (HSL) for dense depth sensing is inspired by work of Ghosh et.al [6], which utilizes hierarchical structured binary classifiers to solve complex multi-class classification problem. Transforming depth sensing from regression to multi-class classification problem for navigation is similar to [7]. Multi-class classification using single complex model [7] is affected by outliers which is mitigated using our proposed HSL approach. Inferencing depth in a multi-label MRF framework [5] is often not suitable for real- time applications such as quadcopter navigation. Furthermore, DTAM [4] is susceptible to breakage (due to insufficient feature tracks) when quadcopter pitches or rolls significantly and also produces semi-dense depth map of environment which is not sufficient for autonomous navigation in 3D environment. DTAM also requires GPU support for generating real

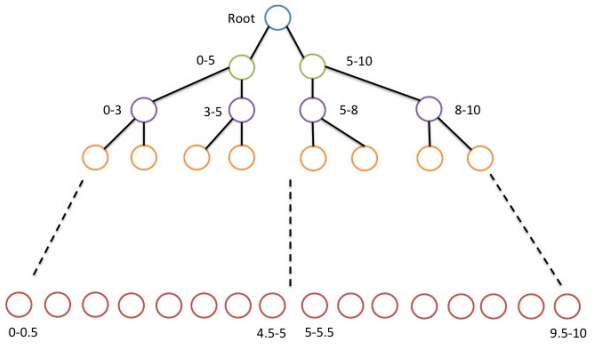

Figure 4: Hierarchical Structured Learning: Created binary tree representing the overall depth range of the environment (0.5-10 meters). Each internal node has separate binary classifier (SVM) and leaf nodes represent the final depth range.

time depth estimation whereas HSL require minimum hardware resources. PTAM [3] and ORB slam [8] as shown in Fig. 3 only produce sparse 3D reconstruction of the environment with average 100-300 features detection per frame whereas our novel HSL approach produces dense depth estimation of similar resolution as that of the input frame. [9, 10] both use CNN based single complex system model whereas our proposed HSL approach sub-divides the depth estimation in multiple sub problems and optimizes each individually for best results. HSL's real time CPU based performance allows on-board processing on miniature quadcopters in future, which may not be possible with other approaches. HSL is primarily designed specifically for autonomous navigation of quadcopter rather than accurate 3D reconstruction, which sets it apart from other approaches. This approach is suitable for autonomous navigation where depth range is more important than exact depth of object from quadcopter.

The Convolutional Neural Network (CNN) based flight planning approach is used in [11], which takes RGB images captured from frontal camera of quadcopter as input and predicts control commands for navigation. The quadcopter motion is controlled, using simple fixed values of control inputs–yaw, pitch and roll, which may not be able handle unplanned drift during the motion of quadcopter. Whereas, our proposed system utilizes minimum time trajectory planning and control module [7] which produces appropriate control inputs for the quadcopter based upon flight planning command. These control inputs are computed from the generated trajectory in each update. Hence, they are applicable to achieve closed-loop control similar to the model predictive controller. As a result, it is able to mitigate such drifts to a certain extent. Dey et al [12] depth perception is also inspired from [5] but utilizes single regression model for depth perception whereas we utilize our novel HSL approach. Ross et al [13] utilize Dagger algorithm for control whereas we suggest using CNN based control flight commands generation along with minimum time trajectory planning and control. Visual appearance of similar environments may vary but their 3D or depth structure is mostly invariant. Where [14, 15] learn using prior approach, our CNN learns from depth structure of environment thus making it more generic and effective in unknown environments. Integrating minimum time trajectory generation also ensures real time performance in complex environments.

Autonomous navigation of quadcopter in indoor environment has been studied previously using various techniques like SLAM [4], ultra-sonic or infrared sensors [16] and even laser scanners [17]. [18] navigates 3D indoor environment utilizing RGB-D sensor while [19] depends upon on-board laser sensors. These approaches either use costly sensors or sensors not suitable for miniature quadcopters due to their high power requirements. This paper focuses on autonomous navigation of miniature quadcopters with limited payload capacity and monocular camera as its primary sensor. Our cost effective solution is inspired from autonomous flight in indoor environment, work by Saxena et. al [20], which used perspective cues from a single image. Unlike [7], the higher level behavior decisions come from a learning module with the control loop being delegated to the lower level tasks of implementation of control commands.

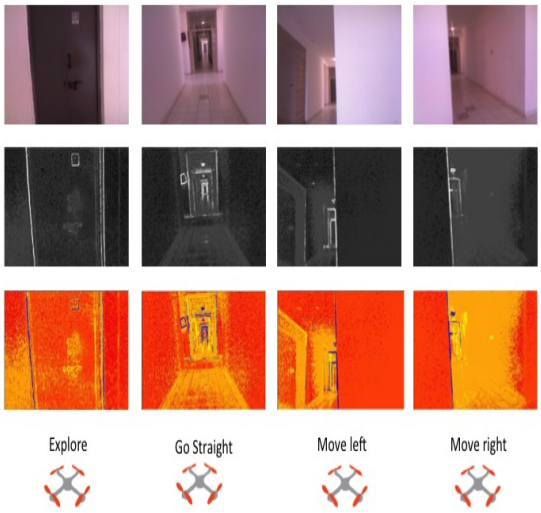

Figure 5: RGB images captured from quadcopter are shown in top row and corresponding depth map generated by HSL in second row with its heat map representation (dark orange near, light orange futher and blue farthest distance) in third row for user visualization. Last row represents the CNN generated flight planning command for the quadcopter.

3. AUTONOMOUS NAVIGATION FRAMEWORK

An indoor environment is the repetitive integration of few basic components like hallway, door, stair, wall etc with varying views. But certain characteristics of these components vary minutely irrespective of their location – depth map of the structure. Hallways have a unique depth structure of long empty space in the middle of the enclosed environment, which could easily be represented using depth map. Understanding their 3D structures instead of traditional RGB image view, could create a more generic solution for autonomous navigation. Therefore, instead of learning RGB based view of an indoor environment, we present a novel method for estimating dense depth map of the environment using our novel Hierarchical Structured Learning (HSL) approach and CNN based module for flight planning. The proposed framework is composed of two major components–dense depth map estimation from single frame obtained from frontal monocular camera using supervised Hierarchical Structured Learning (HSL) approach and CNN based module which takes dense depth map as an input and generates flight planning commands (MoveLeft, MoveRight, Go-Straight, Explore etc.) as output shown in Fig 5. Trajectory planning and control module then generates minimum time trajectory and control inputs for quadcopter based on the flight planning command.

The navigation framework consisting of (HSL, CNN and Trajectory planning and control modules) is implemented over Robot Operating System (ROS) and executed over conventional laptop, connected with quadcopter using Wifi. The video stream is captured from quadcopter frontal camera in real time and transfered

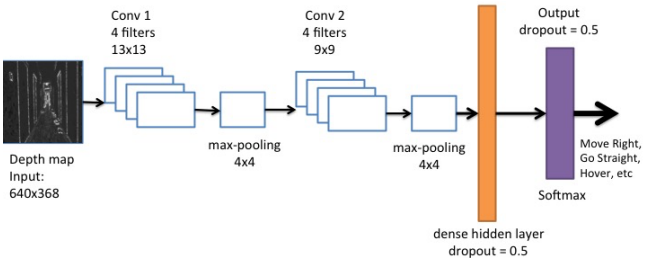

Figure 6: CNN architecture: Input depth map of 640×368 and 2 layers of Convolution and max pooling layers followed by hidden dense layer with dropout of 0.5. Softmax regression is applied at the output layer of CNN

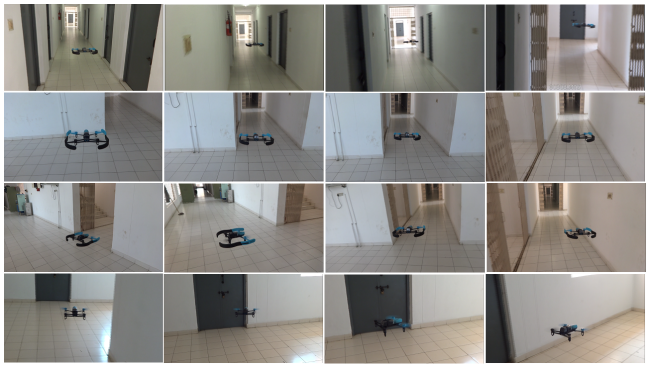

Figure 7: Quadcopter navigation in indoor enviornment: first row depicting GoStraight, second row MoveRight, third row MoveLeft and fourth row Explore.

over Wifi connection to HSL module, which estimates dense depth map for each individual frame. Afterwards, estimated dense depth is provided as an input to CNN module, which generates flight planning commands (MoveLeft, MoveRight, Go-Straight, Explore etc). These commands are utilized by trajectory planning and control module to generate control input for quadcopter. Overall system is able to generate depth map and flight planning commands based on video frame within $\sim 400ms$.

3.1 Hierarchical Structured Learning

The depth estimation module of the proposed navigation framework generates dense depth map in real time from each single frame captured from video stream. This novel approach uses feature vector derived from impressive work by Saxena et.al [5] and uses supervised learning approach for training. The training data set for indoor navigation is collected using Microsoft Kinect RGBD sensor which uses the infra-red light to compute the distance from the target objects. It consists of over 1600 RGB images $\mathbb{R}^3$ of various indoor scenes (wall, hall-way etc) with their corresponding ground truth depth images $\mathbb{R}^1$ representing depth between 0.5 to 10 meters. The images and corresponding depth map obtained from Kinect sensor were of 640×480 resolution. The dataset is divided into 4:1 split for training and testing purposes with total training labels ((number of pixels in each image) $\times$ (number of images in the training set) = $(640 \times 480) \times (1280) = 393216000$). Hierarchical Structured Learning (HSL) was trained in supervised learning approach utilizing RGB-D data from Kinnect with RGB images as input and depth map of environment as desired predicted output. Once trained, video from camera onboard the quadcopter is provided as an input and predicted depth map of the environment is generated as an output. HSL does not require GPU support for real time dense depth estimation and supports non-uniform quantization of depth range, allowing emphasis on certain part of depth range, not feasible with other approaches.

Multi-class classification using single complex model is affected by outliers and error is propagated to all classifications. Estimated classification accuracy may even be affected negatively by an increase in the number of classes. However, our approach has multiple binary classifiers arranged in a hierarchical binary tree where the flow of errors only restricted between siblings and parents nodes. Furthermore, an absolute value of error in depth prediction decreases with each correct prediction at previous levels while moving towards leaf nodes. With the increase in number of classes (n) the total number of comparisons required per pixel are only O(log(n)), height of the tree. Other approaches like "one vs one" and "one vs all" require O(n^2) and O(n) comparisons respectively [21].

In the proposed approach a depth value is estimated for each pixel of the image. The features used in this approach captures energy value by applying the *Laws* mask to the image intensity channel. Similarly, haze is captured by applying a local averaging filter to the color channels. Lastly, to compute the estimate of the texture gradient robust to noise, six oriented edge filters are convoluted with the intensity channel, more details in [5]. The computation of the feature vector $F_n(x,y)$, n = 1,...,17 where the n = size of feature vector (17), for any given pixel in the image I(x,y) comprises of 9 $Law's$ mask, 2 color channels and 6 texture gradients filters.

Initially, the real continuous depth values obtained from Kinnect sensor are quantized into sets S_i, i = 1,...,N, where the covering range of depth values for a set S_i from *near* to far utilizing uniform sampling. Non-uniform sampling could also be integrated, to have sets with smaller resolution for certain range of depth where accuracy is of importance and larger resolution otherwise, to satisfy application requirements. This approach converts the regression depth estimation into a multi class classification problem. These sets S_i, i = 1,...,N, are trained against feature vector using our novel Hierarchical Structured Learning (HSL) supervised learning approach. The depth range represented by these sets have inherent inter-dependencies which are exploited by our fast and intuitive HSL training process. It structures these sets hierarchically and recursively through partitioning them into two subsets where subsets represent apportioned depth range of the parent set, forming a binary tree, as shown in Fig. 4. A classification is applied at each internal node separately which uses Support Vector Machine (SVM) with the linear kernel for binary classification. The depth estimation for each pixel of image starts from root node in the top-down approach, classifying repetitively till it reaches any of the leaf node representing its estimated depth. The complex multi-class classification depth estimation problem is hierarchically transformed into multiple elementary binary *one vs one* classifiers based solution, which yields both high accuracy and better generalization. To achieve real-time performance the fast *liblinear* [22] package with C++ interface is used in our implementation.

3.2 Convolutional Neural Network Based Flight Planning

Convolutional Neural Network is trained using supervised learning method where dense depth estimated using our novel HSL approach is provided as an input along with respective flight planning commands – Move Left, Move Right, Go-Straight, Explore etc. Flight planning commands are generated and recorded manually by flying the drone using remote control in a similar environment. The implementation of Convolutional Neural Network used in our experiments is based on Lasagne package. Lasagne [23] is a lightweight wrapper over Theano python library to build and train

neural networks. The CNN architecture pipeline used in our experiments has following characteristics: An input dense depth image of size 640×368, followed by the layer consisting of 4 convolution filters of size 13×13 and the max-pooling layer with filter size of 4×4. Subsequently, another layer consisting of 4 convolution filters of size 9×9 and max-pooling layer with filter size of 4×4 are concatenated to the pipeline. The output of these layers is provided to the dense hidden layer which is followed by output layer at the end. Both the dense hidden layer and output layer have dropout of 0.5, Fig. 6 describe the complete architecture. The output layer uses Softmax regression which is generalized form of logistic regression. Furthermore, it is trained to provide flight planning command (MoveLeft, MoveRight, GoStraight, Hover, Explore etc) as output in the form of unique binary numbers.

$$Softmax\ P(x_{i,j}) = \frac{exp\ x_{i,j}}{\sum_k exp\ x_{i,k}} \tag{1}$$

$$Flight\ Control = argmax_{i,j}(P(x_{i,j})) \tag{2}$$

where x is an array input to the Softmax regression and element with max probability is selected as next flight planning command. If the selected command has probability below threshold value (experimentally found), default safety hover command is selected instead. Once reached, control is switched back to manual control mode.

3.2.1 convolution

Convolution layers perform 2D convolutions of their input maps with a rectangular filter. Higher activations will occur where the filter better matches the content of the map, which can be interpreted as a search for a particular feature. To add non-linearity, it uses ReLU activation function: $\phi_x = max(0,x)$.

3.2.2 max-pooling

The output of the max-pooling (MP) layers is formed by the maximum activations over overlapping square regions. MP layers decrease the map size, thus reducing the network complexity. MP layers are fixed, non-trainable layers selecting the winning neurons.

In most of the proposed approaches, RGB image is provided as input to CNN for learning [11]. However, our proposed framework provides estimated depth using HSL to CNN as an input thus reducing the input complexity ($\mathbb{R}^3 \rightarrow \mathbb{R}^1$) of the system. This also allows system to be more generic in nature as it learns minutely varying depth structures of the components of environment instead of the views in RGB. The CNN is trained with 50:50 split between training and testing dataset containing over 300 depth images each and respective labeled flight control. The selected depth images represent diverse scenes of environment from set of 1600. This distribution setup between training and testing dataset ensured varied and adequate testing to avoid over fitting condition. The CNN is trained using back propagation for either 500 epochs or if desired threshold accuracy is achieved, which requires about 48 hours on a workstation equipped with an Intel Core i7 CPU with 16 GB RAM. The estimated flight planning command is provided to trajectory planning and control module.

3.3 Trajectory Planning and Control

Unplanned drift in motion or error in state estimation prevalent in most quadcopters, may make quadcopter lean and crash. To avoid such a situation, a visual vanishing point based reference feedback mechanism was developed to align the drone to the middle of the corridor while in motion. This creates drift directional information for trajectory planning, which generates control commands for the quadcopter.

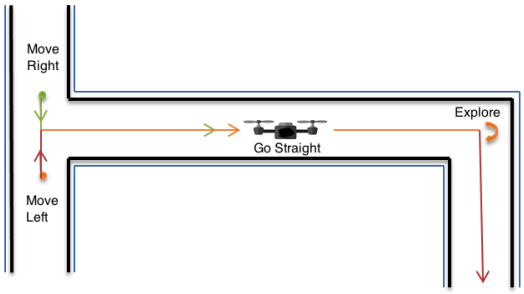

Figure 8: Quadcopter flying through hallway with symantic representation.

3.3.1 Vanishing Point

Indoor environment consists of straight parallel lines which allows use of proven vanishing point technique for navigation. The LSD [24] algorithm is used in our experiment for detecting lines in the environment. The lines should converge towards a point (vanishing point) but because of noise, all the lines may not intersect at the same point. The region in the image which has the highest density of pair-wise line intersections, indicates a high confidence index and thus contains the vanishing point [20]. To achieve this, the image plane is divided into an $M \times M$ grid G and middle of grid element $G_{p,q}$ with maximum intersections is selected as the vanishing point for the frame.

$$(p,q) = arg\ max_{(p,q)} G_{p,q} \tag{3}$$

The current frame vanishing point is likely to be in close proximity to the previous frame. Using this information, a linear motion model for Kalman Filter is constructed to suppress the noise in the vanishing point estimation. The deviation $(\Delta x, \Delta y)$ in vanishing point from the center of the image corresponds to the change in the heading angle γ.

$$\gamma = \tan^{-1}(\Delta x * \frac{field\ of\ view\ of\ camera}{width\ of\ image}) \tag{4}$$

The variation in the heading angle (γ) thus computed is fed to the trajectory generation to adjust the direction of motion.

3.3.2 Trajectory Planning

Quadcopter motion model used in experiment is described by $roll-pitch-yaw$ (θ,ϕ,ψ) set of Euler angles. The proposed trajectory planning and control module generates minimum time trajectory from the current position of quadcopter in every update cycle of the proposed navigation framework. Minimum time trajectory is generated with constraints:

$$\begin{aligned} Minimize \quad & : \Omega = |V_{max} - v_{tk}|^2 \\ Subject\ to & : v_{tk} <= V_{max},\ a_{tk} <= A_{max}\ and \\ & \quad j_{tk} <= J_{max},\ \forall k = 0,\ldots,n \end{aligned}$$

where v_{tk}, a_{tk}, j_{tk} are instantaneous velocity, acceleration and jerk of quadcopter at time tk and $V_{max}, A_{max}, J_{max}$ are the maximum velocity, acceleration and jerk constraints [7].

4. EXPERIMENTS AND RESULTS

We have evaluated the proposed framework on a low cost commercial Bebop quadcopter by Parrot [25] which is equipped with frontal monocular camera, ultrasound altimeter and onboard IMU. It transmits video frames at 640×368 resolution along with IMU

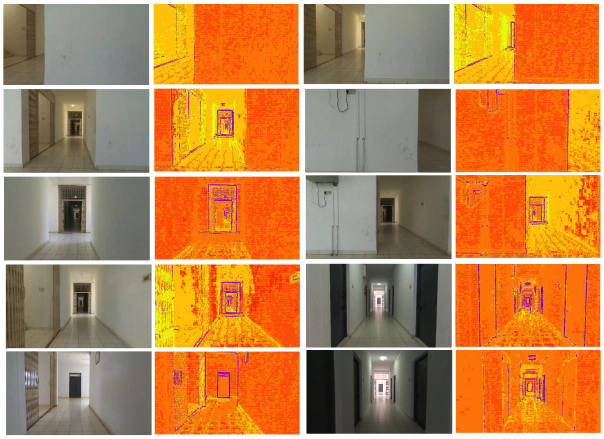

Figure 9: Estimated Depth using HSL with 1st and 3rd column depicting RGB images and 2nd and 4th columns depicting respective estimated depth using our novel (HSL) approach with color coding red, yellow and blue representing near, farther and farthest distances.

Table 1: HSL Depth Estimation

HSL Depth Estimation	Tolerance (meters)		
	0	0.5	1
Depth Estimation vs Ground Truth Accuracy %	16.89	46.14	52.63

information to the host device through wifi connection. For all the experiments including depth estimation, CNN flight planning, trajectory planning and control generation are carried out on a conventional laptop computer running ROS (Robot Operating System) as middle-ware over Ubuntu 14.04 LTS with Intel Core i5 processor @2.6 GHz and 8GB of RAM. Microsoft Kinnect was only utilized for collecting training dataset (RGB-D) of indoor environment for HSL with input being RGB image and depth map as learning parameter. The quadcopter training images and depth data were collected from corridors of multiple building and to ensure the efficacy of experiments, the quadcopter testing was performed on corridor not part of the training dataset.

The HSL approach requires ~ 0.18 sec to estimate dense depth for image of resolution 640×368 and CNN takes ~ 0.2 sec for generating flight planning command. The computation cycle time is approximately equal to system response time ($\sim$ 400ms) of Bebop justifying its practicality. Average accuracy of dense depth estimation using HSL is 16.89% while 46.14% accuracy is observed if tolerance range is increased to $+/-0.5$ meters, as shown in Table 1. Moreover, overall error in the depth prediction using HSL is 0.531 on log distance: $E_{log} = \frac{1}{NxM}\sum|\log(D_g) - \log(D_p)|$, where D_g ground truth depth, D_p estimated depth, N is rows and M columns of image. The accuracy of depth estimation achieved using HSL at some of the internal tree nodes when compared with ground truth

Table 2: Classification Accuracy of HSL nodes

Binary Nodes of HSL (meters)	Classification Accuracy %	Log Distance Error
0-5	74.07	0.256
0-4	73.57	0.263
0-3	57.84	0.293
0-2	12.07	0.613

Table 3: Autonomous Navigation Experiments

Autonomous Navigation	Successful Navigation	Unsuccessful Navigation	Total No of Experiments
Indoor Navigation	34	7	41

is depicted in Table 2. Overall accuracy of classification decreases as we move from top to bottom of HSL tree structure as shown in Table 2 due to a) error predictions of parent nodes are processed by children nodes b) error in prediction at each node of HSL. The data in HSL node representing depth range 0-5 meters as shown in Table 2 is classified further by 0-4 meters, 0-3 meters and 0-2 meters. Overall prediction accuracy decreases as the data moves in top-down approach between nodes due to above mentioned justification. CNN classification is tested on testing dataset of over 300 images and 92.66% accuracy is achieved. Experiments are conducted numerous times in indoor environment, Fig 8 with successful navigation rate of 82.04%, as shown in Table 3. Inherent drift in motion of quadcopter, incorrect depth map estimation due to uneven lighting in hallways, astray vanishing point detection resulted in unsuccessful navigation attempts of the quadcopter. The Fig. 7 depicts the overall motion of quadcopter in indoor environment. First row in Fig. 7 shows GoStraight motion of quadcopter. Second row depicts sequence of MoveRight followed by GoStraight motion after detecting the hallway. Similarly third row shows MoveLeft followed by Gostraight motion profile. Fourth row represents the Explore motion profile after detecting the end of the hallway where quadcopter turns 90^o and move towards next hallway. Fig. 9 showcases HSL depth estimation results in various scenarios in indoor environment. The experiment video showcases the efficacy of our framework for autonomous navigation of quadcopter. The framework is even able to mitigate the effects of unplanned drift in motion while navigating in indoor environment.

5. DISCUSSION

Our proposed approach utilizes Microsoft Kinnect for labeling ground truth depth subsequently used for HSL training but observed to be ineffective in places with direct sunlight, bright objects etc which may introduce outliers or errors in training dataset for HSL. Stereo camera ZED is initially tested but found to be ineffective for generating depth ground truth due to large Min distance (1.5 meters) for depth perception and ineffective on walls with very less texture. The accuracy of the depth perception could be increased by using higher resolution Microsoft Kinect 2 Sensor or using 3D LIDARs. CNN module currently predicts using Intel Core i5 CPU as main processing unit but with increase in complexity of model in future, GPU based model may be chosen. Theano which has dynamic GPU/CPU selection and capability for handling complex Neural Networks made it optimum choice for this experiment. The system consisted of Intel Core i5 CPU and Nvidia 820M 2GB graphics card. Experiments were conducted to evaluate the overall real time prediction performance between CPU and GPU with results recorded in Table 4. GPU based CNN prediction was comparatively faster than CPU but with only 10% improvement. This could be attributed due to simple design of CNN and small 640×368 image size.

Learning rate and momentum play an important role in final accuracy of classification achieved by neural networks. Small value may cause the system to never reach the goal or being stuck in local minima and large value may cause it to keep overshooting the goal value without ever reaching it. These parameters were determined

Table 4: CPU vs GPU performance for CNN prediction

Frame	CPU (Sec)	GPU (Sec)
1	0.277	0.270
2	0.276	0.271
3	0.311	0.272
4	0.300	0.284
5	0.280	0.278

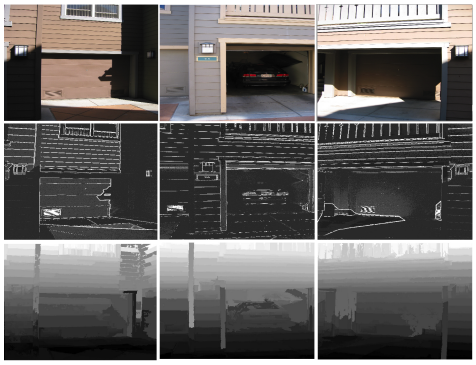

Figure 10: Top row: original Image, middle row: HSL predicted depth and last row : Make3D predicted depth.

experimentally for our work with learning rate $\eta = 0.005$ and momentum $m = 0.01$. Outdoor navigation using similar methodology was also approached but due to limited functionality of Microsoft Kinect in sunlight and above mentioned limitations of ZED stereo camera, activity is deferred. Bebop quadcopter dynamical model is not available to public from the manufacturer but is managed by its internal proprietary software.

The effectiveness in terms of depth prediction accuracy and real time performance of our novel HSL approach was compared with Make3D [5]. Real time performance comparison between the two approaches is described in Table 5. Make3D treats the depth prediction problem as a regression problem whereas HSL as a multi-class problem. Thus for comparison with HSL, predicted depth values of Make3D were discretized into predefined number of bins. Depth prediction accuracy was calculated using formula log distance:

$$E_{log} = \frac{1}{NxM} \sum |\log(D_g) - \log(D_p)|$$

and comparative results are shown in Table 6. It could be seen that HSL provides comparative depth prediction accuracy to Make3D but with real time performance. The depth prediction comparison output is show in Fig. 10. It was observed that Make3D excelled in predicting depth of continuous surfaces but absolute error in depth per region of image was lower in HSL output.

Acknowledgement

This work was supported from grants made available by DeitY under the National Program on Perception Engineering - Phase II.

Table 5: Real time performance comparison between HSL and Make3D

	Depth Prediction Time (Sec)	Image Size	Output Depth Map Size
Make3D	56	1704×2272	900×1200
HSL	0.7		1700×2268

Table 6: Depth prediction comparison between HSL and Make3D

	Depth Prediction Log Error	Discretized Depth Level
Make3D	0.698	20
HSL	0.710	

6. CONCLUSION

This paper presents novel approach to achieve real-time autonomous navigation of a miniature quadcopter. We present a system which seamlessly integrates real time dense depth estimation using Hierarchical Structured Learning (HSL) technique and flight planning using CNN for indoor autonomous navigation of quadcopter with monocular camera as its primary sensor. The numerous experiments convey unequivocally the advantages of depth perception by HSL, while repeatable flights of successful nature in typical indoor corridors confirm the efficacy of the framework. In future, autonomous navigation in natural outdoor environment using similar approach along with moving obstacle avoidance is being actively persuaded.

7. REFERENCES

[1] Aveek Purohit, Zheng Sun, Frank Mokaya, and Pei Zhang. Sensorfly: Controlled-mobile sensing platform for indoor emergency response applications. In *IPSN*. IEEE, 2011.

[2] Joaquin López, Diego Pérez, Enrique Paz, and Alejandro Santana. Watchbot: A building maintenance and surveillance system based on autonomous robots. *Robotics and Autonomous Systems*, 2013.

[3] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *ISMAR 2007*. IEEE.

[4] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *ICCV*. IEEE, 2011.

[5] Ashutosh Saxena, Sung H Chung, and Andrew Y Ng. Learning depth from single monocular images. In *Advances in Neural Information Processing Systems*, 2005.

[6] Yangchi Chen, Melba M Crawford, and Joydeep Ghosh. Integrating support vector machines in a hierarchical output space decomposition framework. In *Geoscience and Remote Sensing Symposium*, volume 2, pages 949–952. IEEE, 2004.

[7] Kumar Bipin, Vishakh Duggal, and K Madhava Krishna. Autonomous navigation of generic monocular quadcopter in natural environment. In *ICRA*. IEEE, 2015.

[8] Raul Mur-Artal, JMM Montiel, and Juan D Tardós. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 2015.

[9] Fayao Liu, Chunhua Shen, and Guosheng Lin. Deep convolutional neural fields for depth estimation from a single image. In *ICCV*, 2015.

[10] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015.

[11] Dong Ki Kim and Tsuhan Chen. Deep neural network for real-time autonomous indoor navigation. *arXiv preprint arXiv:1511.04668*, 2015.

[12] Debadeepta Dey, Kumar Shaurya Shankar, Sam Zeng, and Rupesh Mehta. Vision and learning for deliberative monocular cluttered flight. Springer, 2016.

[13] Stéphane Ross, Narek Melik-Barkhudarov, and Kumar Shaurya Shankar. Learning monocular reactive uav

control in cluttered natural environments. In *ICRA*. IEEE, 2013.

[14] Alessandro Giusti, Jérôme Guzzi, Dan C Cireşan, Fang-Lin He, Juan P Rodríguez, Flavio Fontana, Matthias Faessler, Christian Forster, Jürgen Schmidhuber, Gianni Di Caro, et al. A machine learning approach to visual perception of forest trails for mobile robots. *IEEE Robotics and Automation Letters*, 2016.

[15] Shreyansh Daftry, J Andrew Bagnell, and Martial Hebert. Learning transferable policies for monocular reactive mav control. *arXiv preprint arXiv:1608.00627*, 2016.

[16] James F Roberts, Timothy Stirling, Jean-Christophe Zufferey, and Dario Floreano. Quadrotor using minimal sensing for autonomous indoor flight. In *EMAV, 2007*.

[17] Matthias Nieuwenhuisen, David Droeschel, Marius Beul, and Sven Behnke. Obstacle detection and navigation planning for autonomous micro aerial vehicles. In *(ICUAS)*. IEEE, 2014.

[18] Zheng Fang and Sebastian Scherer. Real-time onboard 6dof localization of an indoor mav in degraded visual environments using a rgb-d camera. In *ICRA*. IEEE, 2015.

[19] Abraham Bachrach, Samuel Prentice, Ruijie He, and Nicholas Roy. Range–robust autonomous navigation in gps-denied environments. *Journal of Field Robotics*, 2011.

[20] Cooper Bills, Joyce Chen, and Ashutosh Saxena. Autonomous mav flight in indoor environments using single image perspective cues. In *ICRA*. IEEE, 2011.

[21] Jonathan Milgram, Mohamed Cheriet, and Robert Sabourin. âĂIJone against oneâĂİ or âĂIJone against allâĂİ: Which one is better for handwriting recognition with svms? In *Tenth International Workshop on Frontiers in Handwriting Recognition*. Suvisoft, 2006.

[22] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.

[23] Lasagne. Python. http://lasagne.readthedocs.org/en/latest/user/installation.html.

[24] Rafael Grompone von Gioi, Jeremie Jakubowicz, and Morel. Lsd: A fast line segment detector with a false detection control. *PAMI*, 2008.

[25] Parrot. Bebop drone. http://www.parrot.com/products/bebop-drone/.