2021 - Crosscutting Research and Advanced Energy Systems Project Review Meeting

Autonomous Aerial Power Plant Inspection in GPSdenied Environments

The University of Texas at El Paso Aerospace Center (cSETR)

Investigators: Dr. Angel Flores Abad and Dr. Ahsan Choudhuri Graduate Students: Julio Reyes Munoz, Guillermo Ortega, Mousumi Rizia and Habib Noshin

May 18, 2021









- Project Description
- Autonomous Drone Platform Updates
- □ Flight Path Generator
- □ Autonomous Navigation in GPS-denied environments
- □ Inspection Payload
- □ System Interface

Project Description

Global planning layer: inspired by synthetic vision. Here a 3D CAD model will act as the synthetic vision system

Local-reacting layer: consists of stereoscopic vision sensors that cover all three axis of the vehicle.



Technology Status and Bench marketing

_

Features:

- Autonomous
- GPS-denied areas
- Close range within 1 ft.
- Can Inspect insideoutside
- Dusty and ashy environments



Intel® Falcon[™] 8+ Drone







Autonomous Drone Platform Updates



The altitude data is fused with the visual estimation using the EKF implemented in PX4.

-New mounting parts - Vibrations reduction - Stable flight



Autonomous Navigation

- Current software architecture



- The software architecture has a modular design with processes running in three different computer. Two of them onboard and one offboard.
- The communication is achieved through the MAVLink protocol running over TCP/IP and UART.

Autonomous Navigation -Implementation of heading setpoints: Yaw angle control

characterization



The inspection cameras are pointing normal to the surface of interest.



Realsense T265 Characterization

Light filtering did not improve 3D tracking.



Autonomous Navigation -Data acquisition in an industrial environment

im061.jpeg	im062.jpeg	im063.jpeg	im064.jpeg	im065.jpeg	im066.jpeg
im067.jpeg	im068.jpeg	im069.jpeg	im070.jpeg	im071.jpeg	im072.jpeg
					7 31
im073.jpeg	im074.jpeg	im075.jpeg	im076.jpeg	im077.jpeg	im078.jpeg
morsjpeg	into rapeg	mor s.peg	morojpeg	inio//.jpeg	morojpeg
im079.jpeg	im080.jpeg	im081.jpeg	im082.jpeg	im083.jpeg	im084.jpeg
	- Constant	and I			
im085.jpeg	im086.jpeg	im087.jpeg	im088.jpeg	im089.jpeg	im090.jpeg
im091.jpeg	im092.jpeg	im093.jpeg	im094.jpeg	im095.jpeg	im096.jpeg



• Flight tests were conducted in a real industrial environment. The left image shows the output of the image acquisition system.

Autonomous Navigation -Trajectory estimation tests in an industrial environment



- The initial tests consisted in manually moving the system around the structure of interest for validating an accurate position estimation.
- The trajectory estimation was acceptable for proceeding with autonomous flights.

Autonomous Navigation -Autonomous wall scanning in an industrial setting



• An autonomous scanning mission was executed along one of the walls, going from side to side at four different altitude levels.



Autonomous Navigation -Autonomous flight tests inside a constrained space



• The system was finally tested in a constrained environment by executing an autonomous mission inside a storage rack.



Photogrammetric Reconstruction

Acceleration Park Hangar (Fabens, TX)

File Edit Filters Render View Windows Tools Help			. 8 :
▷ # ≥ ◊ ◈ ◙ ◙ ⊕ ◻ ∷ ଷ 📊 ▮ ㅎ 卷 ⊕ Ѧ ⊬ ๕ Ҽ 🖌		Desired 1	0
		I I	4
FOV: 60 FPS: 227.3 BO_RENDERING	Mesh: texturedMesh ob) Vertices: 203,596 Faces: 248,345 Selectori.v: 0f. 0	Test/Heshroom/Heshroom Files/ Heshroom(acche/Texturing/ 24b5f/78987d4dcbc98c9313508319ee26 92367/Modified Moniform texturedHesh.obj in 1546 asec	
	Selection: v. 0 f. 0 FC WT	เมษระกาวเกรีย เป็นก	~

Photogrammetric Reconstruction/STL Acceleration Park Hangar (Fabens, TX)







Trajectory Generation

Acceleration Park Hangar (Fabens, TX)





Offline Trajectory Flight Mission Acceleration Park Hangar (Fabens, TX)



Enhancing Localization by Combining SLAM and CAD Model

1. ORB-SLAM2 Library

- 2. RealSense Depth Camera D435 for RGB-D Mode
- 3. Exploring the use of Localization Mode to initialize drone in CAD/STL environment and trajectory
- 4. Save Map using existing extensions
- 5. Working on generating a mesh (STL or the like) from PointCloud data.
- 6. Aim to enhance/replace CAD with SLAM and Photogrammetric Reconstruction



Photogrammetric Reconstruction

C MeshLab 2020.07 - [Project_1]		- 0 ×
File Edit Filters Render View Windows Tools Help		_ 8 ×
	- 0 ≝ Z ∰ % % ½ ∰ % V % %	Q
		Project_1
		textured/tesh.ob) Shading Vert Face None Color Face Mesh User-Oef Back-Face Single Double Fancy Cull Texture Coord On Off apply to all visible layers _
FOV 60 FPS: 7.5 BO_RENDERING	Mesh. toduredMesh.obj Veritces: 993,320 Faces: 1,947,022 Selection v. 0 f 0 FC WT	rectagular region Paptied 10 Sees Applied 11 Fedelet Selected Faces in 926 asec Started Mode Select Faces in a rectagular region Deleted 171 faces Applied filter Delete Selected Faces in 973 masc Started Mode Select Faces in a rectagular region Deleted 28 faces. Applied filter Delete Selected Faces in 902 masc V
		UNRECOGNIZED CARD

SLAM Reconstruction Setup

- 1. Handheld device to achieve full control that employs same components as the UAV.
- 2. Need for a faster approach that is not as computationally expensive as Photogrammetric Reconstruction.
- 3. Semi-Dynamic environments (Static during inspection but dynamic over days or weeks).
- 4. Arises from the need to have most current CAD model.
- 5. Control over features detected in live image acquisition vs static RGB images that might fail in Photogrammetry.



SLAM Reconstruction Results

- 1. Original RGB feed from D435 (Top-Left).
- 2. Modified feed with Feature Detection (Under Previous).
- 3. 3D Point Cloud of features being mapped.



SLAM Reconstruction Results



Inspection Payload



1. **RGB Camera**: Raspberry Pi (Sony IMX219 Sensor)

8 Megapixel Focal length: 3.04 mm

Vertical FOV: 48.8, Horizontal FOV: 62.2 Works with Gstreamer Pipeline on Jetson Nano, coded with Python.

2. **IR Camera**: FLIR Lepton 3.5 camera, Purethermal 2.0

LWIR camera temp range up to 400°C (752°F)

3. **Platform/SBC**: NVIDIA Jetson Nano GPU: 128-core NVIDIA Maxwell[™] CPU: Quad-core ARM® A57.

Low power requirement considering UAV power limitation.

oCoding Language: Python

Defect analysis: AI/DNN Object Detection model in Python (Tensorflow) - Offline

- Models:
- Dataset: created with crack and corrosion together. About 300 images.

1. DNN model YOLOv4-tiny (Precision vs inference speed restriction), customized by transfer learning and trained on Google Colab (training time over 6 hours)

Classes: Crack

Evaluation: e.g. mAP@0.5) = 0.379753 only

2. DNN model YOLOv4, customized by transfer learning and trained on Google Colab (training time over 12 hours)

Classes: Crack & Corrosion

Evaluation: e.g. mAP@0.5 = 0.55

Tools:

Keras & Tensorflow, both open-source neural-network libraries written in Python, with TensorFlow backend.



Payolad Inspection: Intelligent Inspection

I. Offline Inspection using Deep Learning: Inference using 2 quantized models on saved videos (source: online)



(a) Custom TensorRT model



(b) Custom TensorFlowLite model

Media 01. Inference on Saved Videos

Intelligent Inspection on Edge

II. Inspection using AI on Edge: Inference of TensorRT models using SBC onboard camera (Raspberry Pi)



(a) Indoor test in ambient (natural) lighting



(b) Outdoor test in Shaded area in ambient (natural) lighting

Media 02. Real-time Custom TensorRT model Inference

Intelligent Inspection

II. Inspection using AI on Edge: Inference of TensorRT models using SBC onboard camera (Raspberry Pi)





(c) Outdoor test under direct sunlight

(d) Outdoor test under direct sunlight

Media 03. Real-time Custom TensorRT model Inference

IR analysis

- Setup FLIR Lepton 3.5 camera with Purethermal 2.0
 - Installation and compilation of needed libraries e.g. libuvc etc
- Python code
 - to capture/acquire image with pseudo color (.jpeg format, legible but looses temperature data)
 - to capture/acquire image without colormap (.tiff/.png format), keeping temperature data, but not discernable)
 - To normalize temperature image to **radiometric** feed (.tiff/.png format) to make discernible and image saving in .hdf5 format
- RGB and IR camera parallel video feed sync (Python, OpenCV, with inference running on RGB display/feed) in Jetson Nano
- Python code to obtain pixelwise temperature, minimum, maximum etc from IR images



Lepton_Capture_2.jpeg







Lepton_CaptureY16.tiff

IR analysis





Fig. Synchronized Visual and Infrared image acquisition and thermal thresholding

Progress:

• Synchronized IR thresholding (fig above) and 'Thermal spot-meter' (reading temperature at a pre-selected pixel coordinate)

Drone Interface Architecture

 $MongoDB-NoSQL\ database\ with\ flexibility\ in\ \ data\ type\ storage$

Express – Backend framework supporting various middleware with a robust API

React – Javascript library for building user interface

Node.js – Javascript runtime environment with node package managers to create modules



Drone Interface Frontend

Allows specific users to upload images and videos

•

٠

٠

User IP addresses must be added when creating the user but is not publicly stored

Stores images and videos into the MongoDB database

Can be accessed through the URL: https://droneinterface.herokuapp.com/

Drone Interface	Existing Users	Images	Videos	Create New User	
Logged User	S				
Username					Actions
Noshin					delete

Drone Interface Frontend

Drone Interface Existing Users Images Videos Create New User

Upload

Choose File to Upload

Choose Files No file chosen

Current Images

Image	Uploaded	Size
	10/20/2020 7:26:31 PM	200.5KB Remove

Drone Interface Frontend

Drone Interface Existing Users Images Videos Create New User

Choose File to Upload

Choose Files No file chosen

Upload

Current Videos

Video	
► 0:00/1:06	:

Uploaded	Size	
11/4/2020 2:36:54 PM	76063.1KB	Remove

Drone Interface Existing Users Images Videos Create New User

Create New User

Username:

IP Address:

Create User

Technology to Market Path Industry partners: El Paso Electric





Acknowledgment and Disclaimer

Acknowledgment: "This material is based upon work supported by the Department of Energy Award Number DE-FE0031655."

Disclaimer: "This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof."