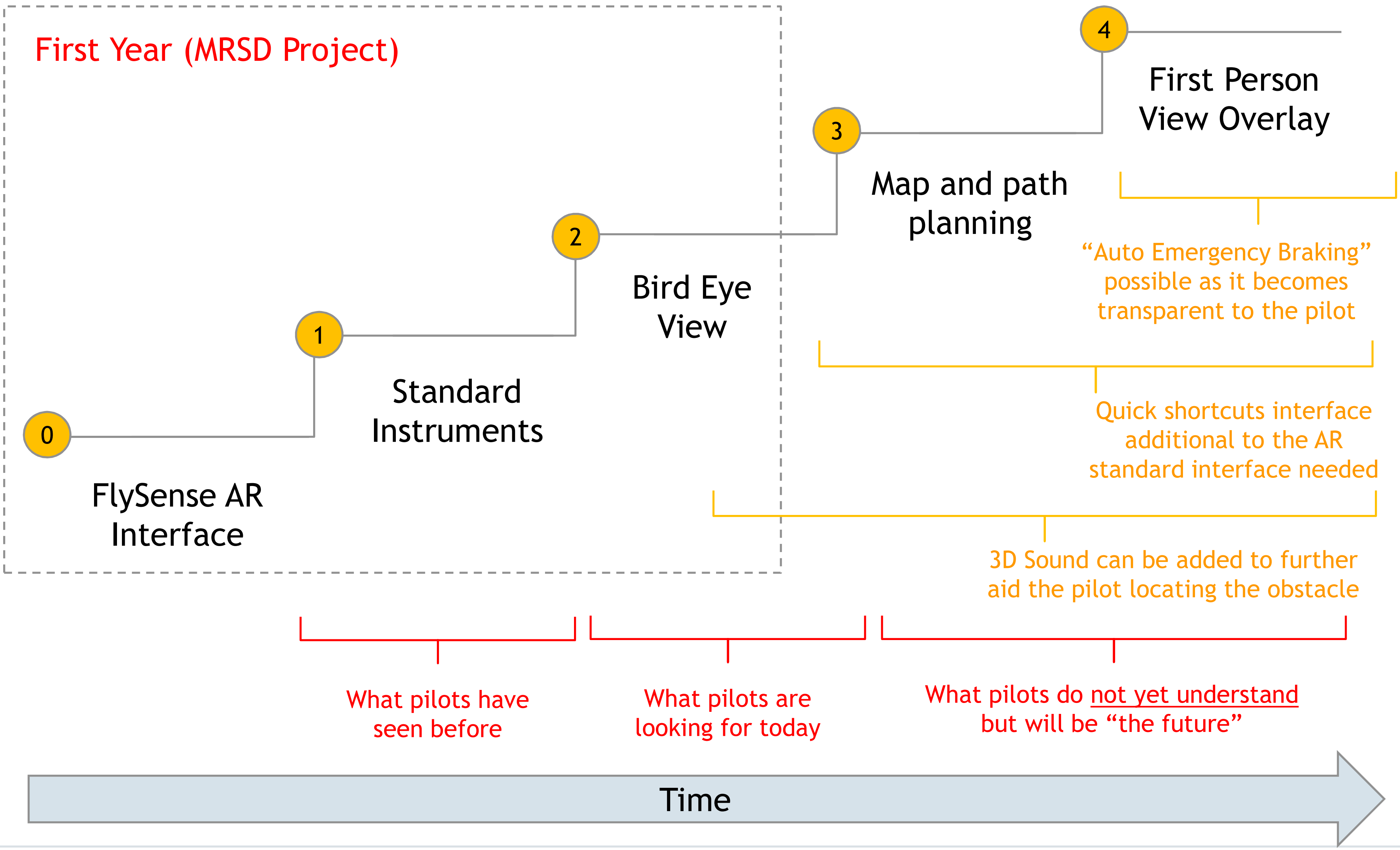
FlySense

Augmented Reality FPV assisted navigation (applied to a helicopter)

Team C
October 2017

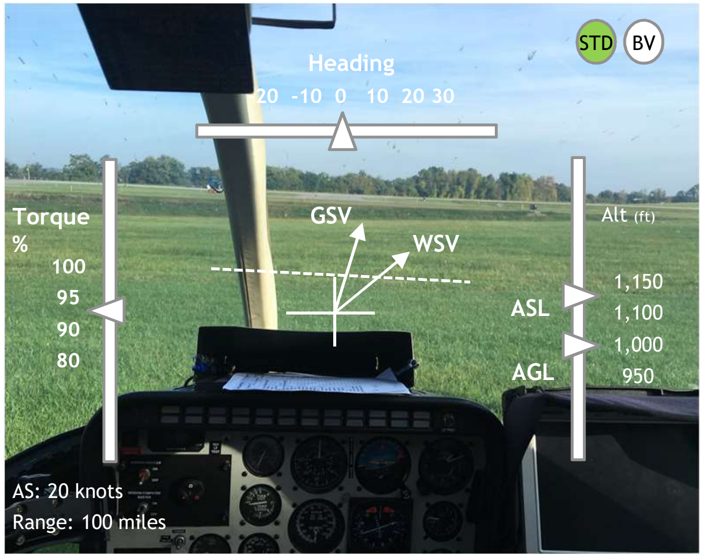
Shivang Bhaveja
Harikrishnan Suresh
Nick Crispie
Joao Fonseca
Sai Nihar Tadichetty

Previous features were segmented based on feedback gathered so far



Requirements downsized to three basic features: "User interface (including voice and 3D sound)", "standard instruments" and "birds' eye view"

- Default is no AR display on to prevent cluttering pilot's view
- Pilot can turn STD instruments on at the beginning of the mission (gestures) or voice commands
- Bird's Eye view can be enabled through voice commands or pops-up automatically near obstacles



- Information from sensors (heading, altitude, avg altitude above ground, Torque, avg speed)
- Combined information from helicopter IMU and AR IMU (artificial horizon animation)
- Presents information computed with inputs from helicopter computer (range from fuel rate

- Cloud point from Velodyne is inside widget with scale of the circle indicated on the side
- All computations and sub-selections are computed at the FlySense Onboard Computer
- Scale and pose of data is selected based on helicopter dynamics

ProjectFolder-AreaFolder-Name-Date-DesktopPublishing-Author

NEA prioritized four different scenarios where the bird's eye view will be tested

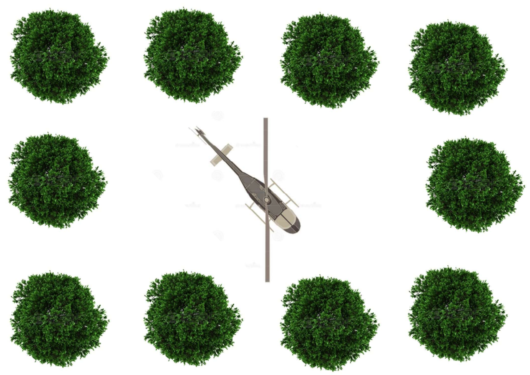
Take-off: Obstacle saturated environment

Pilot backs helicopter, yaws and climbs up at 6^0 ascend path angle (VTO consumes a lot of fuel)

Take-off: Isolated obstacle

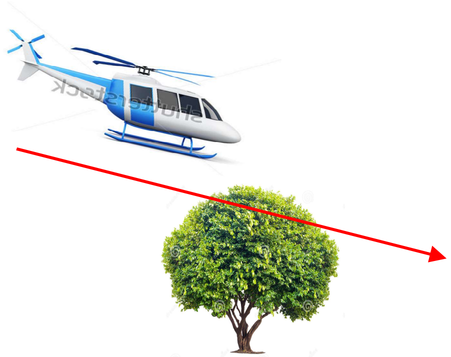
Pilot climbing at 6^0 descent path angle needs to do a quick pull-up to avoid isolated obstacle (hidden by fuselage)

Landing: Obstacle saturated environment



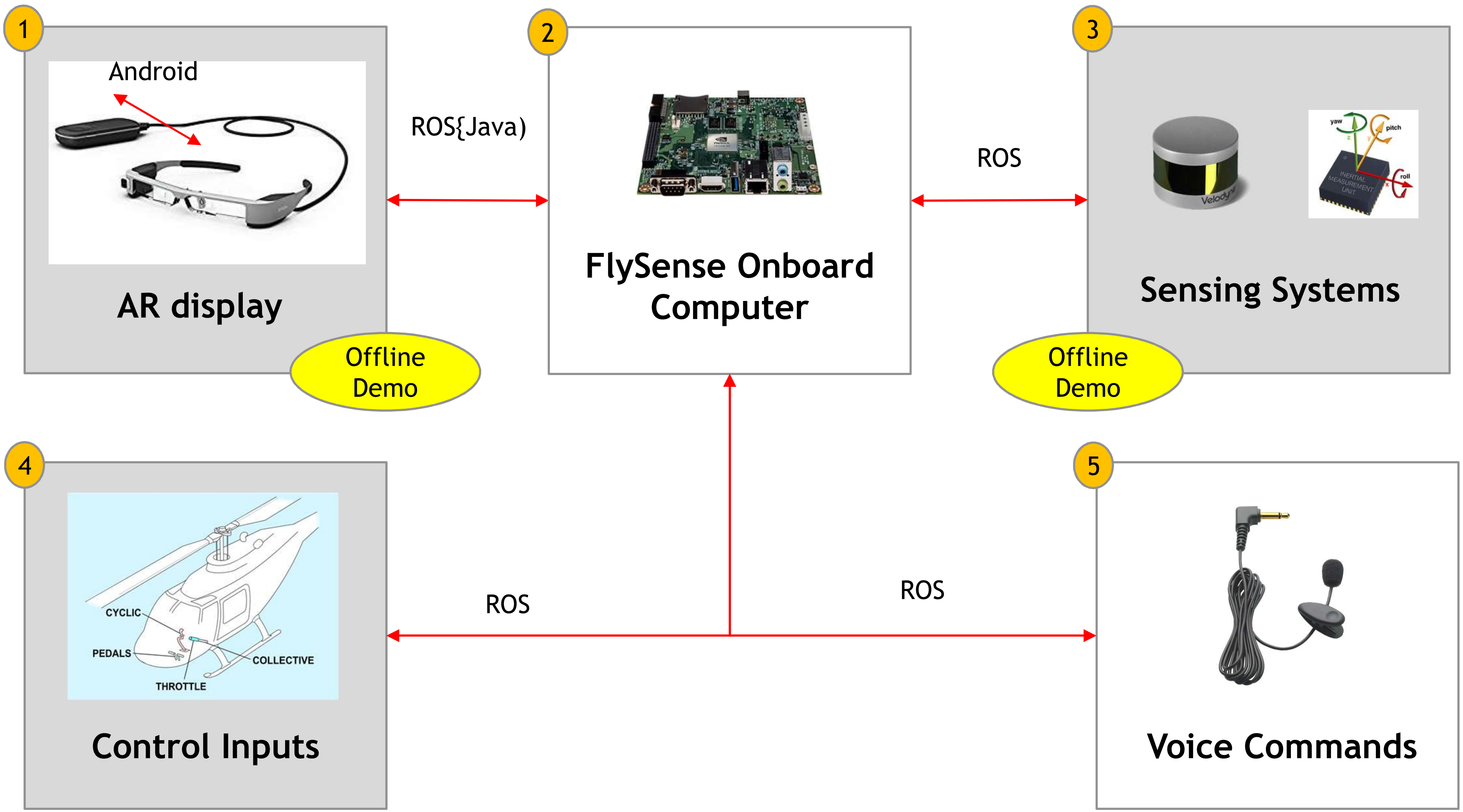
Pilot needs to steer the rotor away from obstacles while performing an almost vertical landing

Landing: Isolated obstacle



Pilot descending with a steep descend path angle (12^0) and does not see isolated obstacle

The cyber-physical architecture was updated to work with NEA helicopter infrastructure



NEA labelled AR interface as one of the most important features of the FlySense solution and thus we are now initiating the testing of the second AR unit

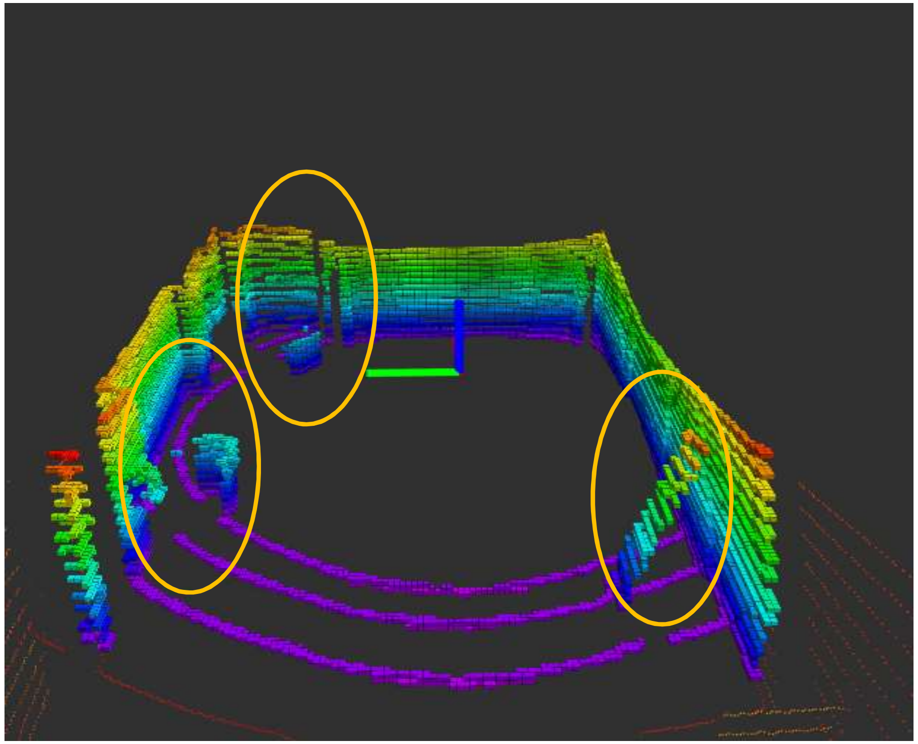
Trade Study winner “Hololens” tested in depth
(borrowed unit from Airlab)

- **Pros**
  - Easy to build applications
  - Excellent head tracking (gaze)
  - 3D Audio (with no background noise)
  - Holographic images (3D virtual world)
  - Resilience to lighting (overlaid graphs)
  - Works both in wireless and wired mode
  - Voice Commands (with no background noise, but only works online)
- **Cons**
  - Windows based development (NEA uses ROS)
  - Limited field of view (-30 to 30 degrees)
  - Feels heavy after using it for 30/40 minutes
  - Hand tracking not good enough for interaction
  - Hologram alignment lost if headset moves relative to the head of the user
  - Voice commands do not work with helicopter background noise (tested at Robolounge)

Next step: Acquisition of new AR unit (Epson)

- **Why Epson might be more adequate?**
  - No need for holograms as per the new streamlined scope (three features only)
  - Android based development (ROS-Java) easy to integrate on NEA helicopter
  - Epson AR is used on a start-up that is trying to do similar AR interfaces (track record)
  - Substantially lower cost compared to Hololens from Microsoft
- **Additional feedback gathered from pilots and CMU experts (Aaron Steinfield/Jean Oh)**
  - Pilots prefer lighter units
  - Pilots prefer voice commands to gestures
  - Voice Commands better than hand gestures
  - The simplest the interface the better
  - Vibrations in helicopter could change relative posture of headset vs pilot head

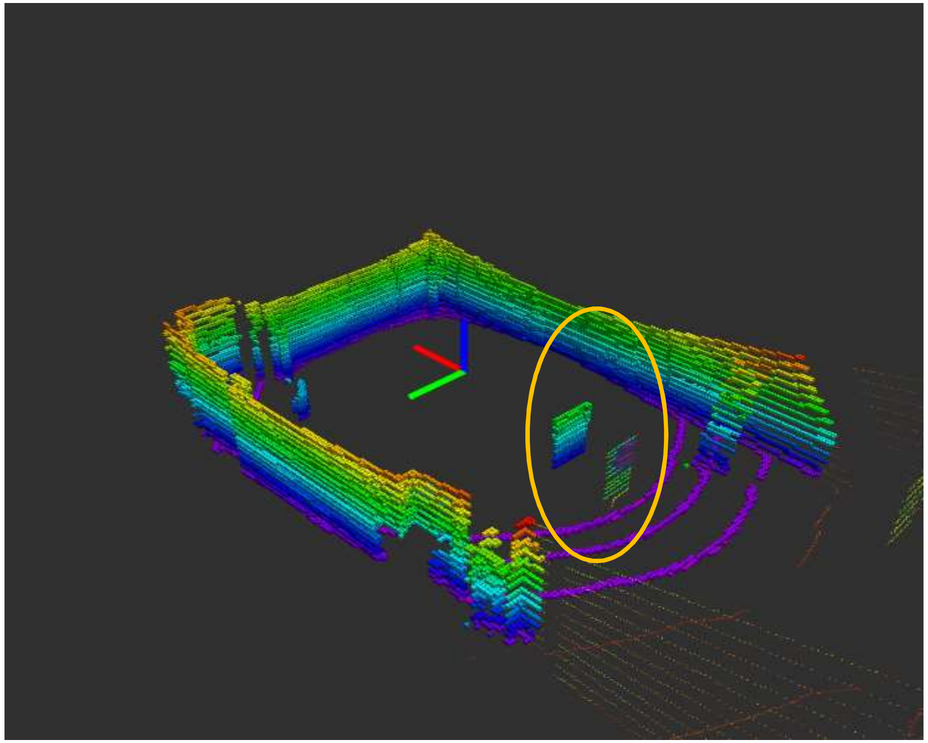
First attempt at 3D mapping for static obstacles



Take aways

- **Objective of test:**
  - Test octomaping algorithm for static obstacles of different sizes
- **What we learned:**
  - Wall and obstacles were detected and reconstructed
  - Floor detection parameters need to be optimized
  - 3D mapping algorithm range needs to be expanded
  - Artefacts present in 3D map
- **Next steps:**
  - Optimize parameters (floor and range)
  - Be able to extract only the relevant information

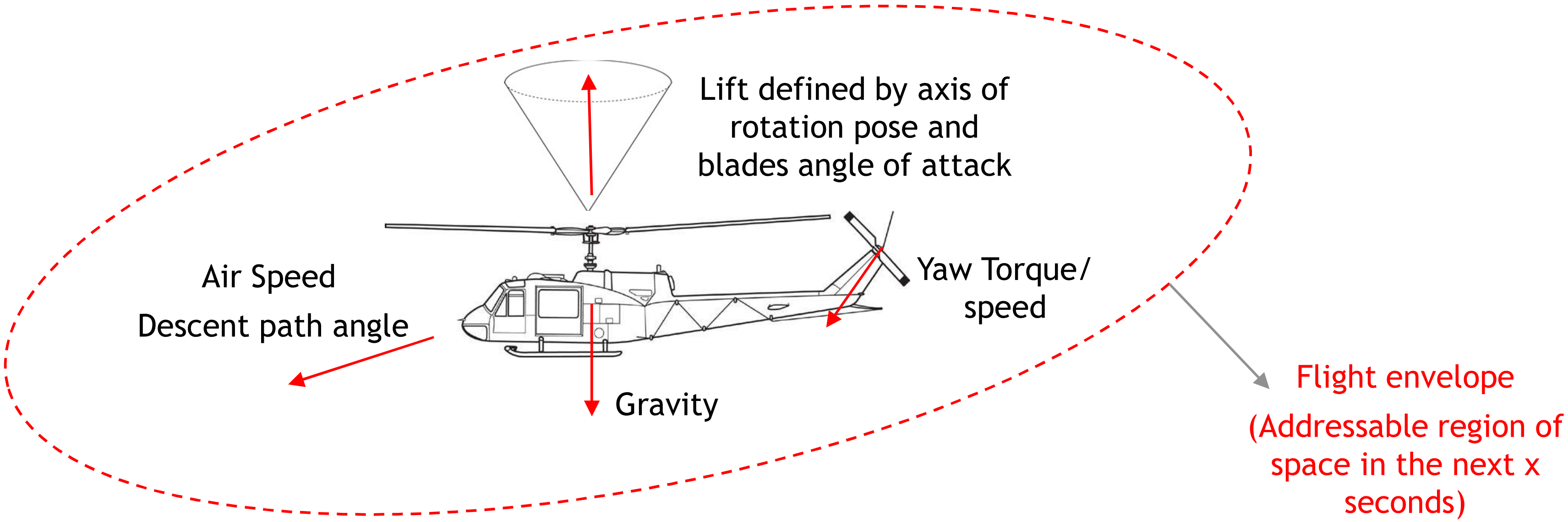
First attempt at 3D mapping for slowly moving obstacles



Take aways

- **Objective of test:**
  - Test octomaping algorithm for moving obstacles
- **What we learned:**
  - Octomapping algorithm fails (ghost generated when object moves)
- **Next steps:**
  - Optimize parameters for real time detection
  - Be able to extract only the relevant information

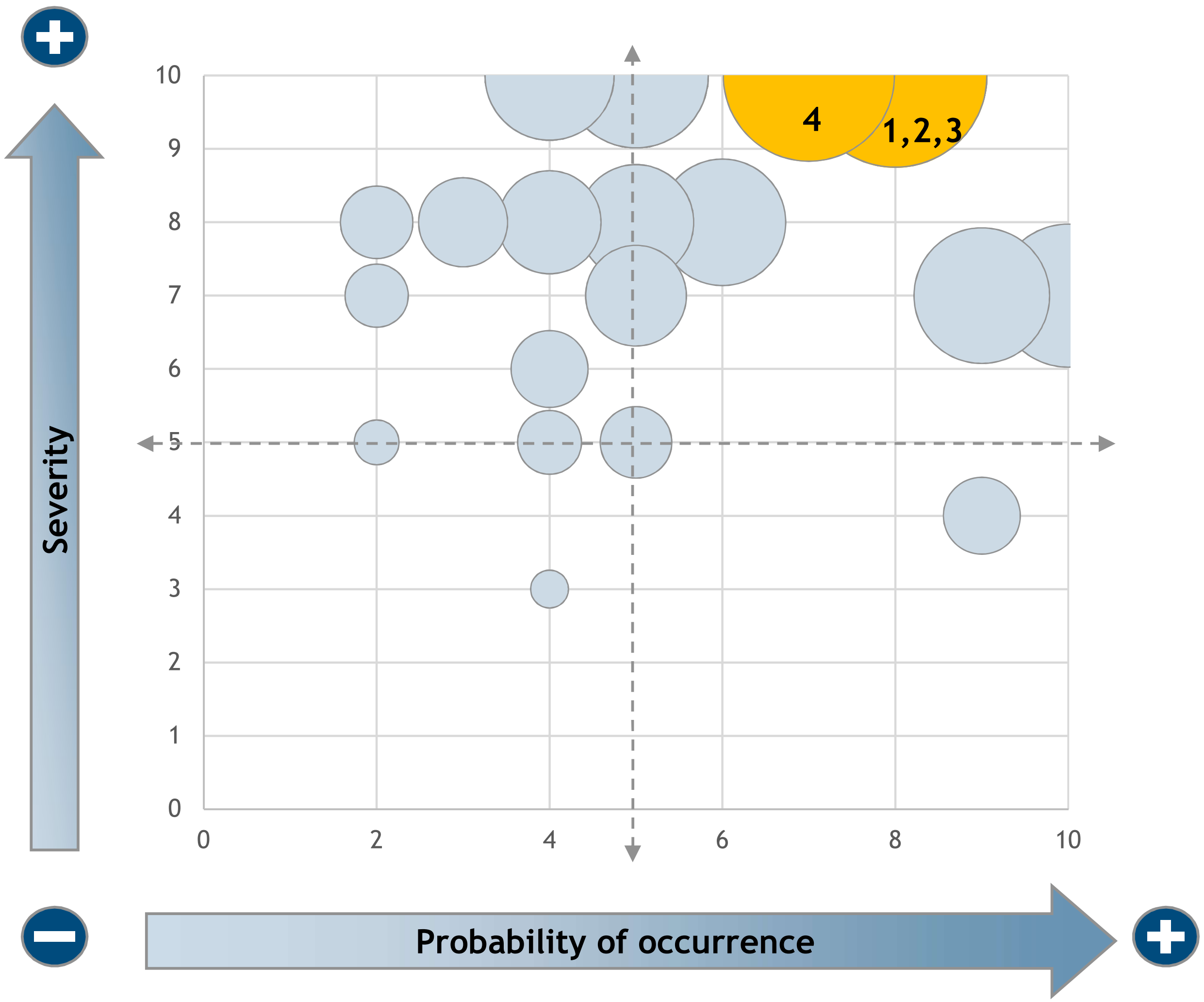
The dynamics of the vehicle together with the current pilot input will be used to adapt the point cloud window to show in the bird's eye view



Pilot interface	Controls	Direction	Pilot input controls	Helicopter reaction
Left lever	**Collective pitch**	Pith	Main Rotor: Blades angle of attack	Move up/down Decelerate/Accelerate Horizontally
	Throttle/Torque	--		Self-adjusted o maintain RPM at Main Rotor
Right lever	**Longitudinal cyclic pitch**	Pitch	Pose of Main Rotor Rotation Axis: Forward/backwards	Nose pitches down/up Move forward/backward
	Lateral cyclic pitch	Roll	Pose of Main Rotor Rotation Axis: left/right	Nose rolls left/right Move sideways
Anti-Torque pedals	**Collective pitch of the tail rotor**	Yaw	Increase/decrease torque and engine speed of tail rotor)	Heading control (left/right)

Preliminary

We are currently deploying mitigation actions with owners to address the key risks identified



Three major risks identified

1. Octomap (Shivang/Hari)

- Issue: Octomap cannot process LiDAR data real time
- Mitigation: Increase onboard computer processing power/lower map resolution

2. AR Headset (Nihar)

- Issue: AR headset not robust to light changes
- Mitigation: Select another AR headset

3. UI/UX voice commands (Joao)

- Issue: Voice commands do not work
- Mitigation: Bird's eye pops-up automatically

4. Flight hardware procurement (Joao)

- Issue: too much time to make drone test
- Mitigation: Use NEA drone (from start)

Note: 29 risk identified so far

Backup

Tentative

Objectives tree

Provide Enhanced Situation Awareness to pilots using AR

- Low visibility
- Landing maneuvers
- Cluttered environments
- (..)

Enhance visibility	Provide real time information	Keep pilot close to reality
Rear View	Provide standard sensor information (ATT, ALT,…)	Feel “natural” to the pilot
Top View	Plot simplified path to intended destination	Ensure uncluttered view of (relevant) reality at all times
Bottom View	Localize and map surrounding obstacles	Frame refresh rate faster than human perception
Front View (e.g. infrared)	Identify safe flight envelope[1] (Green, Yellow, Red)	All computation resources are onboard the helicopter

[1] Based on the current dynamics, input control from pilot and standard vehicle operating constraints

AR = Augmented Reality

Functional Requirements

Feature	The system SHALL
Input 1	• Receive visual data • Receive sensor state variable data • Receive pilot control input
Process / Plan 2	• Filter sensor data and visual data • Fuse sensor data/state estimation • Compute flight envelope • Localize and map surrounding obstacles • Compute path to destination • Segment environment for safe flight envelope
Output / Convey 3	• Convey state variables (heading, attitude, altitude,....) • Convey rear, lower, front and side views • Convey safe flight envelope • Display path to destination

Note 1: Sensor data includes all standard state variables measured in an helicopter (heading, attitude, altitude…)

Note 2: Sensor data processing to be adapted to the quality of the signals received from the helicopter sensors.

Tentative

Performance Requirements (focused almost exclusively on the output functional requirements)

Feature		The system WILL HAVE
Setup time (headwear)		• < 2 minutes (including strapping, booting and user setting)
Refresh rate	3	• 25 Hz (typical frame rate for human eye perception, standard used in movie theatres)
Latency	3	• <.4 s for flight envelope information • <.2 s for real time world feeds
Resolution	3	• Greater than 360p
Overlay quality	3	• Overlay accuracy less 5 degree

Note: Intermediate stages performance KPIs to be detailed during the design and engineering phase.

Non-functional requirements

Segmentation	The system WILL
Installation	• Easily integrate with helicopter interfaces • Leverage pre-existing helicopter sensors • Be easy to setup (hardware and software) • Be modular by nature
Interaction with Pilot	• Be predictable • Feel natural to the pilot • Be easy to put/remove headwear • Be comfortable to wear headwear for long periods of time
Information Displayed	• Be clear and simple • Be non intrusive to the pilot • Be non distracting for the pilot
Other criteria	• Be substantially more affordable than available solutions (e.g. fighter jet pilots) • Be closely align virtual information with the real world