Drone flights data planning and processing

Team: “Neon Genesis Tetrahedron”

Captain: Dmytro Kolisnyk
Danylo Levoshko
Andriy Agarkov

Background

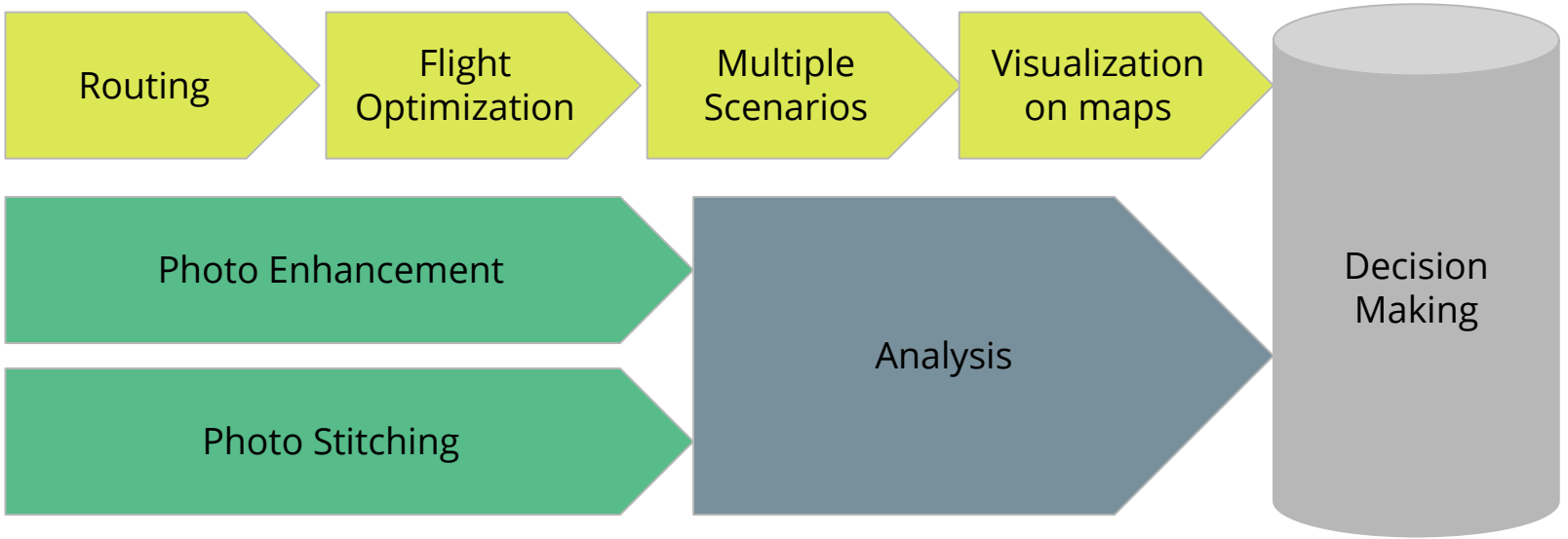
The drones' creation and use have become one of the major breakthroughs in world intellectual achievements. Initially, drones were associated exclusively with the military industry, but now their scope is expanding, technologies are becoming more functional and affordable.

At present drones are widely used in express delivery, journalism, medicine transportation, saving lives, observing wildlife, combating crime, assistance in catastrophes, helping farmers, etc.

Prerequisites in Ukraine

Using drones in agriculture is one of the promising areas for Ukraine, which can help to increase productivity and reduce agribusiness costs. Agriculture is one of the leading sectors of Ukraine's economy - as of 2018, agriculture covers almost 17% of Ukraine's GDP and brings in about 38% of foreign exchange earnings. However, its efficiency can be greatly enhanced. Drones equipped with thermal imagers measure plant growth. Multispectral sensors allow to monitor the application of irrigation, fertilizers, and pesticides in the required places. It can be monitored the health of crops, fungal or bacterial infections.

Problem Decomposition



Design Assumptions

Ability to enter custom Drone characteristics

Ability to choose from predefined Drones

Ability to enter base GPS coordinates

Route Visualization in maps

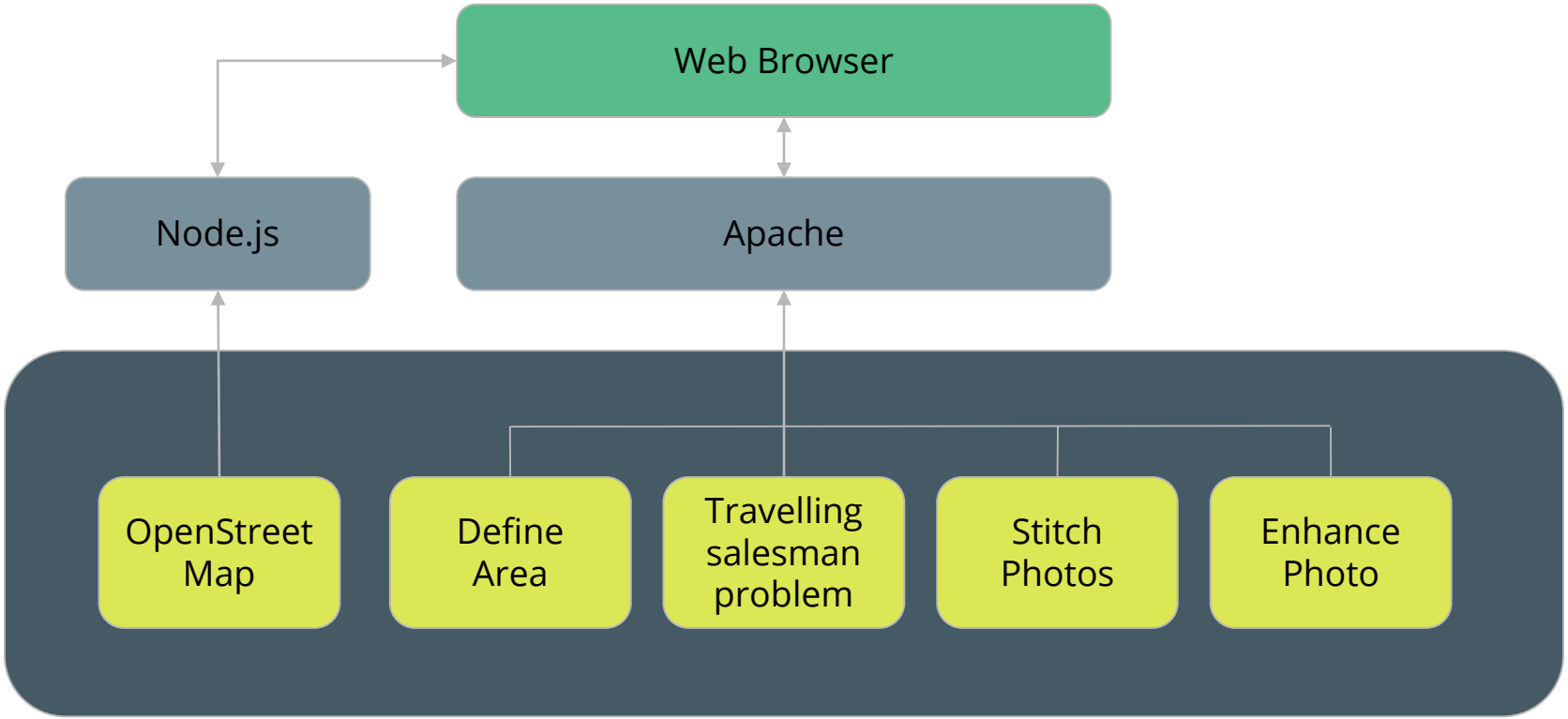
Ability to choose start point from map

Available 24/7

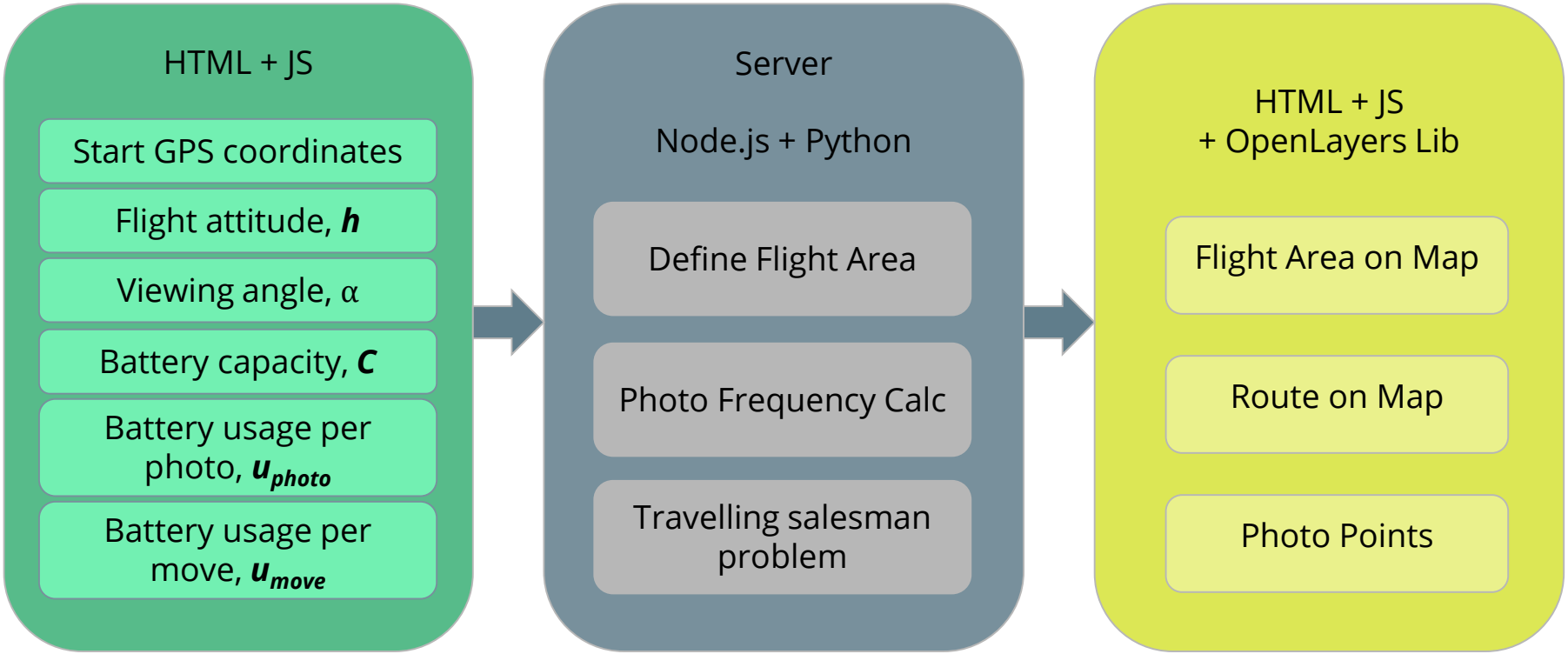
Available from desktops, tablets, mobiles

Cross-platform solution

Application Architecture



Drone Flight Planning + Visualizing



Define Area Model

$$m = l\sqrt{\frac{C - \sqrt{2}lu_{move}}{lu_{move} + u_{photo}}}$$

$$l = 2h\tan\frac{\alpha}{2}$$

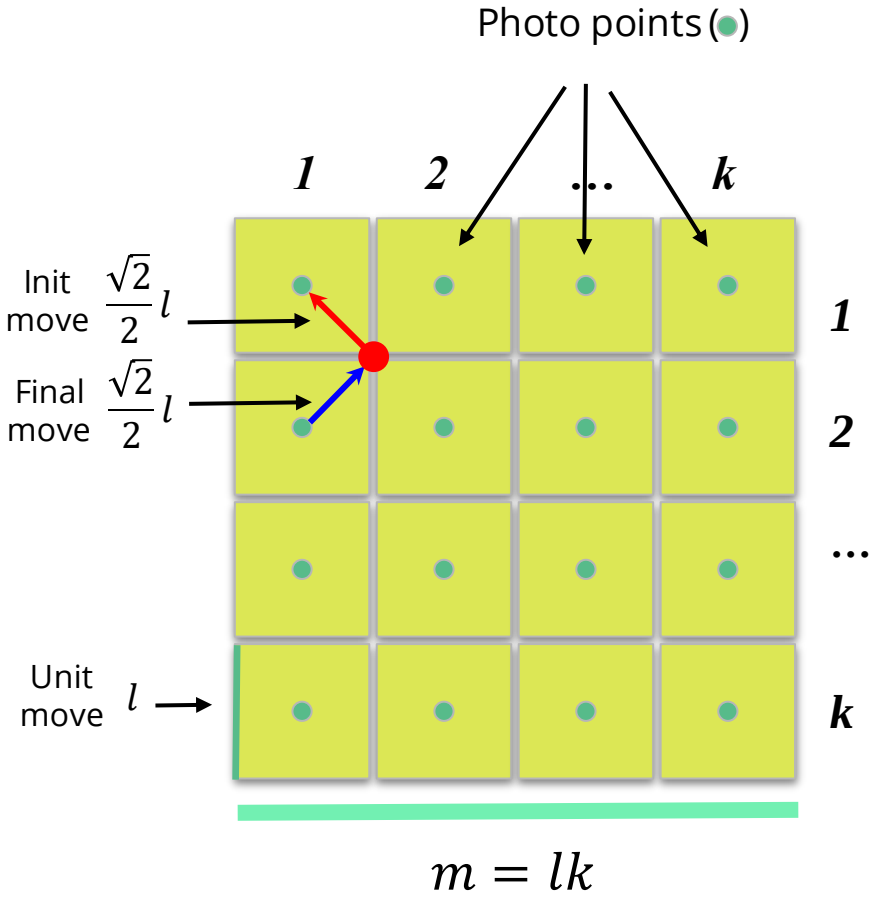
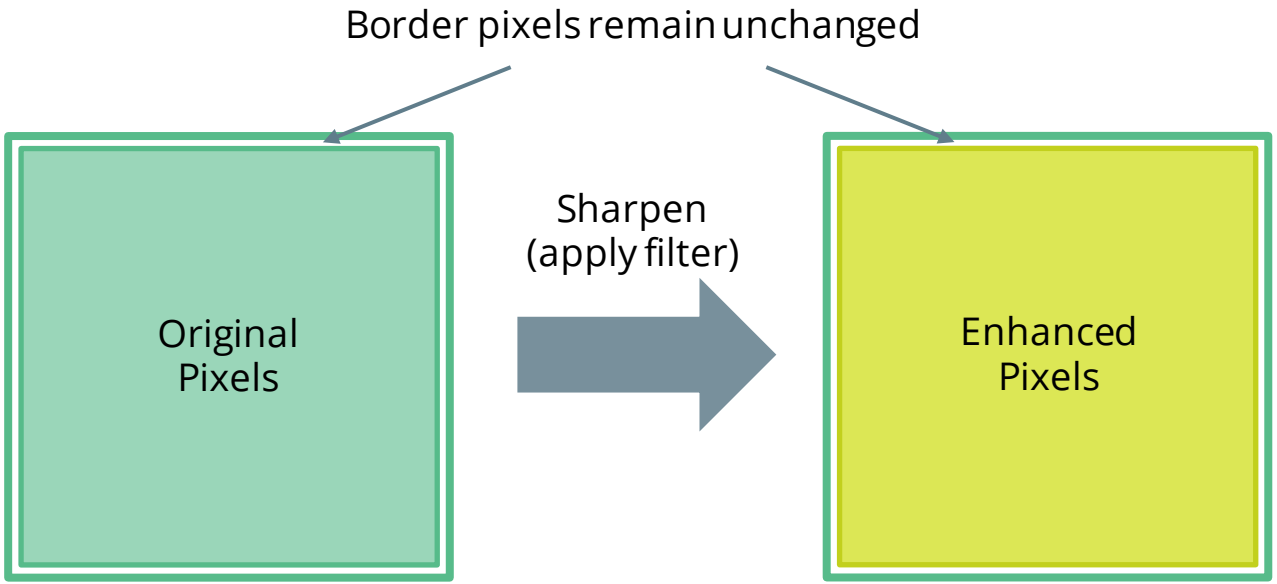
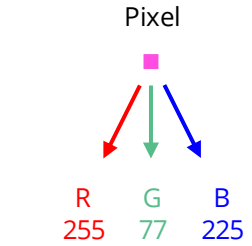


Photo Enhancement





C = {R,G,B} = {255, 77, 225}

$$\begin{cases} R\epsilon[0;255] \\ G\epsilon[0;255] \\ B\epsilon[0;255] \end{cases}$$

Sharpening Algorithm

C_{xy}

C_{x-1y-1}	C_{x-1y}	C_{x-1y+1}
C_{xy-1}	C_{xy}	C_{xy+1}
C_{x+1y-1}	C_{x+1y}	C_{x+1y+1}

m_{ij}

-1	-1	-1
-1	9	-1
-1	-1	-1

$$(RGB)_{xy} \begin{cases} R_{x_i y_j} \cdot m_{ij} \xrightarrow{\theta_R} R'_{x_i y_j} \\ G_{x_i y_j} \cdot m_{ij} \xrightarrow{\theta_G} G'_{x_i y_j} \\ B_{x_i y_j} \cdot m_{ij} \xrightarrow{\theta_B} B'_{x_i y_j} \end{cases} (R'G'B')_{xy}$$

$$\theta_c = \sum_{i=1}^{3} \sum_{y=1}^{3} C_{x_i y_j} \cdot m_{ij}$$

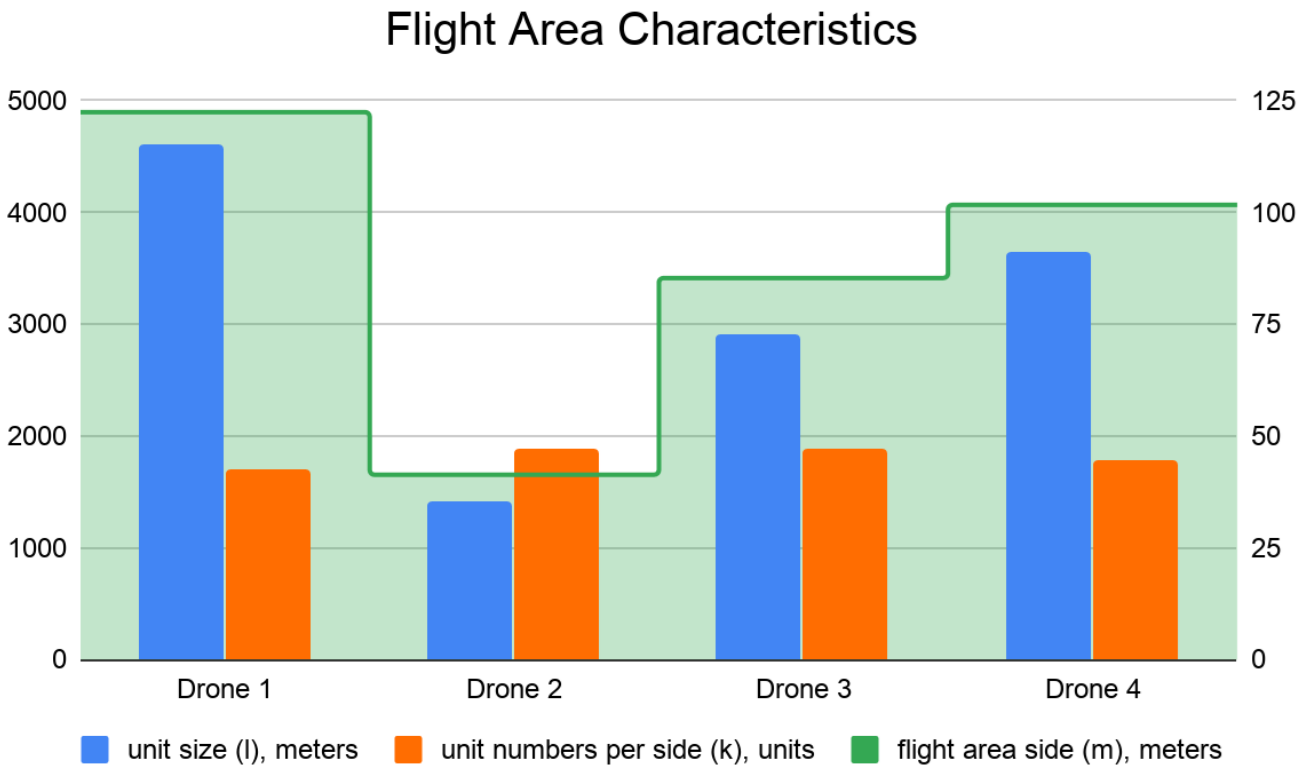
$$\begin{cases} \theta_c > 255 \Rightarrow \theta_c = 255 \\ \theta_c < 0 \Rightarrow \theta_c = 0 \end{cases}$$

C – color, R – red, G – green, B - blue

Photo Stitching - PIL(Pillow) vs OpenCV

PIL (Pillow)	OpenCV
Python Image Library	Mainly aimed at real-time Computer Vision
Conversion between image types	
Image transformation (flip, rotate, warp etc)	
Image filtering (blur, sharpen, etc)	
Easy to use	Loaded with algorithms well-suited for data science and vision-based robotics
Lightweight	Machine learning: neural network, SVM, K-NN, etc
Use when you want to cut and resize images, or do simple manipulation	**Agile tool for more complicated problems like stitching multiple images**

Flight Area Scenarios



Flight Planning Result

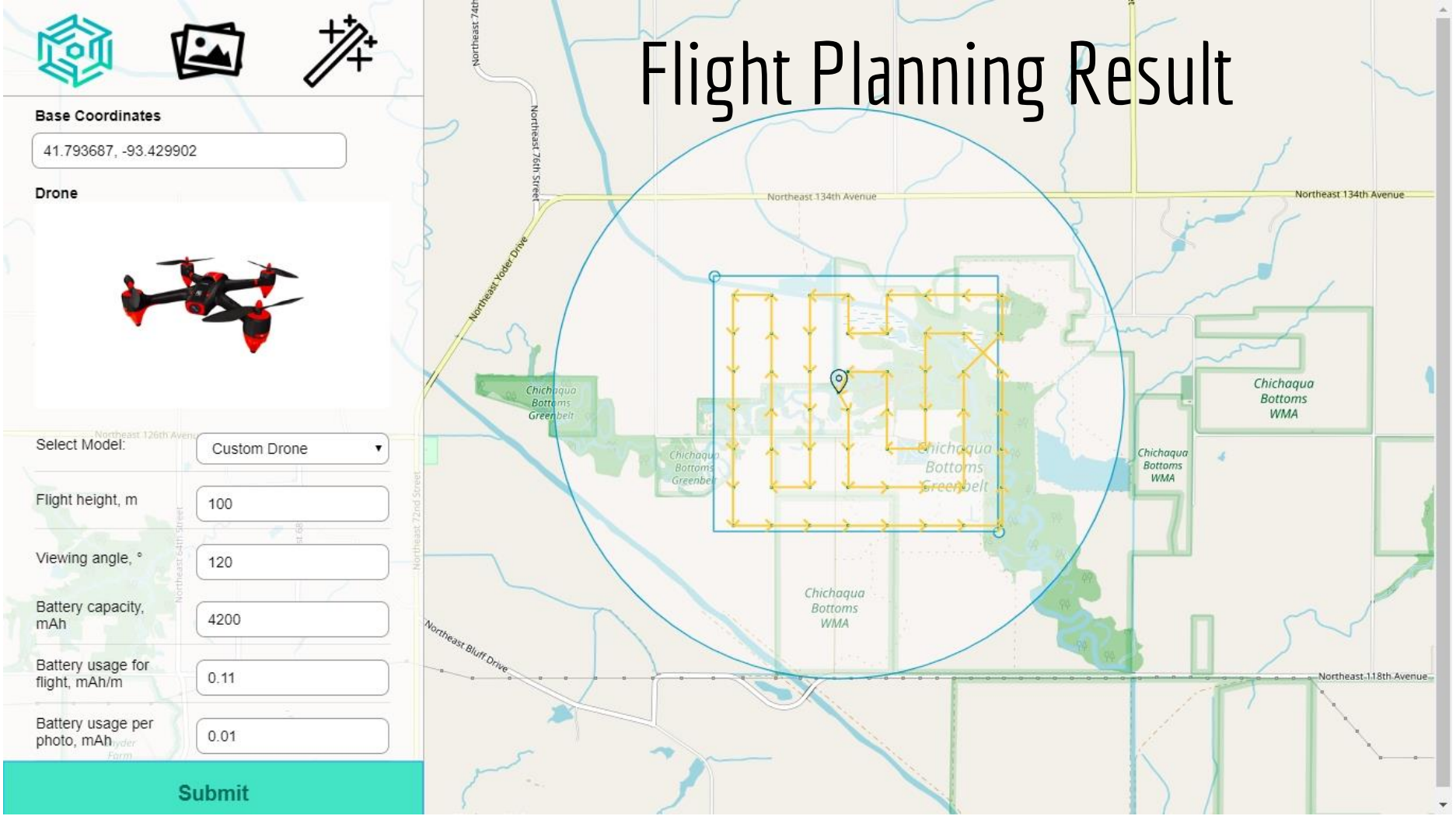
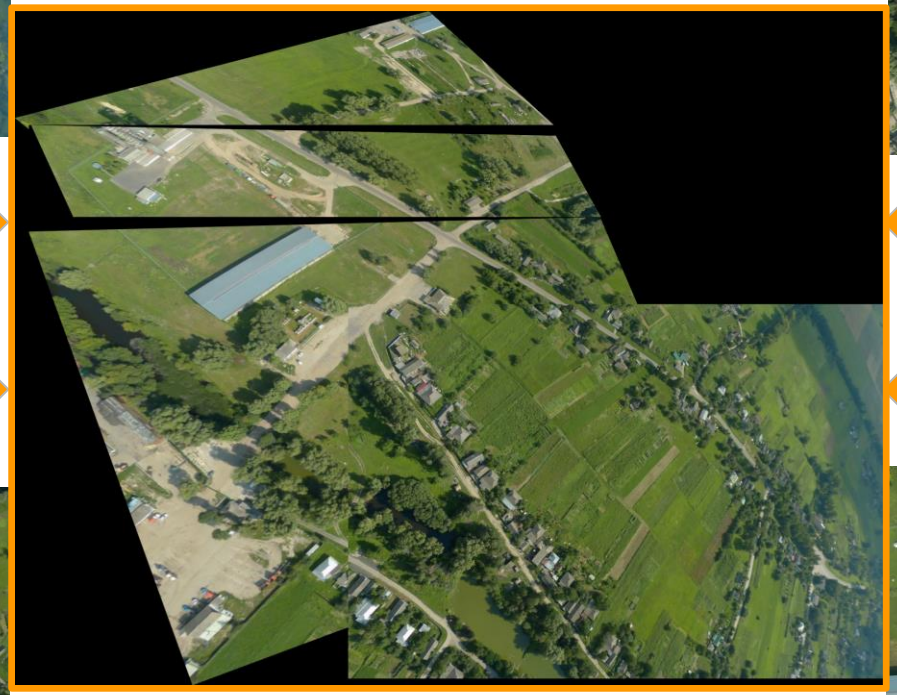


Photo Enhancement Result

Before **After**

Photo Stitching Result



Next Steps

- Crop Analysis Module
  - Gathering data for various crops in different conditions (to be able to compare with actual photo and make a decision regarding watering, fertilizing etc)
  - Using thermal sensors, multispectral images analysis
- Cost-Benefit Analysis Module
  - Financial efficiency calculation
  - Dashboard with tables and charts with recommendations (needs identification, observations frequency, costs etc)
- Using in other areas
  - Search & Rescue
  - Nature observations and research

Team members contribution

Dmytro Kolisnyk	• Application Architecture Design • Flight Planning + Visualizing Module • Photo Enhancement Module • Presentation
Danylo Levoshko	• Front End Design • Photos Stitching Module
Andriy Agarkov	• Travelling Salesman Problem algorithm implementation