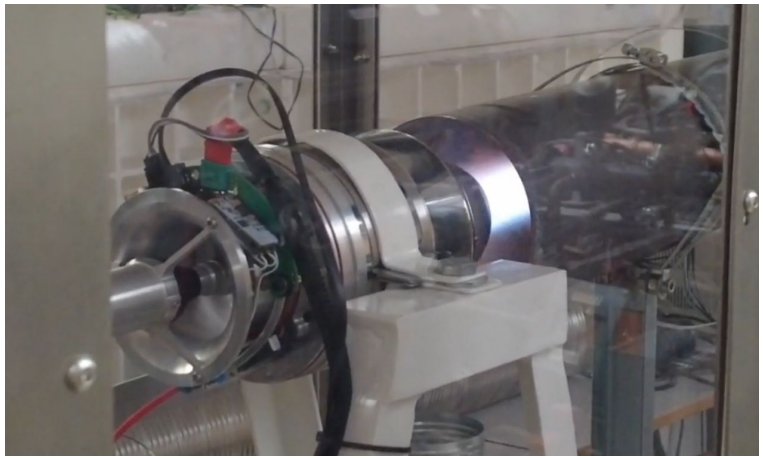
Wizyjny system namierzania i śledzenia obiektów latających

Jakub Karbowski
Wydział Informatyki, Elektroniki i Telekomunikacji
Kierunek Informatyka

AGH DRONE ENGINEERING

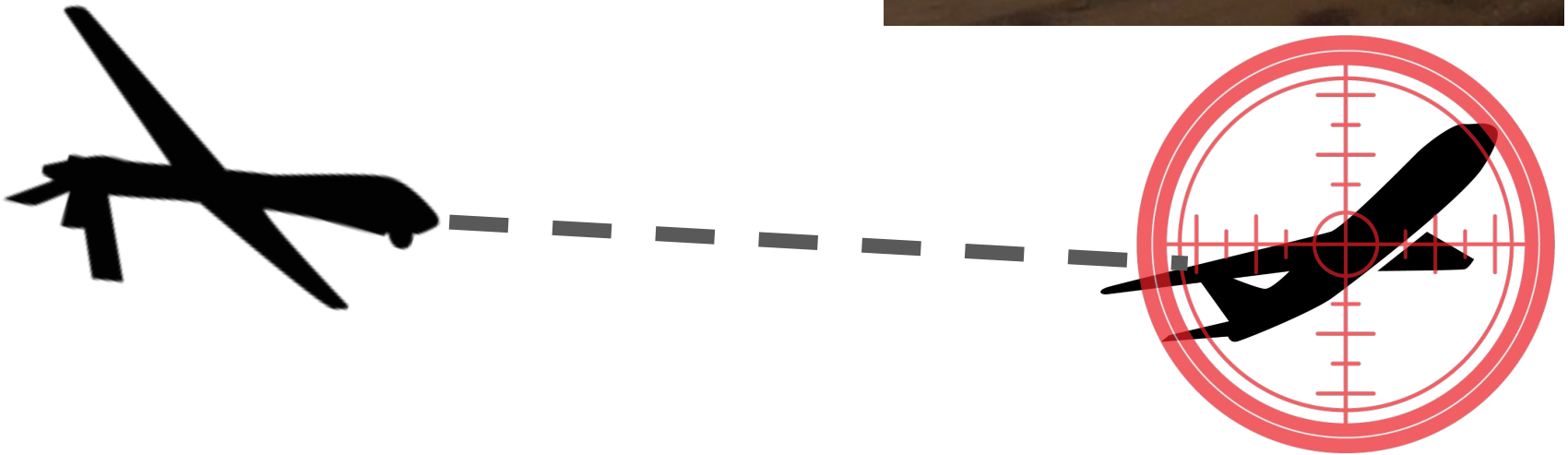
drone.agh.edu.pl

facebook.com/AGHDroneEngineering



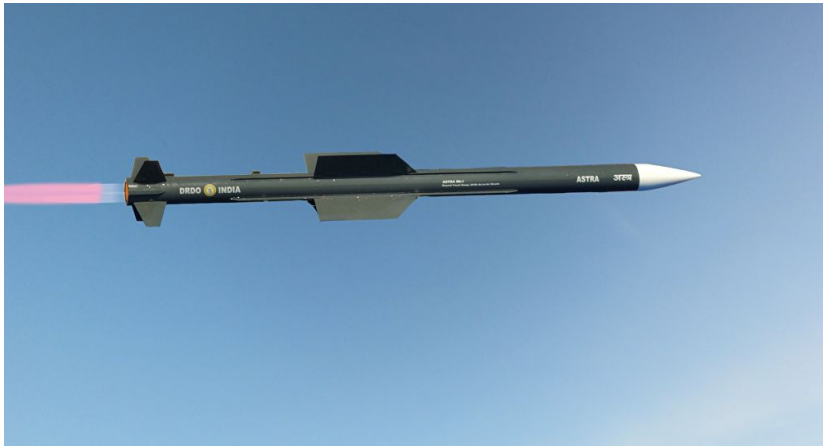
Projekt Hawkeye

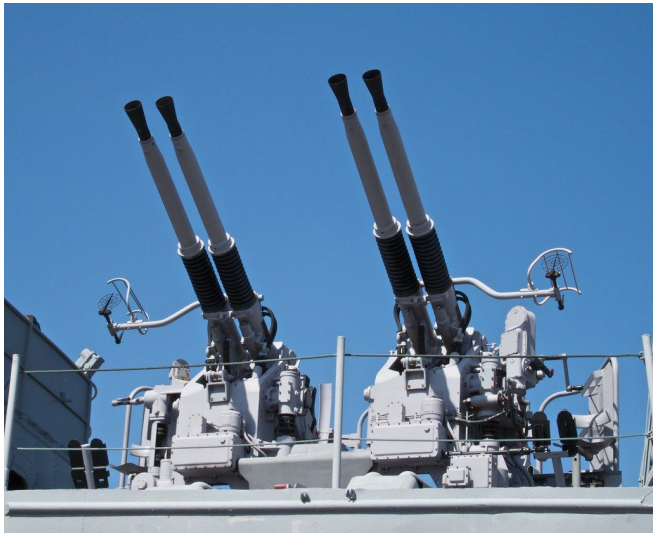
- System namierzania
- Automatyczna akwizycja celów
- Utrzymywanie śledzenia podczas gwałtownych manewrów



Zastosowania

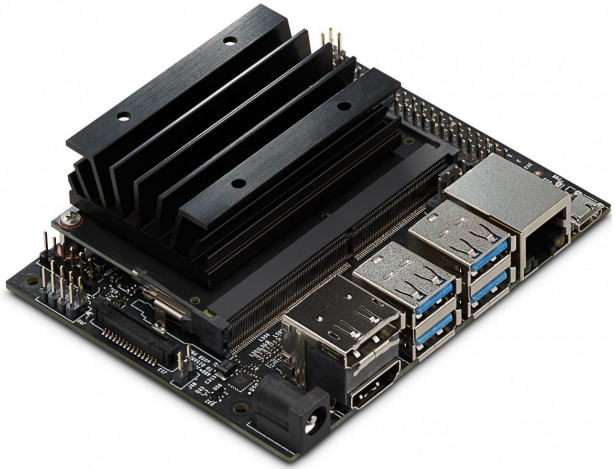
- Autonomiczne myśliwce
- Systemy wspierania pilota
- Rakiety naprowadzane
- Autonomiczne działa przeciwlotnicze





Wymagania

1. Wydajność - śledzenie szybkich (i uciekających) celów
2. Kompaktowość
3. Praca w trudnych warunkach



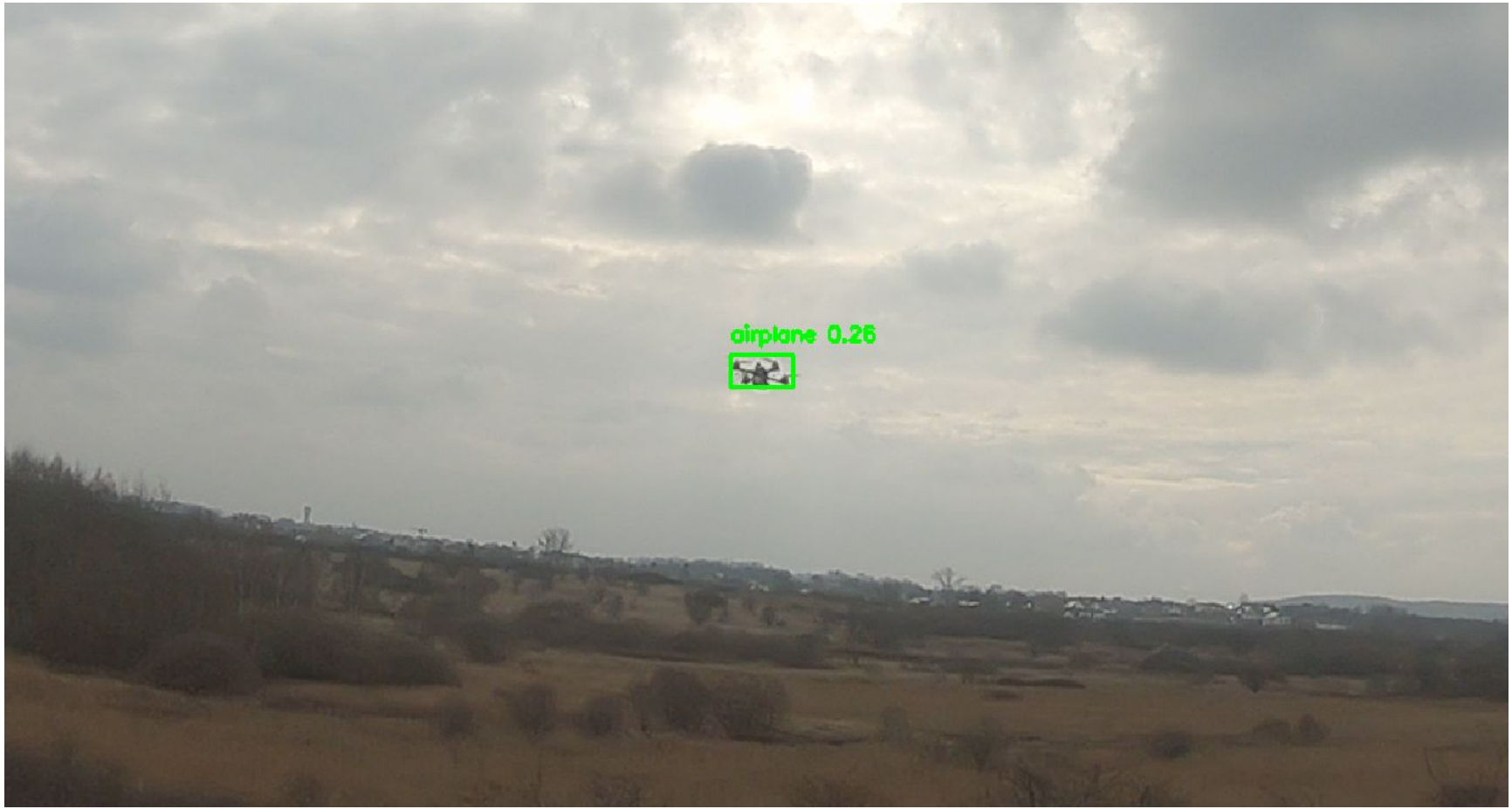
Nvidia Jetson

Chmury zasłaniające cel

Możliwe rozwiązania

Rozpoznające wygląd	Rozpoznające ruch
“To wygląda jak potencjalny cel”	“To rusza się jak potencjalny cel”

Sieci neuronowe - detekcja obiektów



Rozpoznawanie wyglądu

- Prosta, efektywna implementacja: sieci neuronowe.
- Nie śledzi tego, czego nie rozpoznaje.

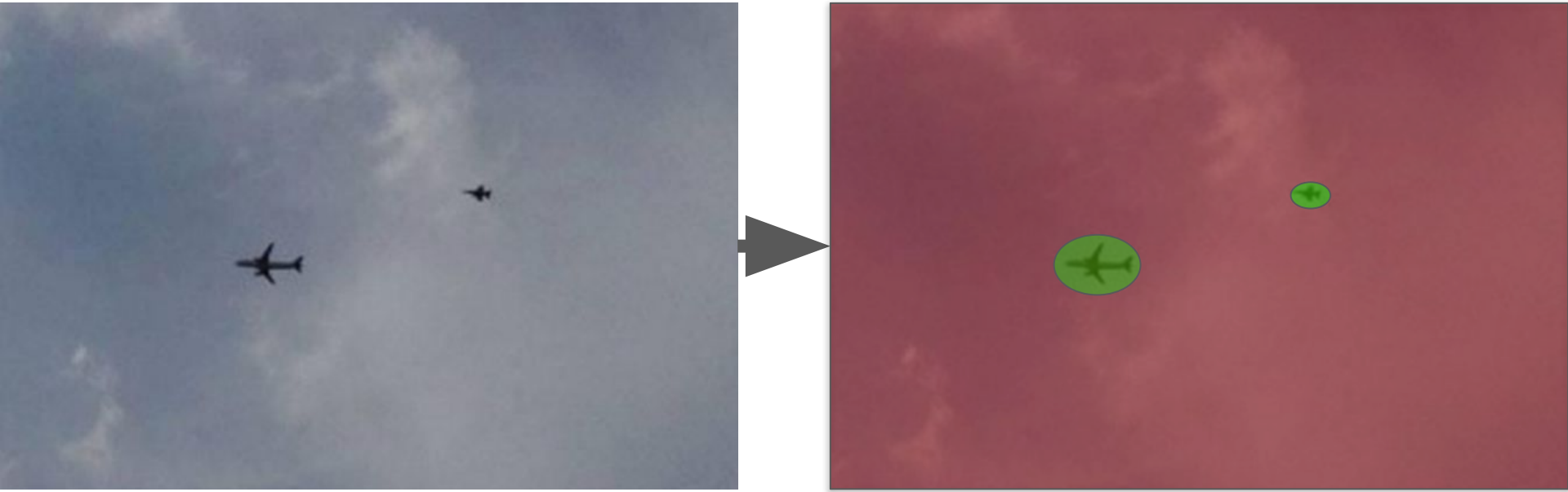
Czy na pewno tak musi być?

Gdzie jest dron?

Porównując kolejne klatki można zaobserwować ruch obiektów

Rozpoznawanie ruchu: idea algorytmu

Chcemy wyróżnić obszary ruchu na statycznym tle



Jak zdefiniować “tło”?

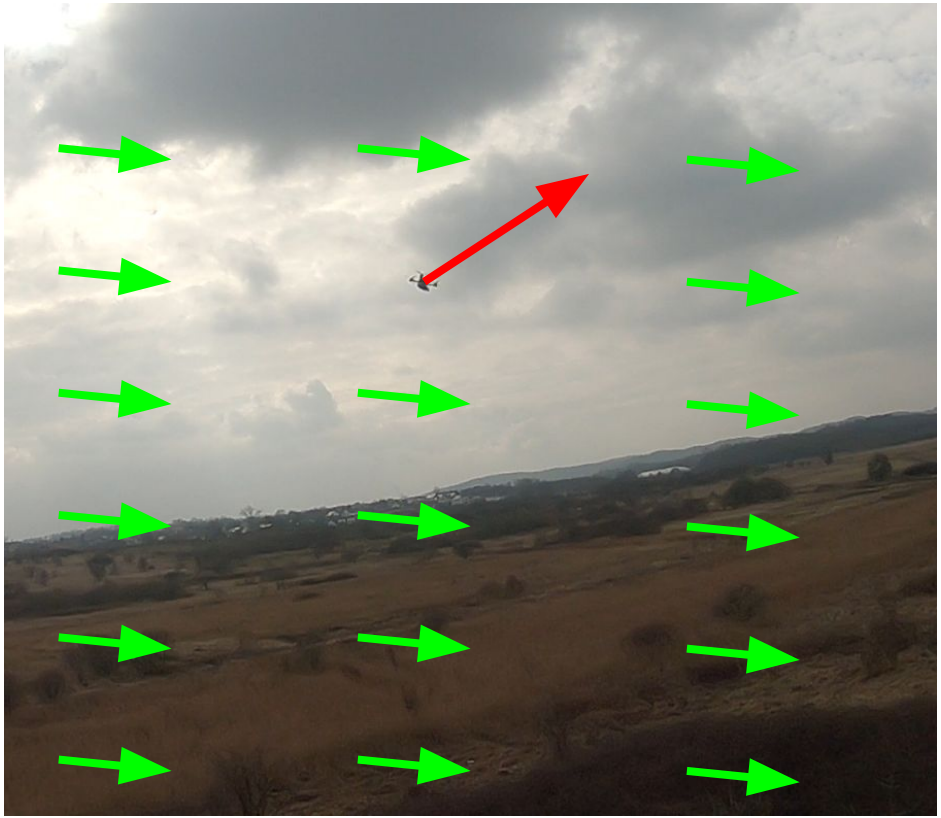
Przyjmujemy tło jako największy spójnie poruszający się obszar obrazu.

Przykład:

Nasz pojazd skręca w lewo, tło przesuwa się w prawo.

Cel przemieszcza się względem tła.
Możemy to rozpoznać.

Ruchomym obiektem jest wszystko, co rusza się inaczej niż większość obrazu.



Wektory pola ruchu na obrazie

Co jeśli cel rusza się zgodnie z tłem?

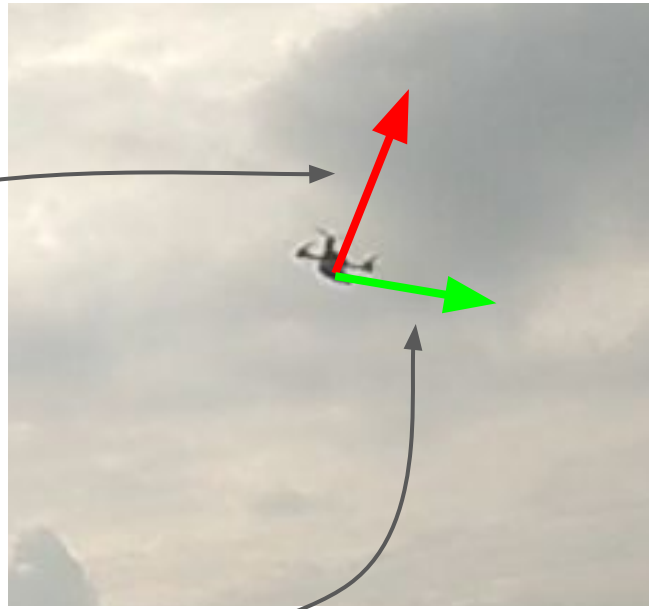
- Problem staje się nierozstrzygalny.
- Nie można określić czy obiekt jest częścią tła bez dodatkowej wiedzy (klasyfikacja na podstawie wyglądu).

Rozwiązanie:
Powrót do algorytmu rozpoznającego wygląd kiedy nie mam informacji o ruchu

Wiemy, że to dron.
Ale nie wiedząc czym jest dron, nie wiemy czy nie jest częścią nieba

Metoda wykrywania ruchomych obiektów

1. Stwórz model opisujący ruch tła.
2. Porównaj faktyczny ruch każdego piksela z oczekiwanym ruchem wyliczonym na podstawie modelu tła.



Taki ruch został zmierzony

↓

Niezgodność => to jest ruchomy obiekt

↑

Tak “powinien” ruszać się punkt

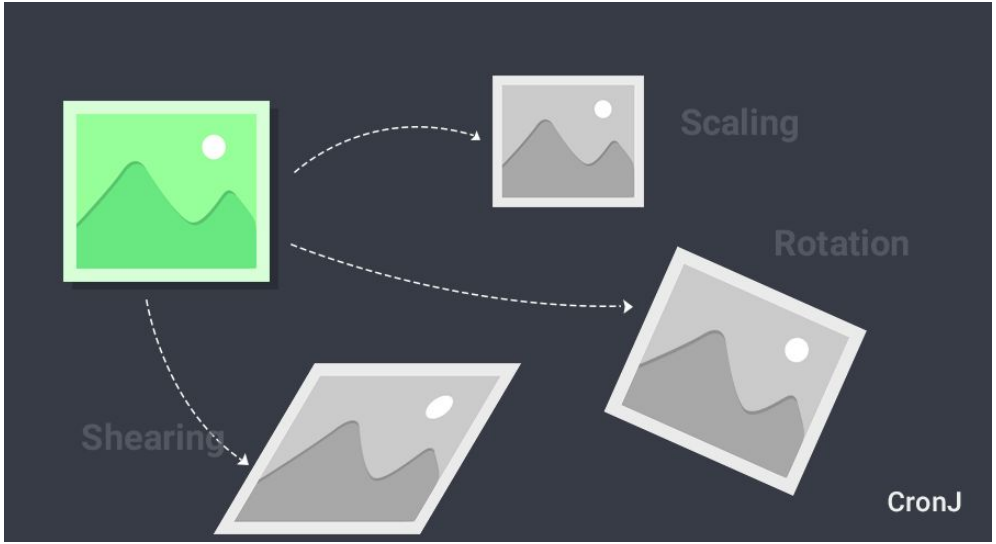
Modelowanie ruchu tła

Uproszczenie

Tło znajduje się daleko => Można przyjąć transformację afiniczną

Dla klatek blisko w czasie i odległego tła model sprawdza się dobrze.

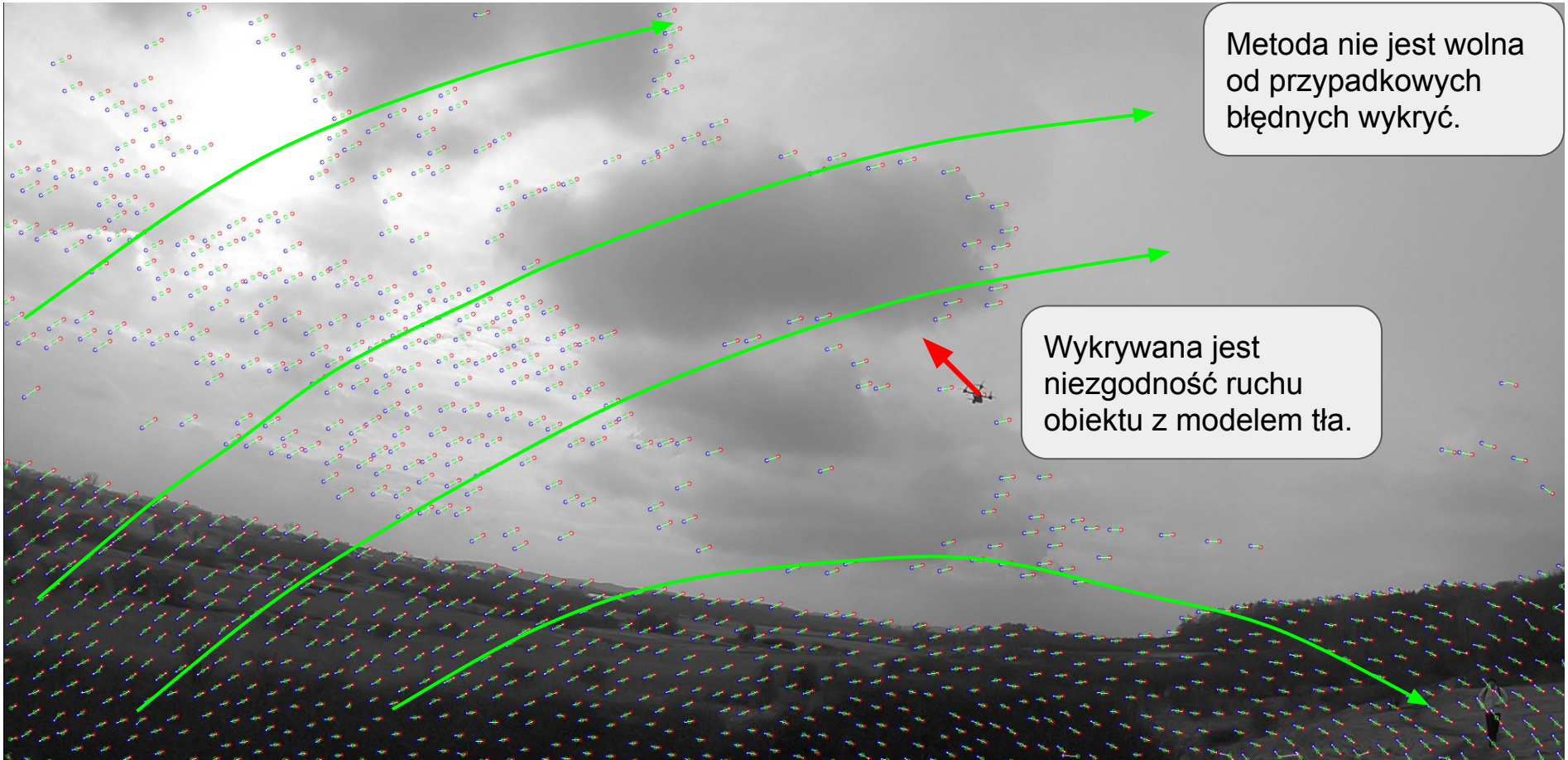
Model można uogólnić do homografii (więcej stopni swobody).



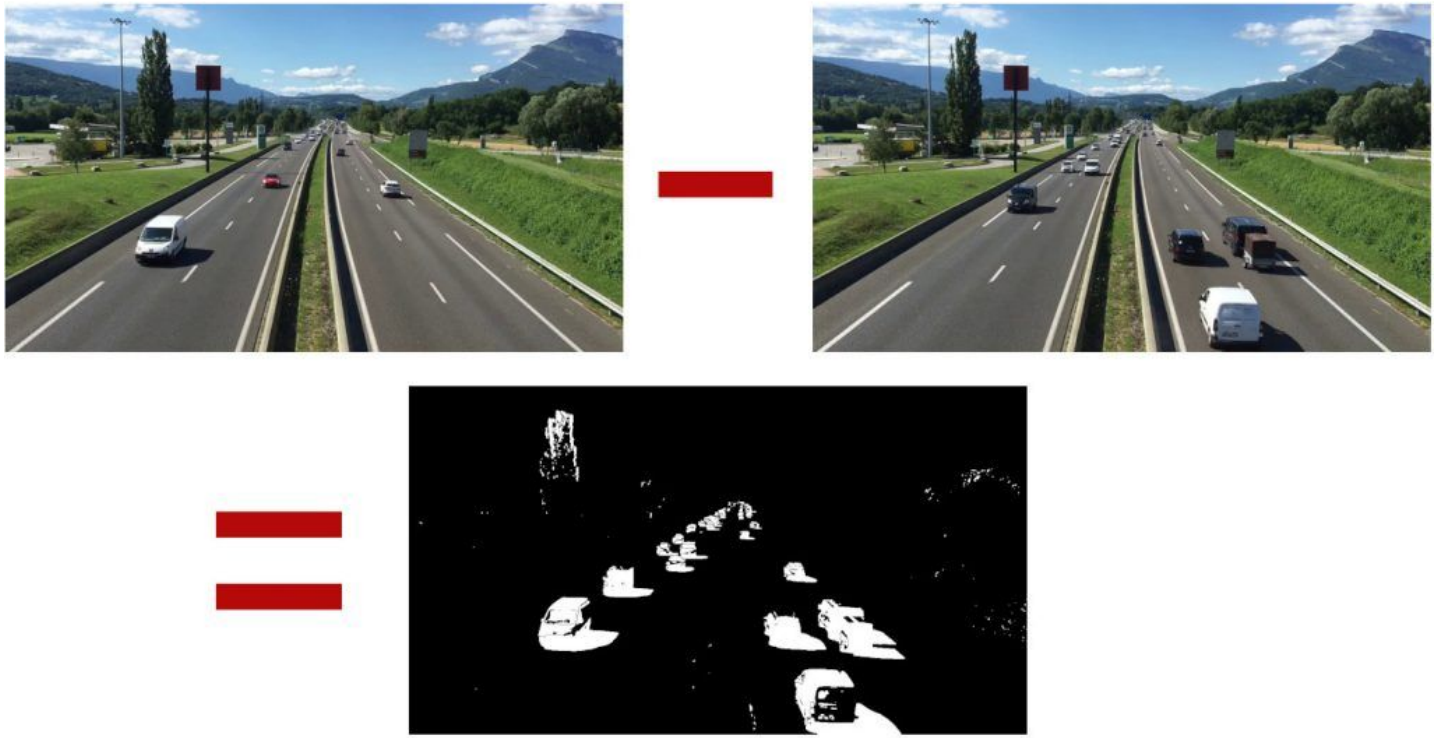
Modelowanie ruchu tła

1. Szukamy charakterystycznych punktów (metoda Shi-Tomasi).
2. Liczymy ich przesunięcie względem poprzedniej klatki (Lucas-Kanade optical flow).
3. Chcemy dopasować model afiniczny/homograficzny na podstawie par punkt-punkt.
4. Estymacja transformacji - metoda RANSAC:
    a. Model jest obliczany na podstawie losowo wybranego podzbioru punktów.
    b. Statystycznie, będą to punkty z tła (stanowią większą część danych).
    c. Spełnione zostaje poprzednie założenie: tłem jest największy obszar.
5. Uzyskujemy model ruchu obliczony na podstawie dominującego obszaru.
6. Model wykorzystujemy do detekcji anomalii ruchu.

Algorytm w praktyce



Inna metoda: Background Subtraction

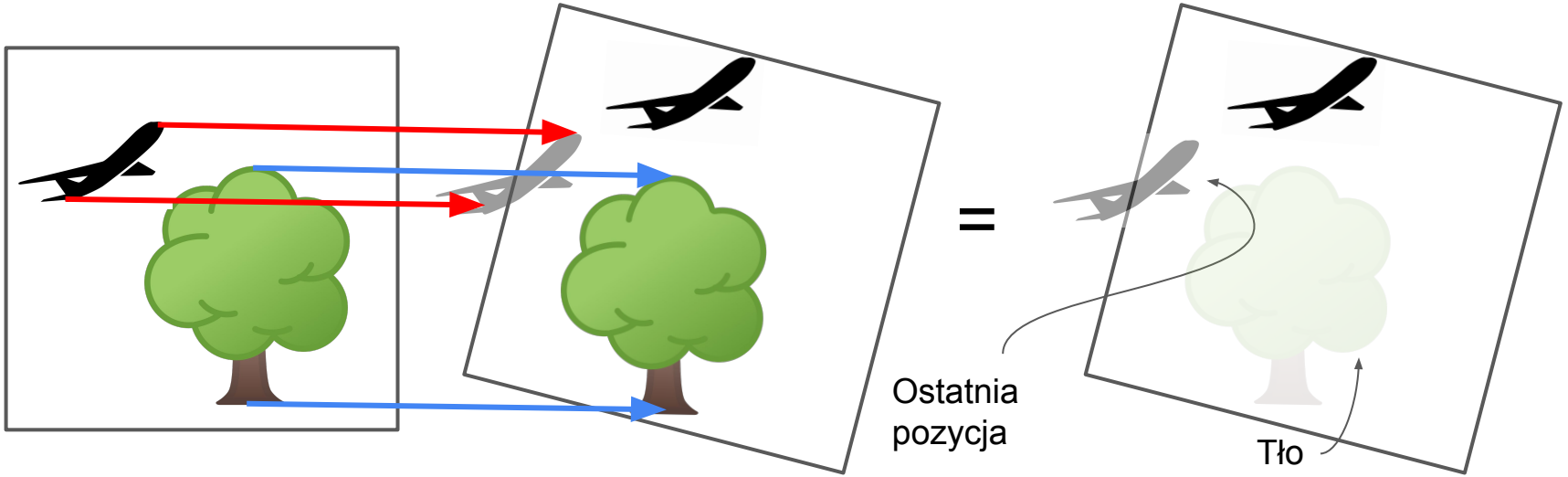


Działa dla statycznej kamery. Problem z obsługą ruchomej kamery.

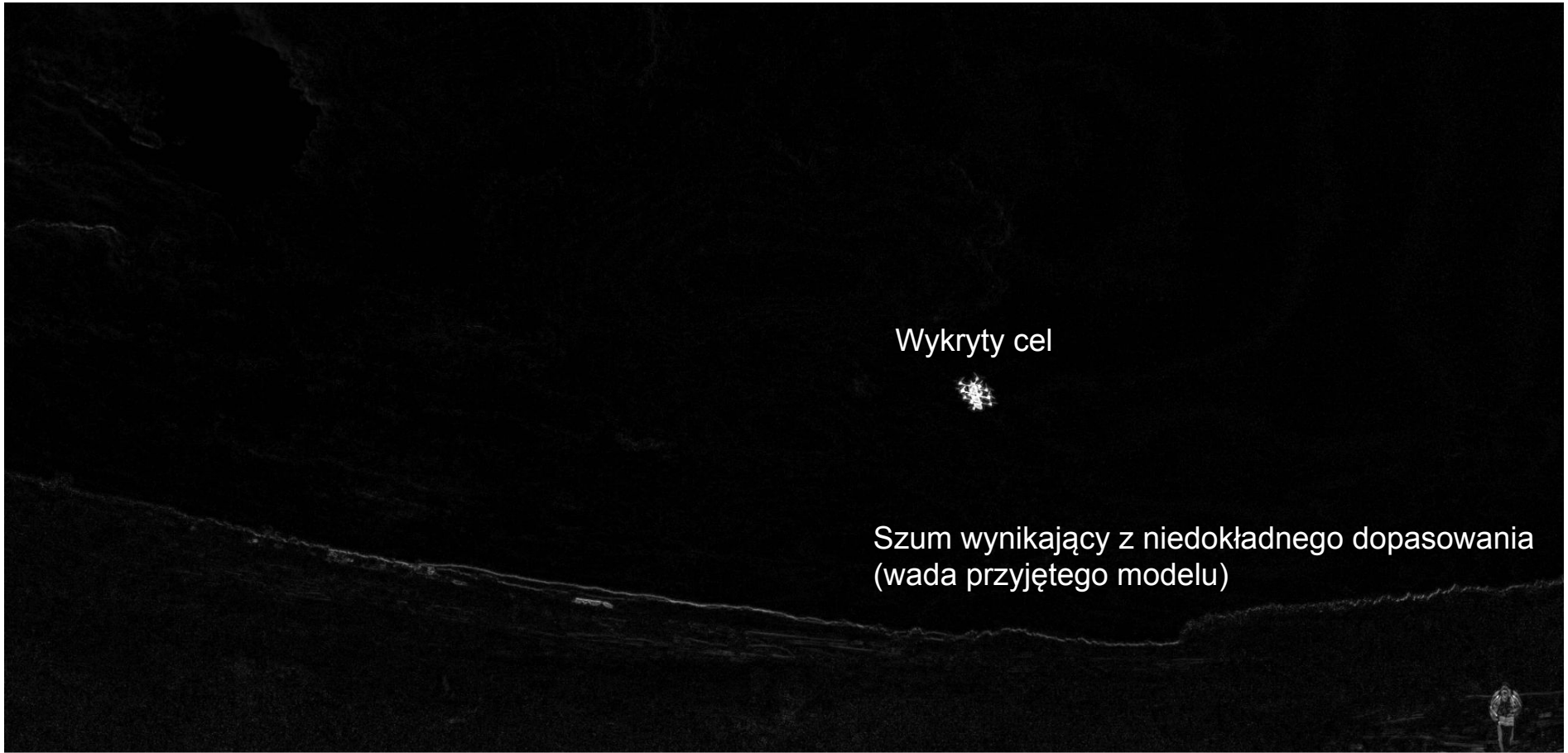
Background subtraction dla ruchomej kamery

Stosujemy algorytm obliczania afinicznej transformacji tła:

1. Obliczamy model ruchu dla tła.
2. Nakładamy dwie klatki na siebie za pomocą obliczonego modelu.
3. Odejmujemy nałożone obrazy aby otrzymać maskę tła.

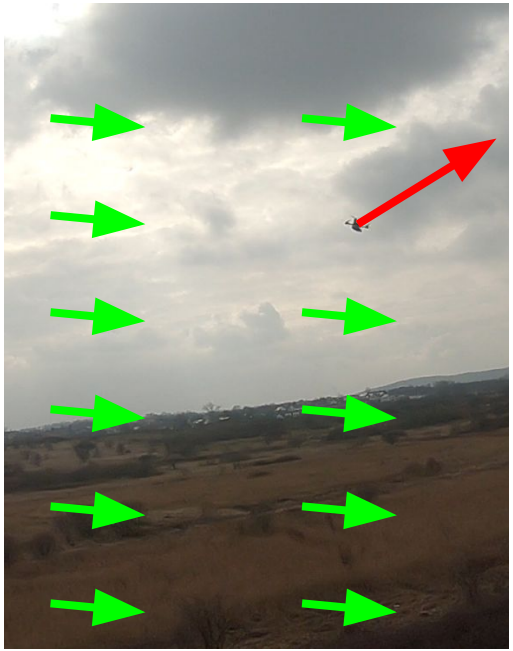


Background subtraction w praktyce

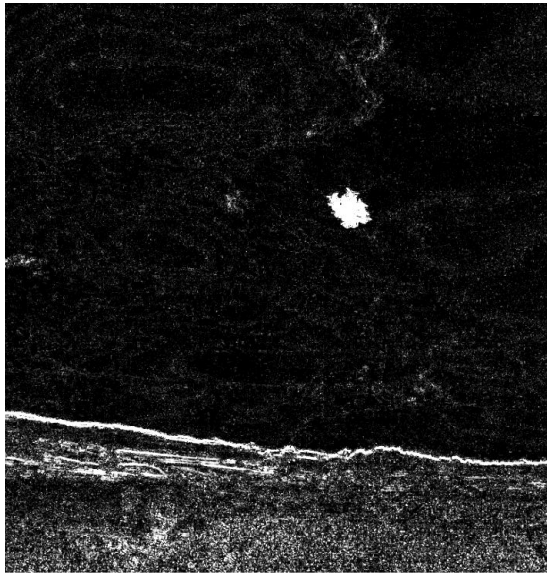


Połączenie obydwu metod

Sprzeczności pomiędzy modelem ruchu tła a prawdziwym ruchem są sprawdzane tylko dla punktów z maski tła metody Background Subtraction.



x



Punkty z tła zostały wykluczone.

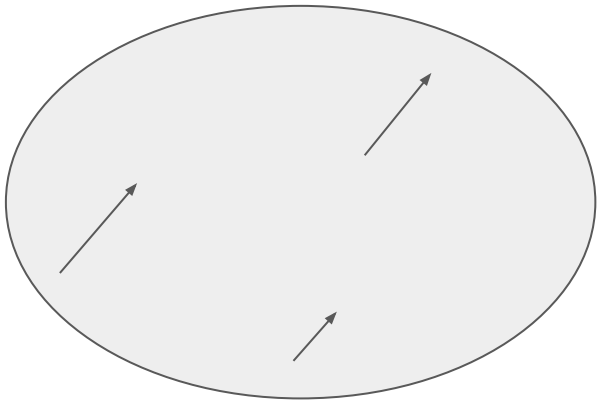
=

Filtracja

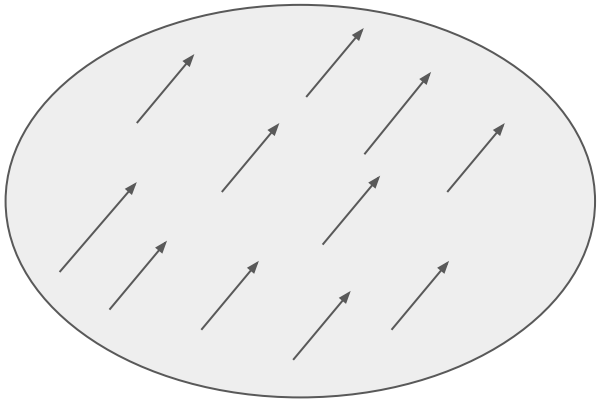
1. Wykryte punkty zbieramy w klastry.
2. Dla każdego klastra sprawdzamy warunki:
   a. Gęstość punktów > threshold
   b. Wariancja kierunków wektorów < threshold

Pozostałe klastry definiują bounding box wykrytych celów, które przepuszczane są przez filtr Kalmana.

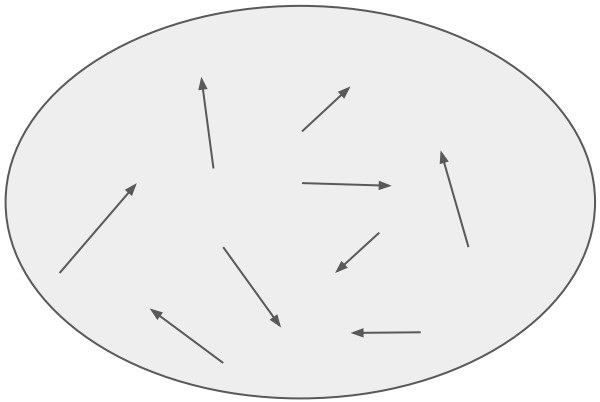
Mała gęstość



Poprawny klaster

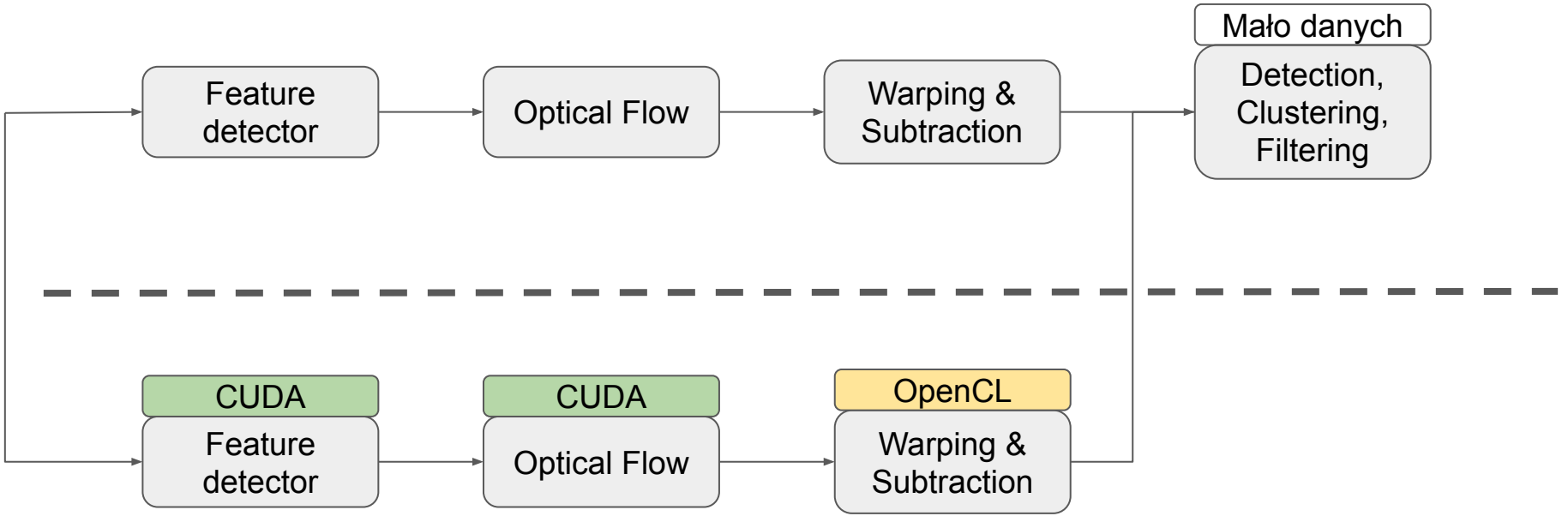


Duża wariancja kierunków



Wydajność

CPU

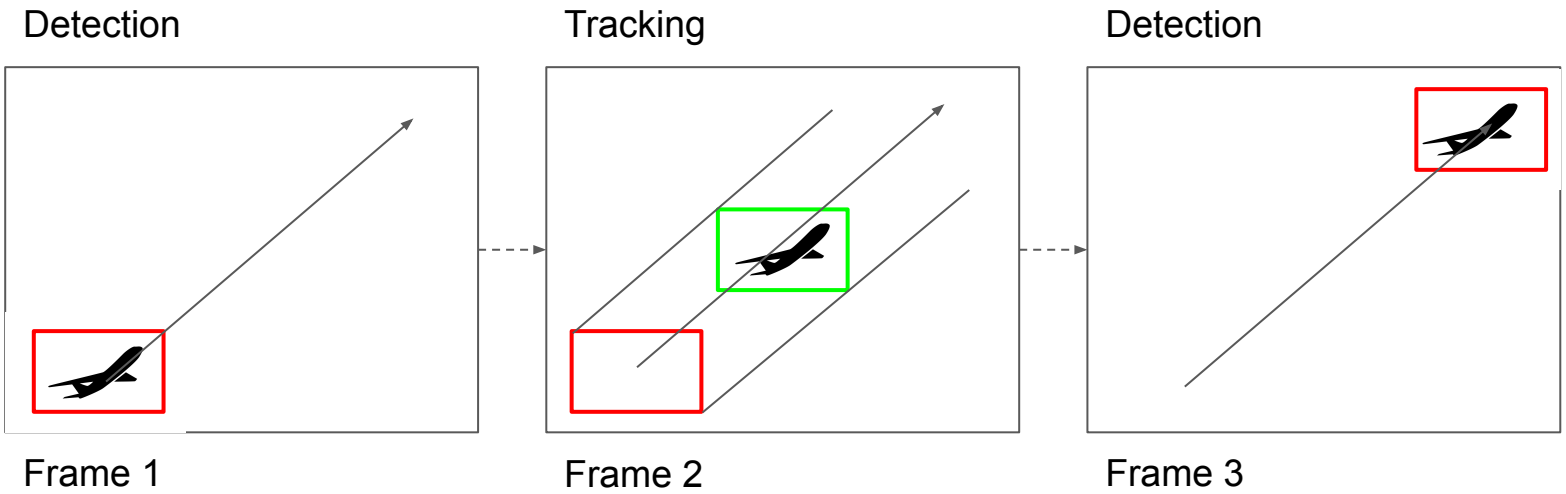


GPU

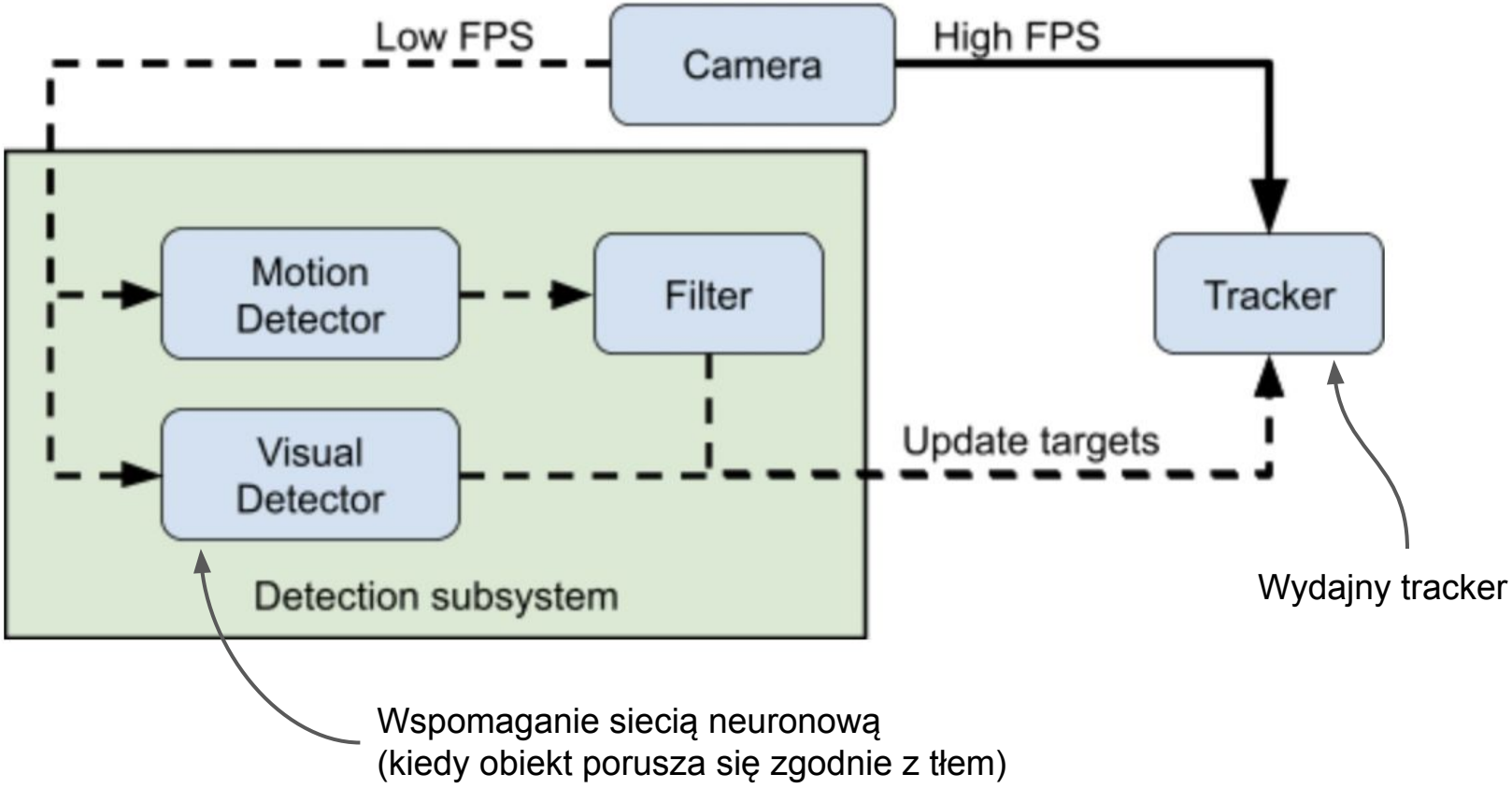
Wspomaganie systemu detekcji

- Algorytm nie musi przetwarzać każdej klatki
- System podzielony na 2 komponenty:
  - Detekcja
  - Śledzenie (łata dziury pomiędzy detekcją)

Tracker jest co pewien czas aktualizowany przez detektor.



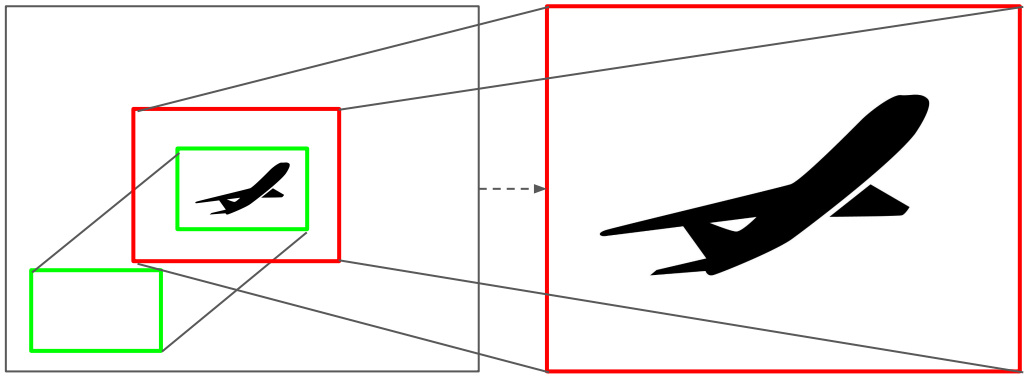
Pełny system



Detekcja za pomocą Region of Interest

- Obiekt za mały dla sieci neuronowej
- Algorytm detekcji ruchu
- Przybliżenie ROI

- Obraz powiększony
- Mniejsza rozdzielczość
- Możliwa detekcja i klasyfikacja siecią neuronową



Full Frame

Region of Interest

Możliwe jest dalsze śledzenie obiektu siecią neuronową w ROI

Demo

Dalszy rozwój

Klasyfikacja rozpoznanych obiektów

Plane (95%)

Drone (70%)

Vehicle (90%)

Szukanie zadanych celów

Zwiększenie jakości wykrywania/śledzenia

- Upscaling/denoising za pomocą sieci neuronowych
- Lokalne stosowanie dense optical flow do lepszego śledzenia celów
- …

Dziękuję za uwagę

Jakub Karbowski
carbon @ student.agh.edu.pl

AGH DRONE ENGINEERING