West University of Timișoara

Doctoral Thesis

Cloud Movement Forecasting based on Wind Modeled from Satellite Imagery and a Modified Flocking Algorithm

Author:
Marius E. Penteliuc

Advisor:
Dr. Marc E. Frîncu

DOCTORAL THESIS

Cloud Movement Forecasting based on Wind Modeled from Satellite Imagery and a Modified Flocking Algorithm

Author:
Marius E. PENTELIUC

Advisor:
Dr. Marc E. FRÎNCU

Contents

- Motivation
- State of Art
- World Model
- Modified Boids Algorithm
- Results
- Conclusion

*(*Reference Notation, 2022)*

Contents

- Motivation
- State of Art
- World Model
- Modified Boids Algorithm
- Results
- Conclusion

Contents


Solar Irradiance
the flux of radiant energy per unit area

(Mordor Intelligence, 2021)

Romania Solar Energy Market 2021-2026

Compound Annual Growth Rate 10%

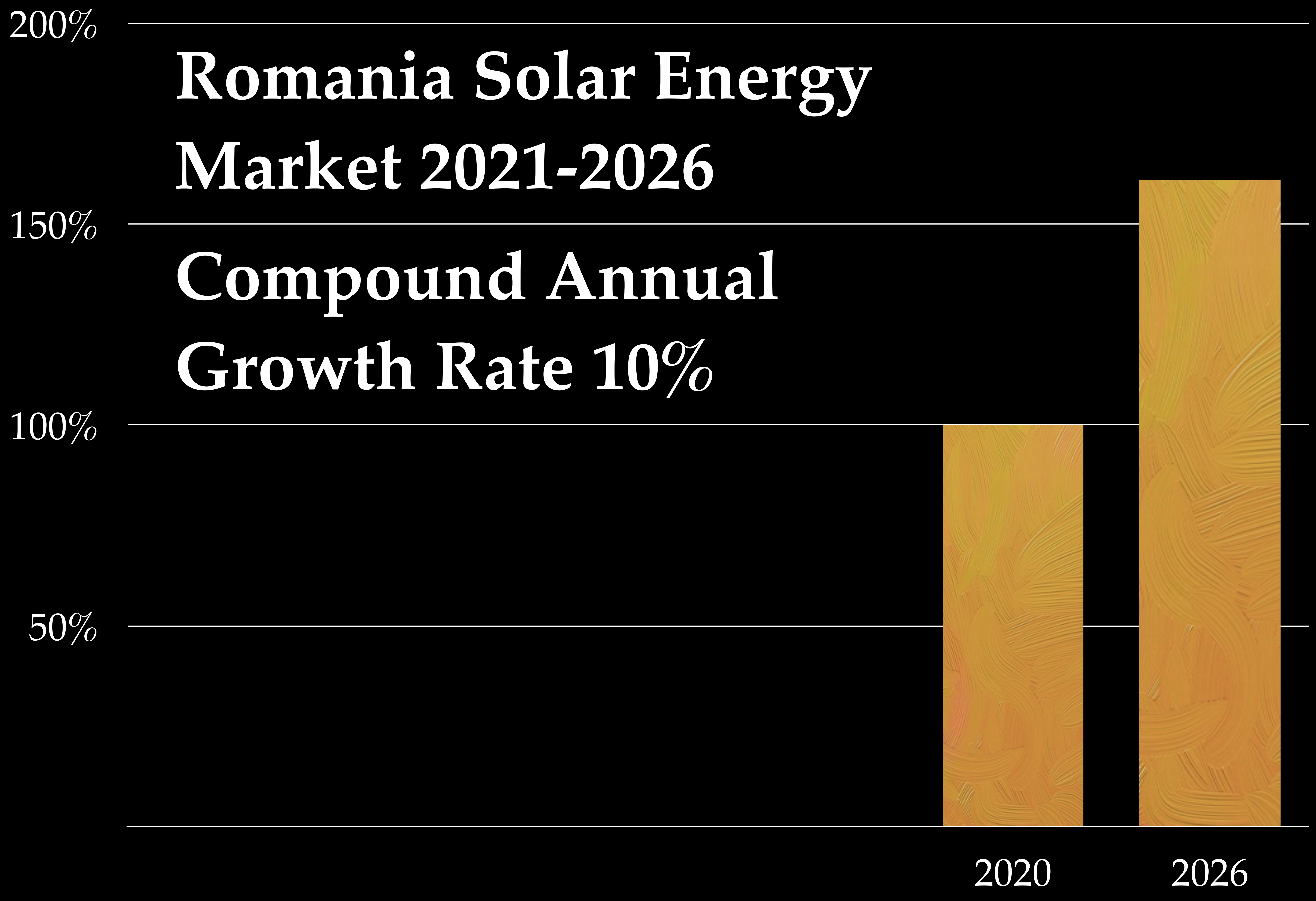
200%
150%
100%
50%

2020 2026

(Mordor Intelligence, 2021)

Romania Solar Energy Market 2021-2026

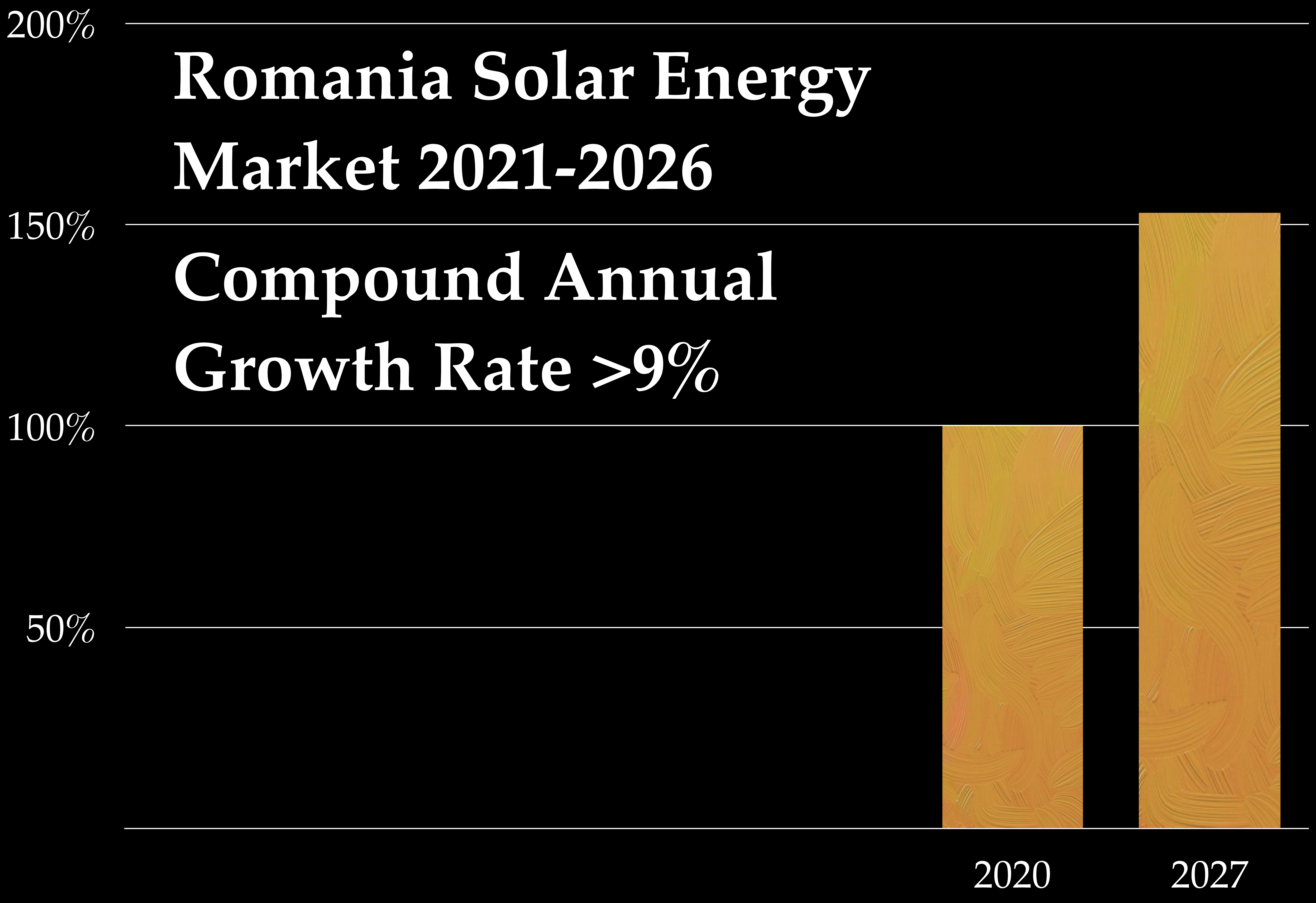
Compound Annual Growth Rate 10%



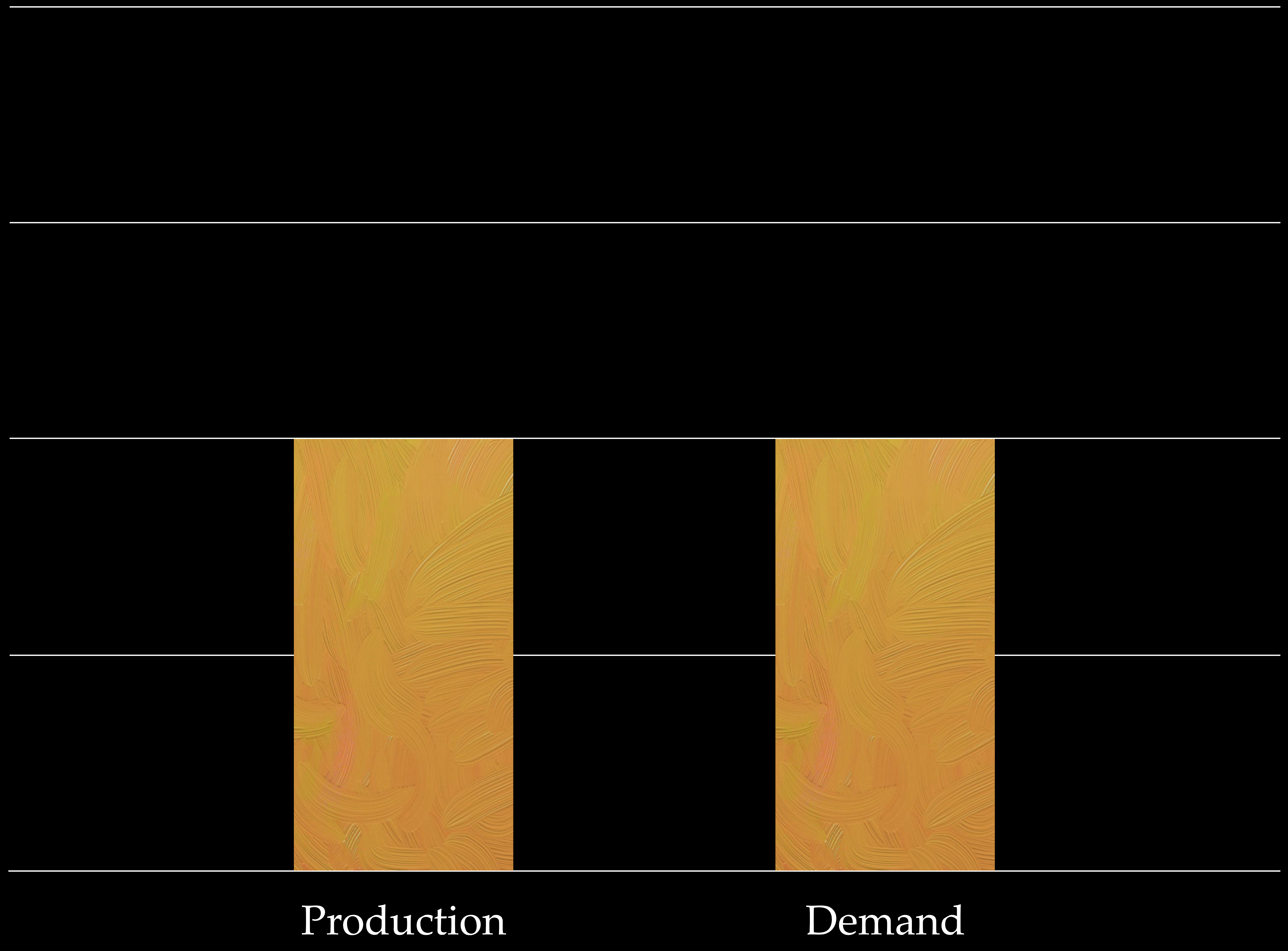
(*Mordor Intelligence, 2021*)

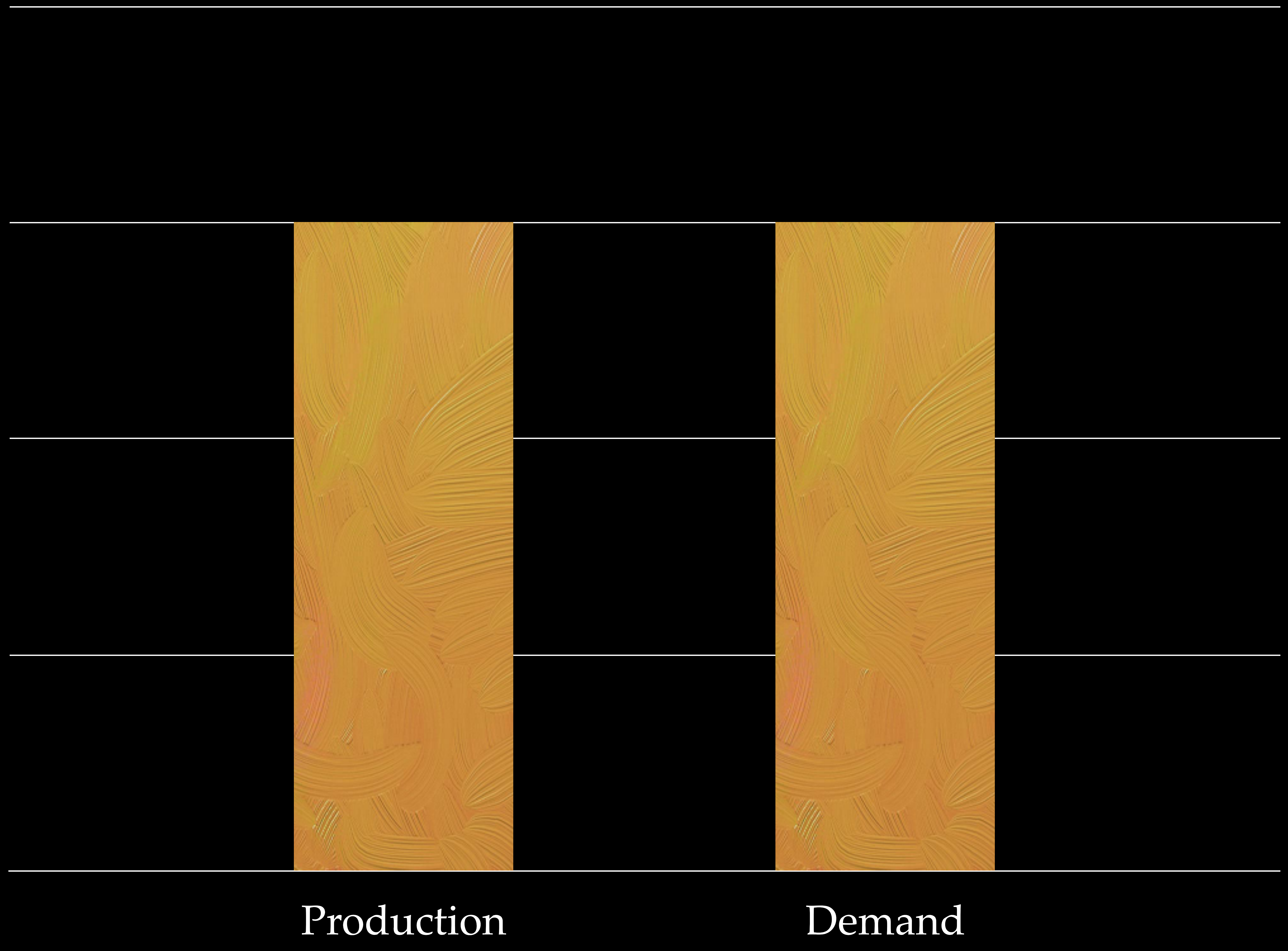
Romania Solar Energy Market 2021-2026

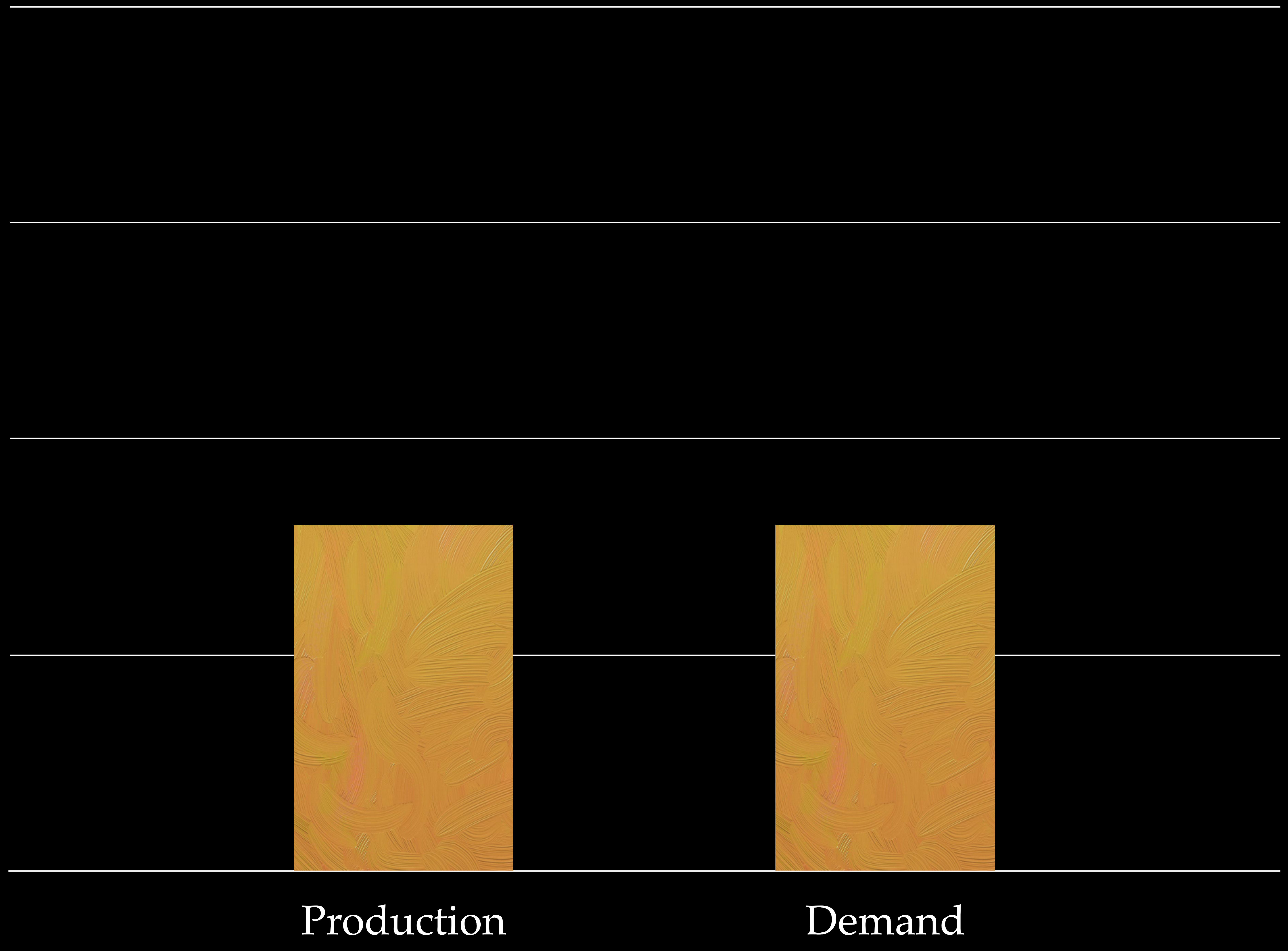
Compound Annual Growth Rate >9%

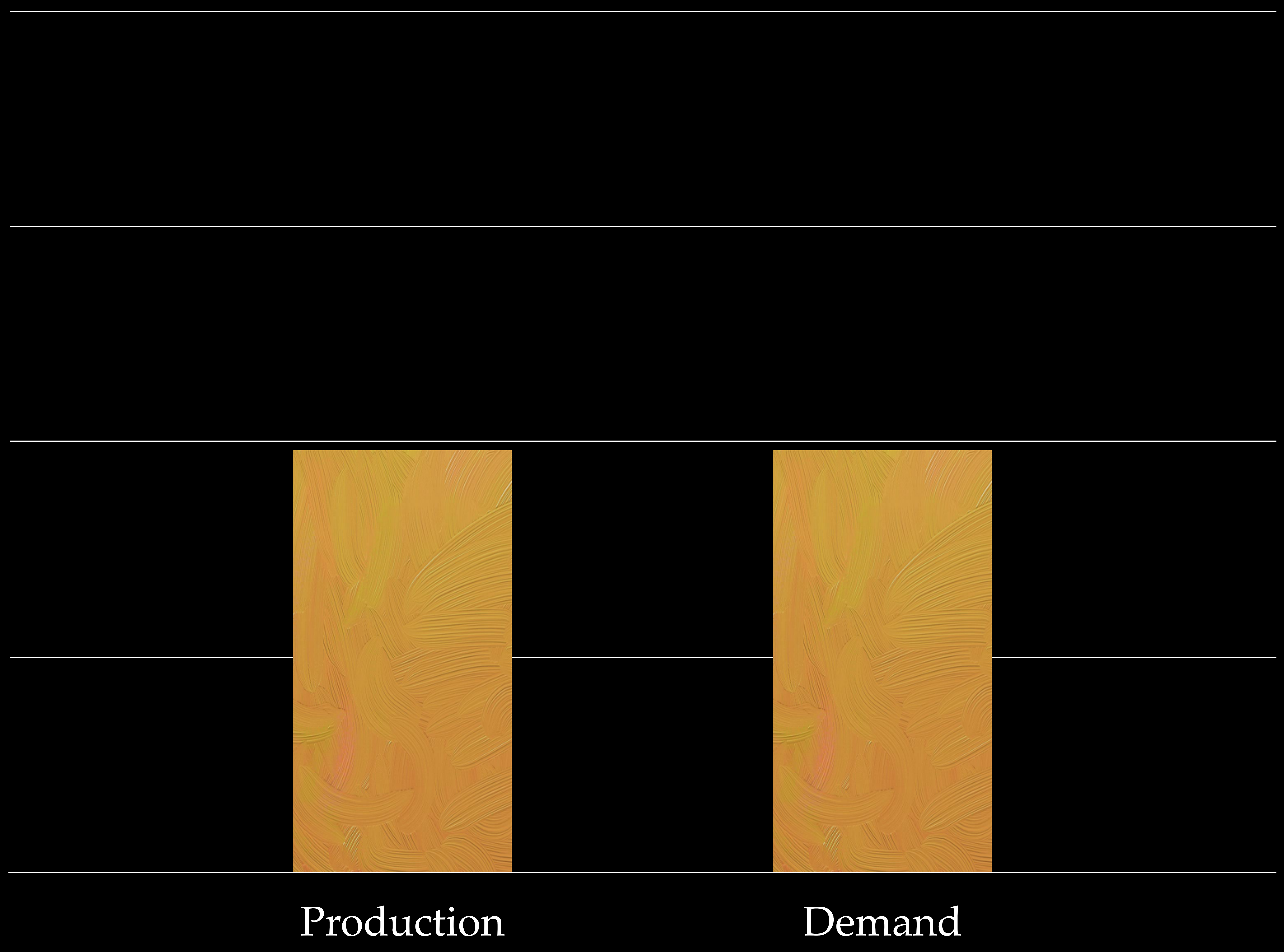


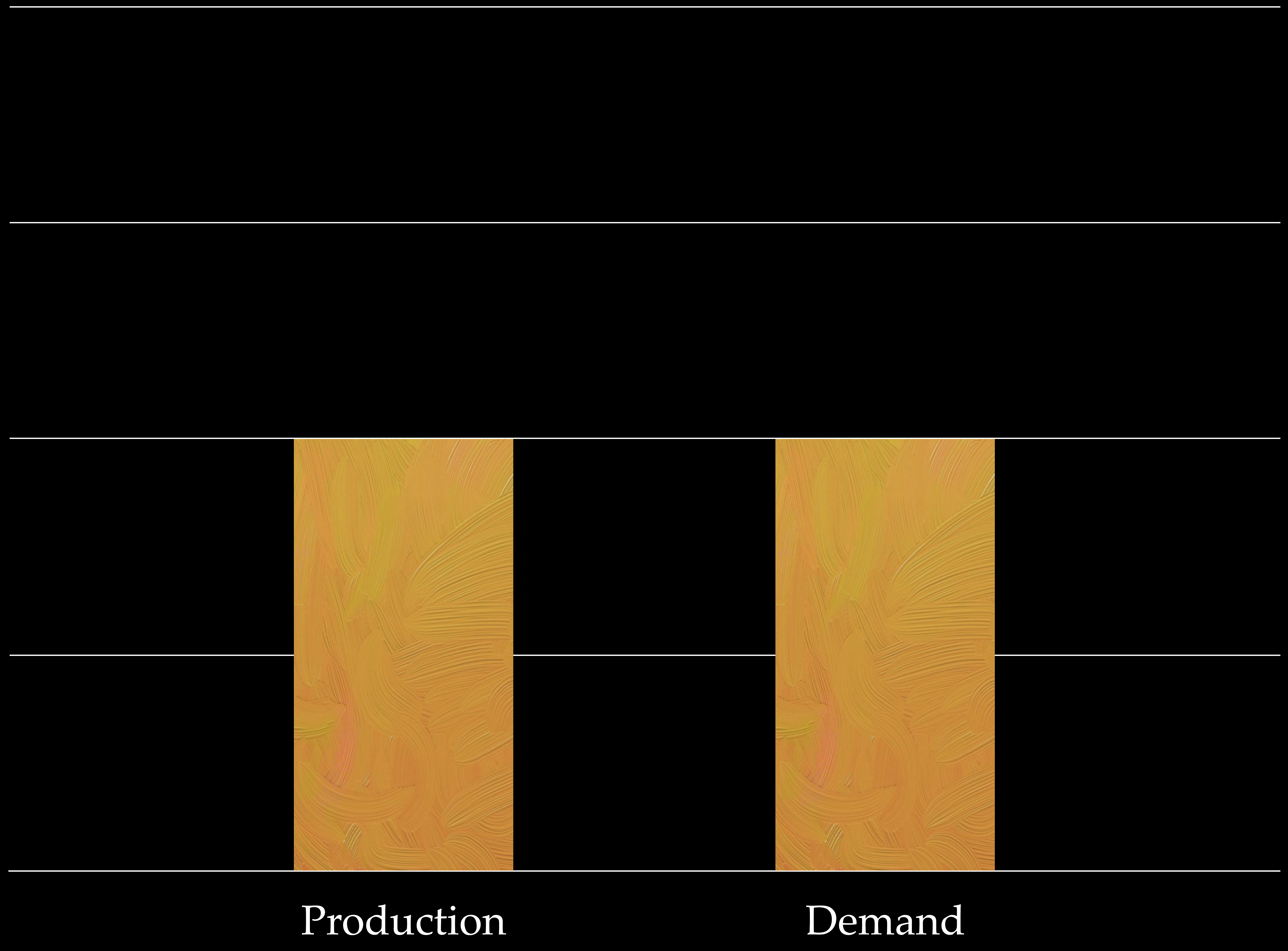
Production Demand

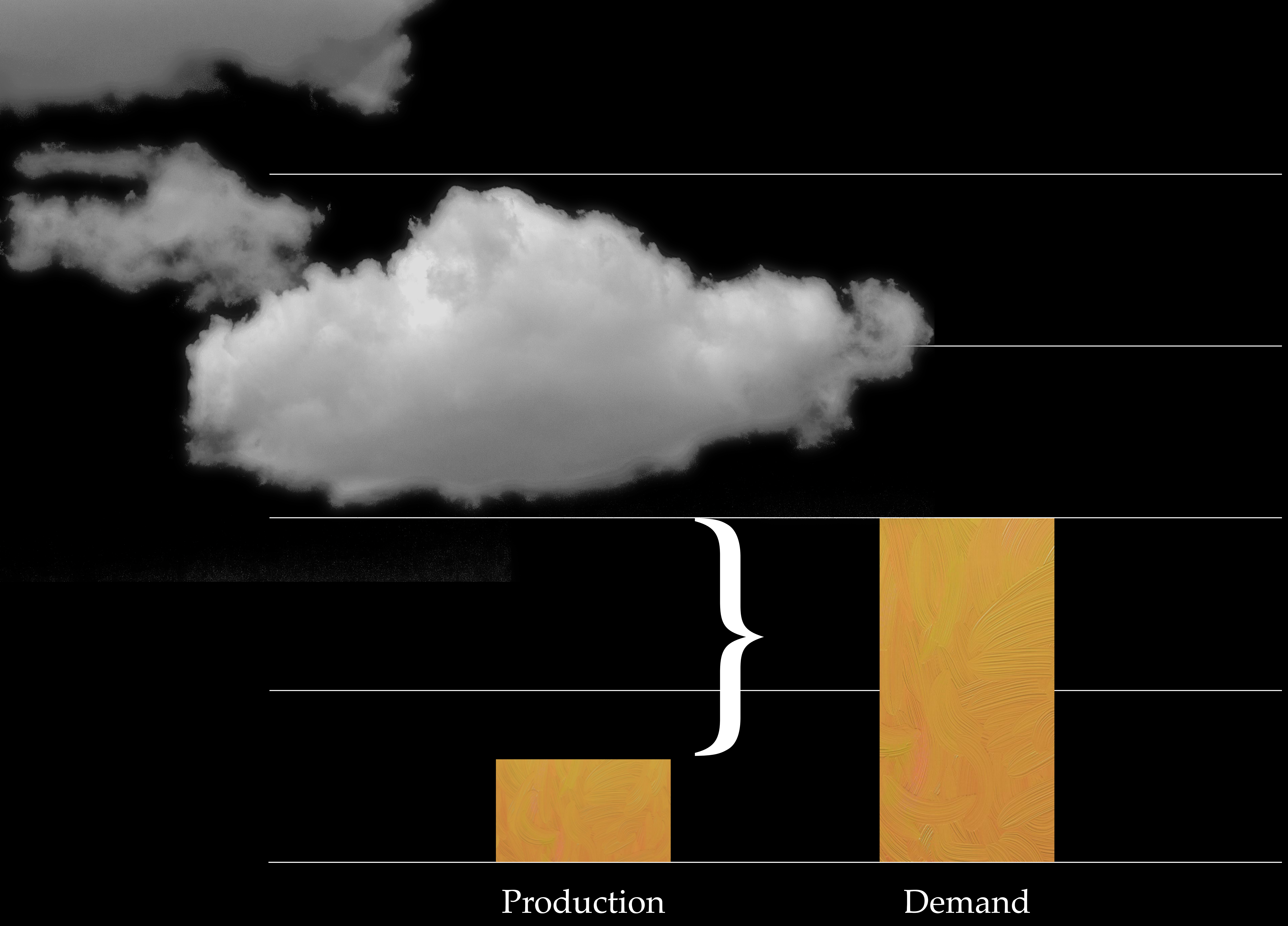

Production
Demand


Production
Demand


Production
Demand


Production
Demand


Production
Demand


Production
Demand

Forecasting Time Scales

(Alonso-Suárez et al., 2020)

nowcasting

short-term

mid-term

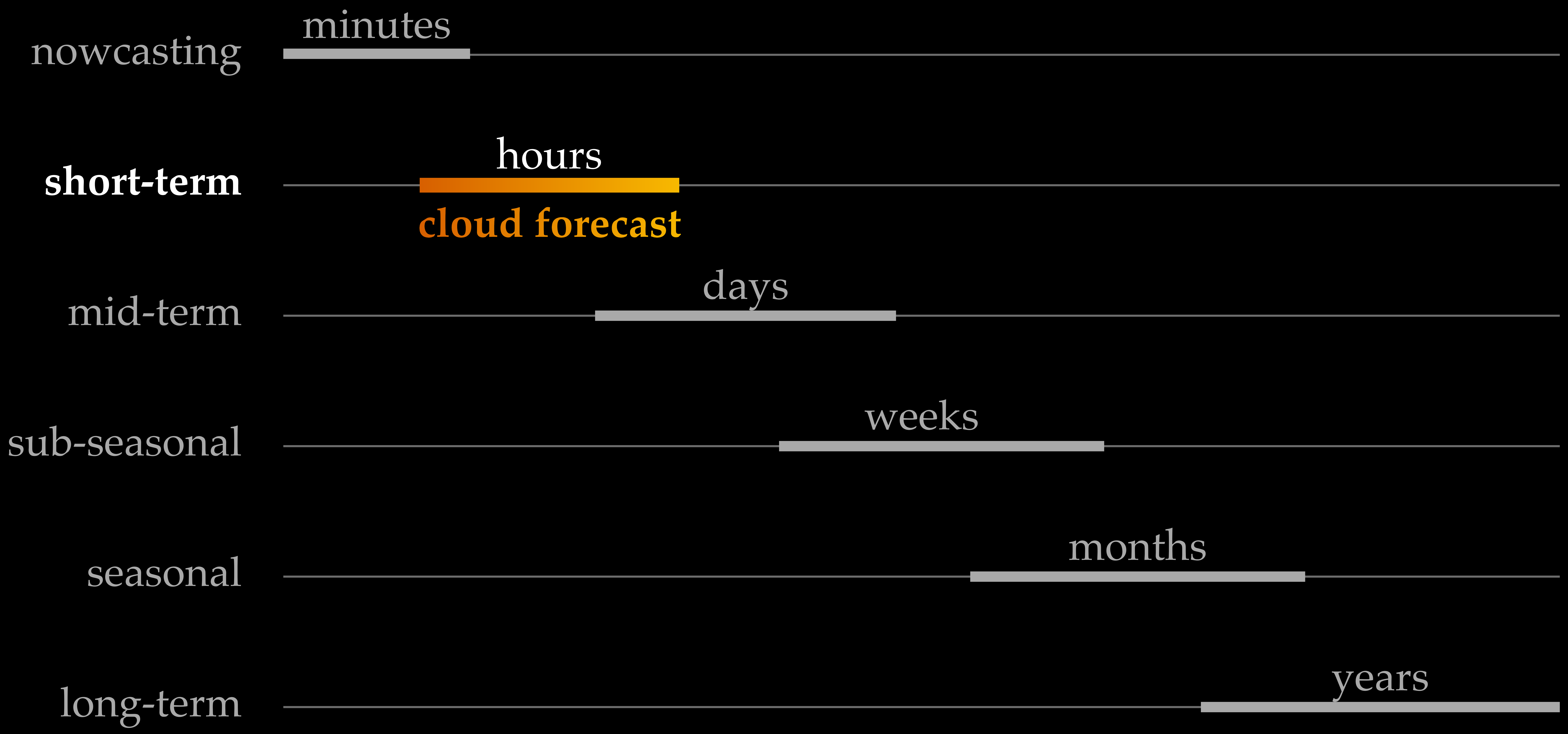
sub-seasonal

seasonal

long-term

Forecasting Time Scales

(Alonso-Suárez et al., 2020)

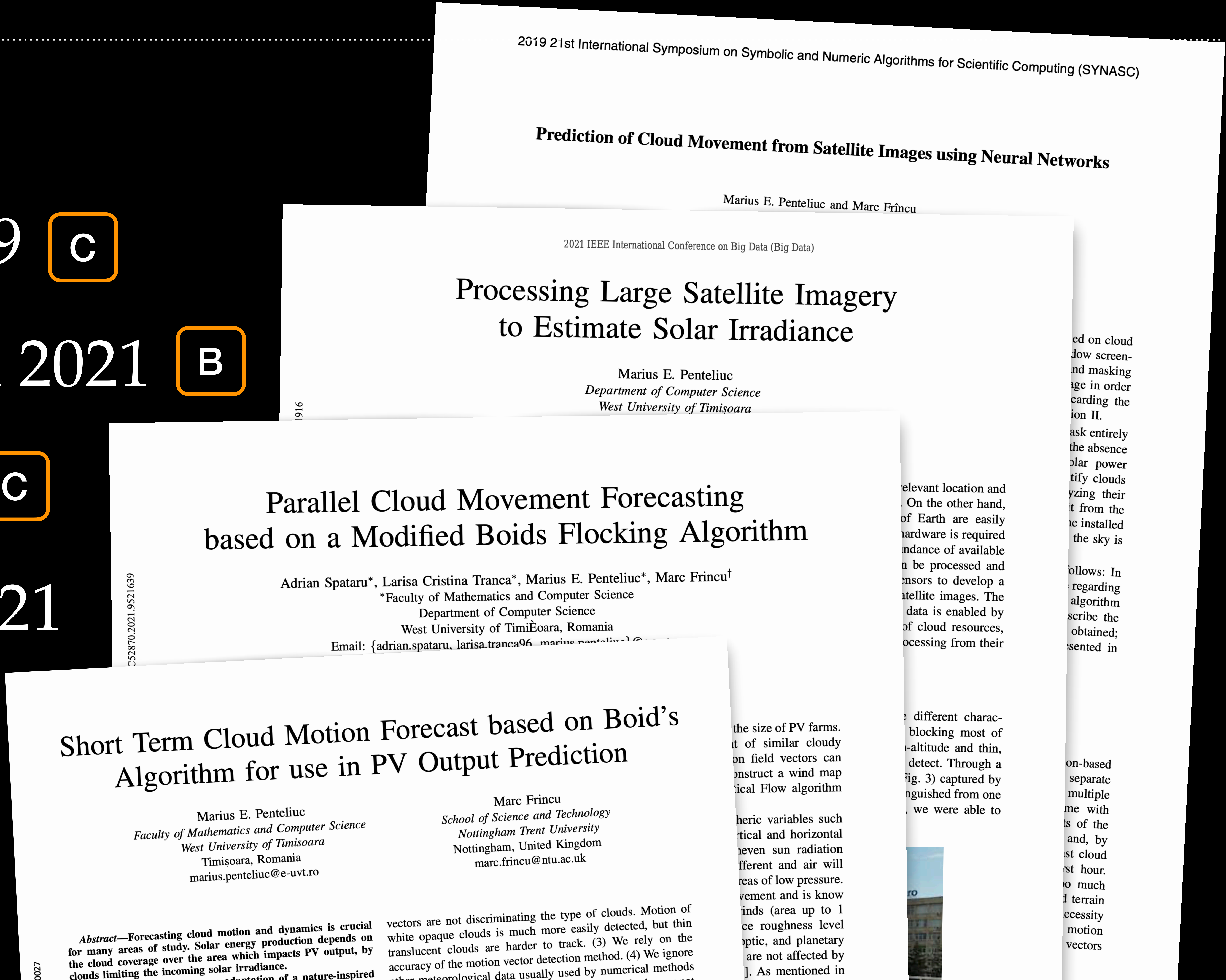


Published Papers

- SYNASC 2019
- IEEE BigData 2021
- ISPDC 2021
- IEEE ISGT 2021

Published Papers

- SYNASC 2019 C
- IEEE BigData 2021 B
- ISPDC 2021 C
- IEEE ISGT 2021
  DOMAIN SPECIFIC



This work was supported by a grant from the
Romanian Ministry of Education and Research,
CNCS - UEFISCDI project number
PN-III-P1-1.1-TE-2019-0859,
within PNCDI III.

Contents

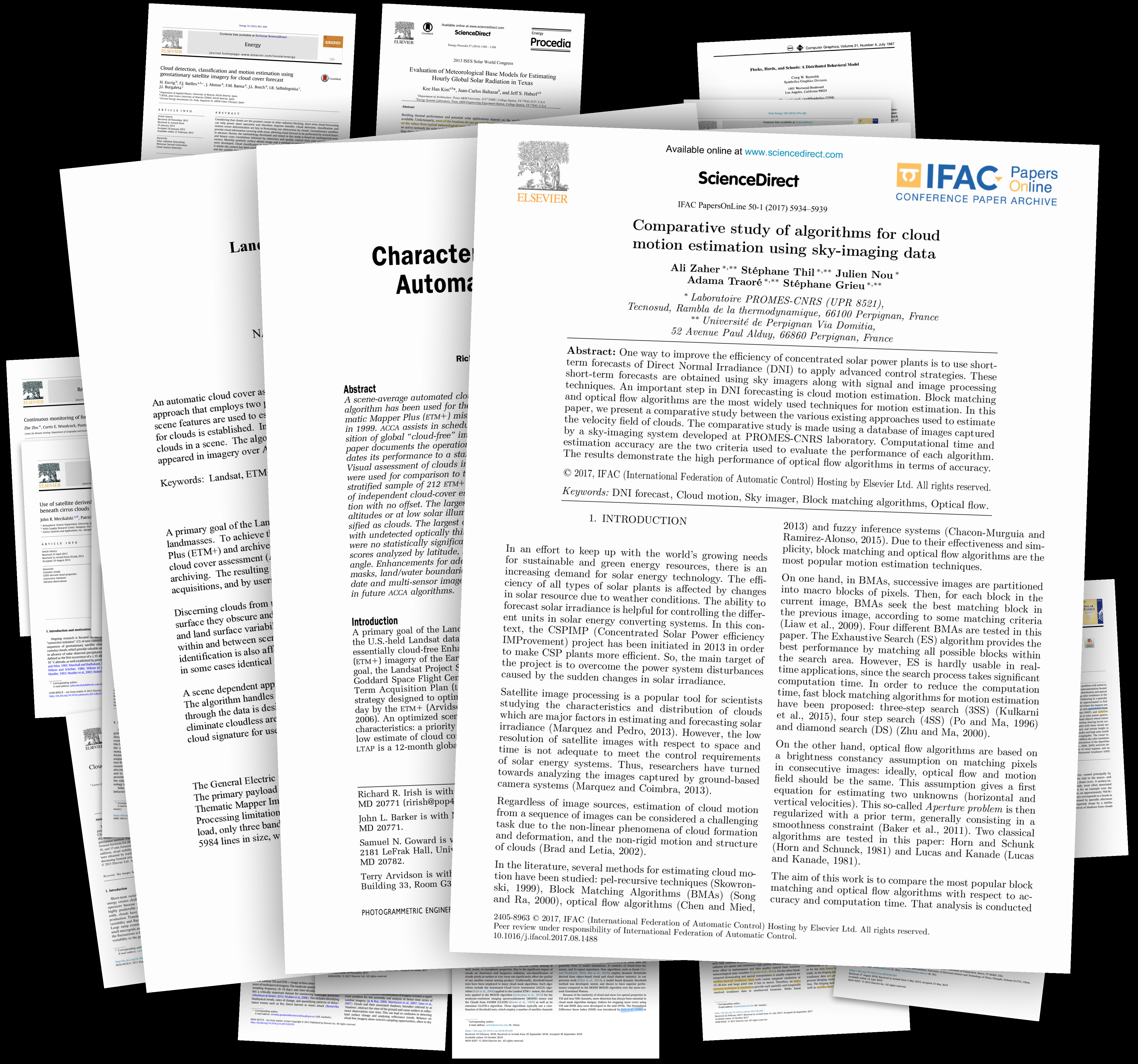
- Motivation
- **State of Art**
- World Model
- Modified Boids Algorithm
- Results
- Conclusion

State of Art


Cloud detection, classification and motion estimation using geostationary satellite imagery for cloud cover forecast
Evaluation of Meteorological Base Models for Estimating Hourly Global Solar Radiation in Texas
Energy Procedia
Solar Energy
Intra-hour cloud index forecasting with data assimilation
Energy Conversion and Management
Image phase shift invariance based cloud motion displacement vector calculation method for ultra-short-term solar PV power forecasting
remote sensing
Making Landsat Time Series Consistent: Evaluating and Improving Landsat Analysis Ready Data
Introducing two Random Forest based methods for cloud detection in remote sensing images
Renewable Energy
Intra-day solar probabilistic forecasts including local short-term variability and satellite information
Short-term solar irradiance forecasting via satellite/model coupling
Robust cloud motion estimation by spatio-temporal correlation analysis of irradiance data
Remote Sensing of Environment
New neural network cloud mask algorithm based on radiative transfer simulations
Fmask 4.0: Improved cloud and cloud shadow detection in Landsats 4-8 and Sentinel-2 imagery

State of Art

(Su, 1984; Irish, 2000; Irish et al., 2006)



- ✓ Early cloud assessment method;
- ✓ Processed satellite imagery;
- ✓ Filtered cloudy images;

- ✗ Computed cloud cover percentage;
- ✗ Hardware had limitations;
- ✗ Utilized **orbiting** satellites.

State of Art

(Wang et al., 1999; Goodwin et al., 2013; Escrig et al., 2013)

- ✓ Fused multiple images;
- ✓ Constructed cloud-free images;
- ✓ Isolated clouds using time series;

- ✗ Need captures of cloud-free areas;
- ✗ Time series constructed from **years** of pictures.

State of Art

(Mecikalsky, Minnis, and Palikonda, 2013; Zhu and Helmer, 2018)

- ✓ Quantify clouds beneath other clouds;
- ✓ Fewer images than other methods;
- ✓ Less bands needed for computation;

- ✗ Nowcasting of thunderstorms;
- ✗ Developed for tropical and subtropical regions;

State of Art

(Zhu and Woodcock, 2012 - 2019)

✓ Inherits earlier method's strengths;
✓ Is frequently updated;
✓ Offers extended hardware and sensor support;

✗ Goal to remove clouds;
✗ Needs **external** data;

State of Art

(Chen et al., 2018)

Remote Sensing of Environment 219 (2018) 62–71

Contents lists available at ScienceDirect

Remote Sensing of Environment

journal homepage: www.elsevier.com/locate/rse

New neural network cloud mask algorithm based on radiative transfer simulations

Nan Chen[a,*], Wei Li[a], Charles Gatebe[b], Tomonori Tanikawa[c], Masahiro Hori[d], Rigen Shimada[d], Teruo Aoki[e], Knut Stamnes[a]

[a] Department of Physics, Stevens Institute of Technology, Hoboken, USA
[b] NASA GSFC, Greenbelt, MD, USA
[c] Meteorological Research Institute, Tsukuba, Japan
[d] Japan Aerospace Exploration Agency, Tsukuba, Japan
[e] Okayama University, Okayama, Japan

ARTICLE INFO

Keywords:
Cloud mask
Machine learning
Radiative transfer

2010 MSC:
00-01
99-00

ABSTRACT

Cloud detection and screening constitute critically important first steps required to derive many satellite data products. Traditional threshold-based cloud mask algorithms require a complicated design process and fine tuning for each sensor, and they have difficulties over areas partially covered with snow/ice. Exploiting advances in machine learning techniques and radiative transfer modeling of coupled environmental systems, we have developed a new, threshold-free cloud mask algorithm based on a neural network classifier driven by extensive radiative transfer simulations. Statistical validation results obtained by using collocated CALIOP and MODIS data show that its performance is consistent over different ecosystems and significantly better than the MODIS Cloud Mask (MOD35 C6) during the winter seasons over snow-covered areas in the mid-latitudes. Simulations using a reduced number of satellite channels also show satisfactory results, indicating its flexibility to be configured for different sensors. Compared to threshold-based methods and previous machine-learning approaches, this new cloud mask (i) does not rely on thresholds, (ii) needs fewer satellite channels, (iii) has superior performance during winter seasons in mid-latitude areas, and (iv) can easily be applied to different sensors.

1. Introduction

1.1. Background

A reliable cloud mask is essential for satellite remote sensing of land, ocean, or cryospheric properties. Due to the significant impact of clouds on shortwave and longwave radiation, mis-identification of cloudy pixels as surface or vice versa can significantly affect the quality of any satellite remote sensing product. Traditionally, threshold-based tests have been employed in many cloud mask algorithms. Such algorithms include the Automated Cloud Cover Assessment (ACCA) algorithm (Irish et al., 2006) applied to the Landsat ETM+ sensor, the cloud tests applied in the MOD35 algorithm (Ackerman et al., 2010) for the moderate-resolution imaging spectroradiometer (MODIS) sensor and the Clouds from AVHRR (CLAVR) (Stowe et al., 1999) as well as its extension CLAVR-x algorithm. These algorithms typically use a combination of threshold tests, which employ a number of satellite channels located in the visible (VIS), near infrared (NIR), shortwave infrared (SWIR), and thermal infrared (TIR) wavelength ranges (e.g. MOD35 uses 19 bands — 10 reflectance bands and 9 thermal infrared bands) to detect clouds and snow/ice. The thresholds used in these tests are generally from 1) model simulations, 2) statistics of cloud/clear-sky scenes, and 3) expert experience. New algorithms, such as fmask (Zhu and Woodcock, 2012; Zhu et al., 2015), employ dynamic thresholds derived from object-based cloud and cloud shadow statistics. In our previous work (Chen et al., 2014), a model based dynamic threshold method was developed, tested, and shown to have superior performance compared to the MODIS MOD35 algorithm over the snow-covered Greenland Plateau.

Because of the similarity of cloud and snow/ice optical properties in VIS and near NIR channels, snow detection has always been essential in cloud mask algorithm designs. Indices for mapping snow cover using VIS and SWIR data were developed in the mid-1970s. The Normalized Difference Snow Index (NDSI) was introduced by Hall et al. (1995) to

* Corresponding author.
E-mail address: nchen@stevens.edu (N. Chen).

https://doi.org/10.1016/j.rse.2018.09.029
Received 10 February 2018; Received in revised form 25 September 2018; Accepted 30 September 2018
Available online 10 October 2018

- ✔ Differentiate clouds from snow;
- ✘ Needs **training** for each sensor and scene configuration;

State of Art

(Hamill and Nehrkorn, 1993; Escrig et al., 2013)


Energy 55 (2013) 853–859

Contents lists available at SciVerse ScienceDirect

Energy

journal homepage: www.elsevier.com/locate/energy

Cloud detection, classification and motion estimation using geostationary satellite imagery for cloud cover forecast

H. Escrig [b], F.J. Batlles [a,b,*], J. Alonso [b], F.M. Baena [b], J.L. Bosch [b], I.B. Salbidegoitia [c], J.I. Burgaleta [c]

ARTICLE INFO

Article history:
Received 28 November 2012
Received in revised form 25 January 2013
Accepted 28 January 2013
Available online 21 February 2013

Keywords:
Solar radiation forecasting
Meteosat Second Generation
Cloud motion detection

ABSTRACT

Considering that clouds are the greatest causes to solar radiation blocking, short term cloud forecasting can help power plant operation and therefore improve benefits. Cloud detection, classification and motion vector determination are key to forecasting sun obstruction by clouds. Geostationary satellites provide cloud information covering wide areas, allowing cloud forecast to be performed for several hours in advance. Herein, the methodology developed and tested in this study is based on multispectral tests and binary cross correlations followed by coherence and quality control tests over resulting motion vectors. Monthly synthetic surface albedo image and a method to reject erroneous correlation vectors were developed. Cloud classification in terms of opacity and height of cloud top is also performed. A whole-sky camera has been used for validation, showing over 85% of agreement between the camera and the satellite derived cloud cover, whereas error in motion vectors is below 15%.

© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

A Short-Term Cloud

- ✓ Short-term cloud forecasting using cross-correlation;
- ✓ Computing motion vectors;

- ✗ Parts of the scene are interpolated;
- ✗ Advection applied to entire image;
- ✗ Image segmenting leads to a lack of **granularity**.

State of Art

(Chow, Belongie, and Kleissl, 2015)

- ✓ Optical Flow-based method;
- ✓ Better than cross-correlation methods;
- ✓ Captured multiple independent cloud motions;

- ✗ Forecasts just for the camera's location;
- ✗ Unstable clouds lower forecast confidence.
- ✗ No **geographic mapping** of pixels;

State of Art

(Alonso-Suárez et al., 2020; Perez et al. 2010)

- ✔ Irradiance forecasts up to three hours;
- ✔ Often used in weather forecasting;

- ✘ No cloud position or extent;
- ✘ Outperformed by image-based forecasts;
- ✘ **Site-specific** forecasts.

Disadvantages

Poor Hardware | Sensor Training | Maintenance Cost | Cloud Groups

Complex Logic | Scene Training | Orbiting Satellites | Straight Motion

No Granularity | Pixel Geo-mapping | Camera Location Specific | Cloud Cover

Cloud Type Insensitivity | Cloud-Free Observations | Cloud Removal

Region Specific | Thunderstorms | Image Filtering | Snow Discrimination

Days-ahead Forecasting | Interpolation | Irradiance Sensors | Many Images

External Data | Terrain Advection | Sky Imagery | Image Segmentation | Nowcasting

My Contribution

Poor Hardware Sensor Training Maintenance Cost Cloud Groups

Complex Logic Scene Training Orbiting Satellites Straight Motion

No Granularity Pixel Geo-mapping Camera Location Specific Cloud Cover

Cloud Type Insensitivity Cloud-Free Observations Cloud Removal

Region Specific Thunderstorms Image Filtering Snow Discrimination

Days-ahead Forecasting Interpolation Irradiance Sensors Many Images

External Data Terrain Advection Sky Imagery Image Segmentation Nowcasting

My Contribution

Commodity Hardware Sensor Training **Low Maintenance Costs** Cloud Groups

Complex Logic Scene Training **Geostationary Satellites** **Curved Paths**

Per Pixel Forecast Pixel Geo-mapping Camera Location Specific Cloud Cover

Cloud Type Insensitivity **Utilize Cloudy Pixels** Cloud Removal

Generalized Application Thunderstorms Image Filtering Snow Discrimination

Days-ahead Forecasting Interpolation Irradiance Sensors **Start with Two Images**

No Additional Input Terrain Advection Sky Imagery **Pixel Level Motion Detection** Nowcasting

My Contribution

Geostationary Satellites
Commodity Hardware
Low Maintenance Costs
Start with Two Images
No Additional Input
Utilize Cloudy Pixels
Curved Paths
Pixel Level Motion Detection
Per Pixel Forecast
Generalized Application

Increased time series frequency compared to orbiting satellites.

Enables hours-ahead forecasts.

Larger horizon compared to all-sky cameras.

My Contribution

Geostationary Satellites

Commodity Hardware

Low Maintenance Costs

Start with Two Images

No Additional Input

Utilize Cloudy Pixels

Curved Paths

Pixel Level Motion Detection

Per Pixel Forecast

Generalized Application

No need for powerful super-computers.

No specialized hardware requirements for acquiring data.

My Contribution

Geostationary Satellites

Commodity Hardware

Low Maintenance Costs

Start with Two Images

No Additional Input

Utilize Cloudy Pixels

Curved Paths

Pixel Level Motion Detection

Per Pixel Forecast

Generalized Application

No historical data needed to start forecasting, only last hour images.

Only provide images, no other ground-measured data or weather information.

My Contribution

Geostationary Satellites

Commodity Hardware

Low Maintenance Costs

Start with Two Images

No Additional Input

Utilize Cloudy Pixels

Curved Paths

Pixel Level Motion Detection

Per Pixel Forecast

Generalized Application

Not necessary to filter out cloudy images. Forecasts are made based on previous cloud motion.

My Contribution

Geostationary Satellites

Commodity Hardware

Low Maintenance Costs

Start with Two Images

No Additional Input

Utilize Cloudy Pixels

Curved Paths

Pixel Level Motion Detection

Per Pixel Forecast

Generalized Application

Simulate realistic behavior of cloud movement on a global scale.

My Contribution

Geostationary Satellites

Commodity Hardware

Low Maintenance Costs

Start with Two Images

No Additional Input

Utilize Cloudy Pixels

Curved Paths

Pixel Level Motion Detection

Per Pixel Forecast

Generalized Application

Motion detection and forecasts are per pixel, not groups of pixels or image segments.

My Contribution

Geostationary Satellites

Commodity Hardware

Low Maintenance Costs

Start with Two Images

No Additional Input

Utilize Cloudy Pixels

Curved Paths

Pixel Level Motion Detection

Per Pixel Forecast

Generalized Application

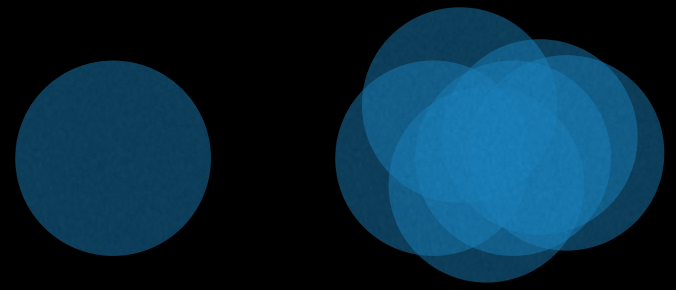
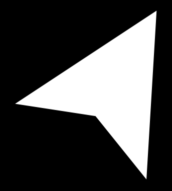
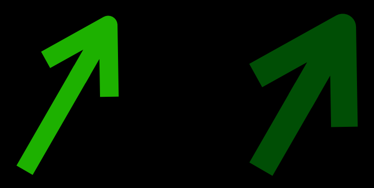
Solution ignores background information (which is still) and works with observed motion.

No training needed for a particular region.

Contents

- Motivation
- State of Art
- **World Model**
- Modified Boids Algorithm
- Results
- Conclusion

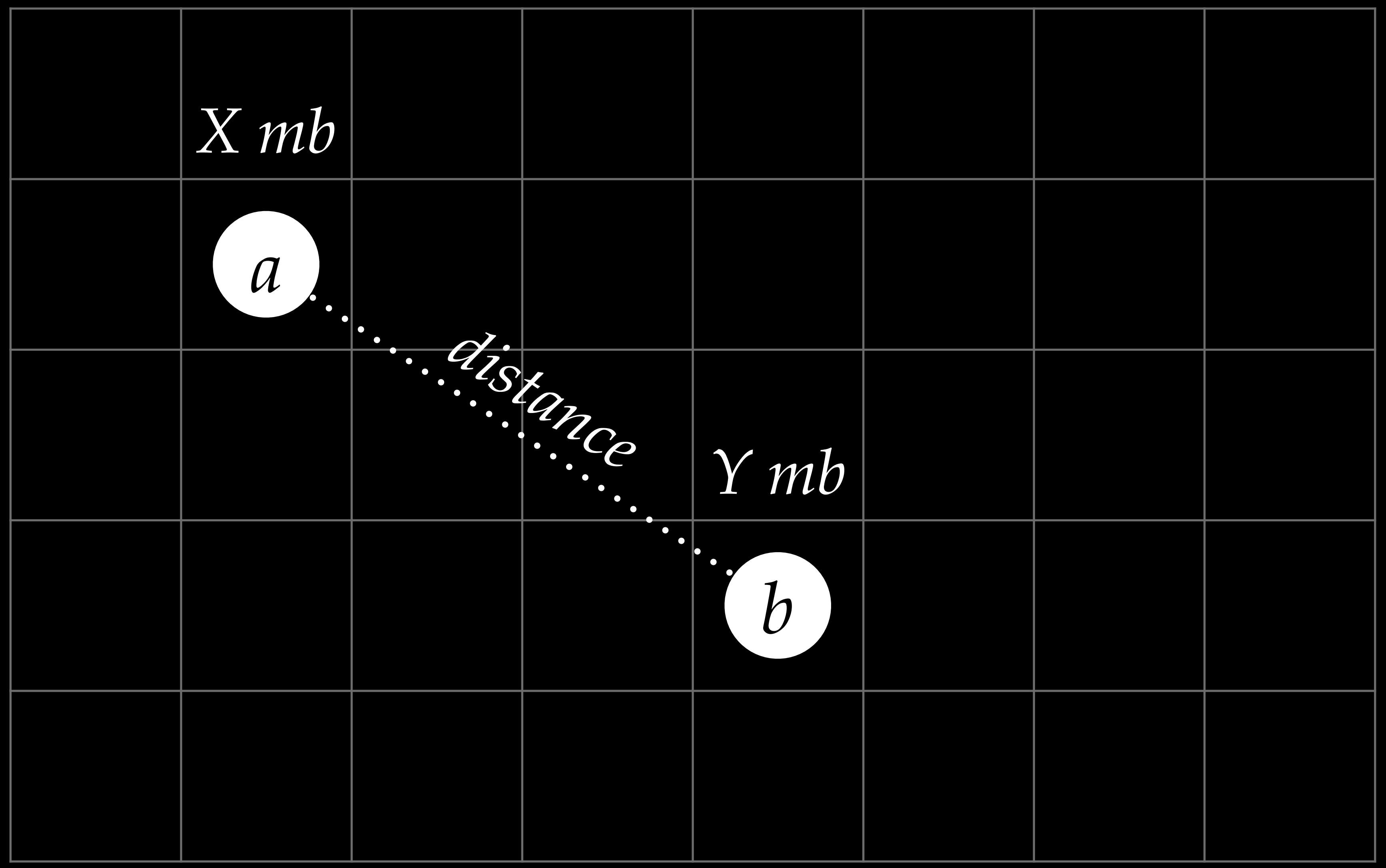
Notations

scene	S
wind	W
wind map	W_{map}
cloud	C 
clear	$\overline{C}$
cloud mask	CM
boid object	b 
motion vector	$\vec{v}$ 

The Pressure Gradient Force

(*Petersen et al., 1998a*)

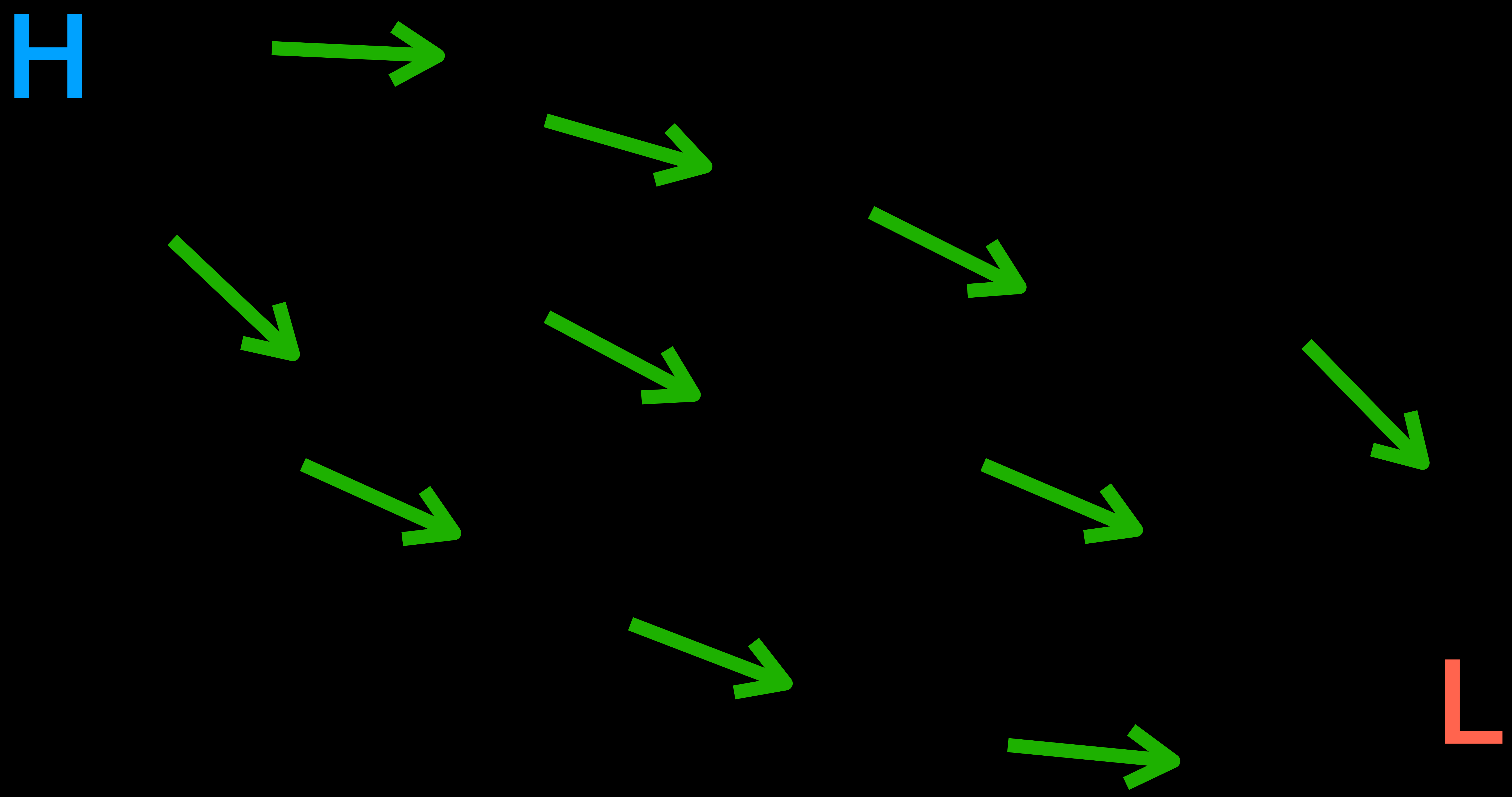
$$\overrightarrow{\textbf{Pressure Gradient Force}} = \frac{X - Y}{\textbf{distance}}$$



high pressure = cold air H

low pressure = warm air L

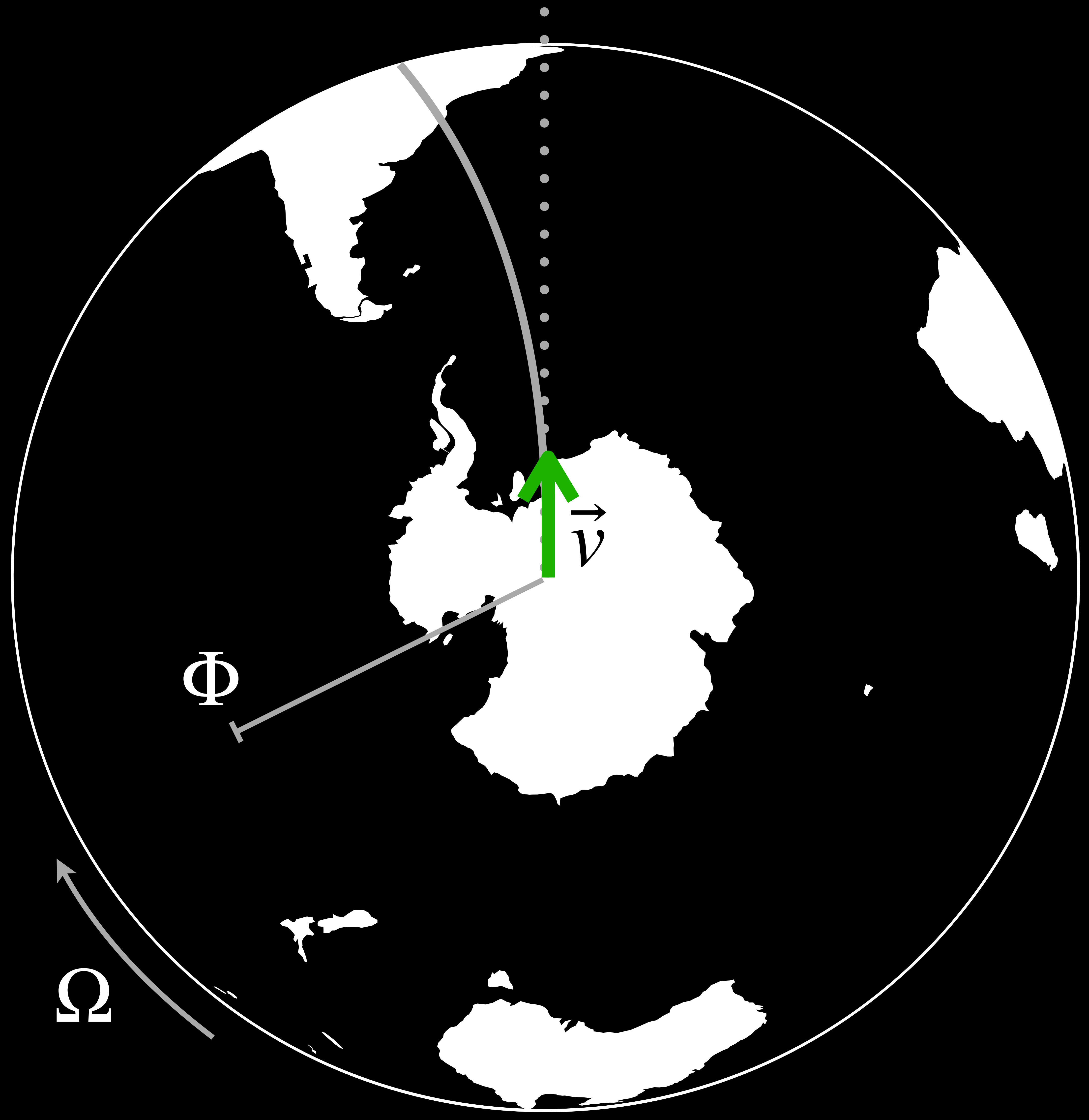
The Pressure Gradient Force



The Coriolis Force

(Coriolis, 1832 & 1835)

$$\overrightarrow{\textbf{Coriolis Force}} = 2\Omega\vec{v}\sin\Phi$$



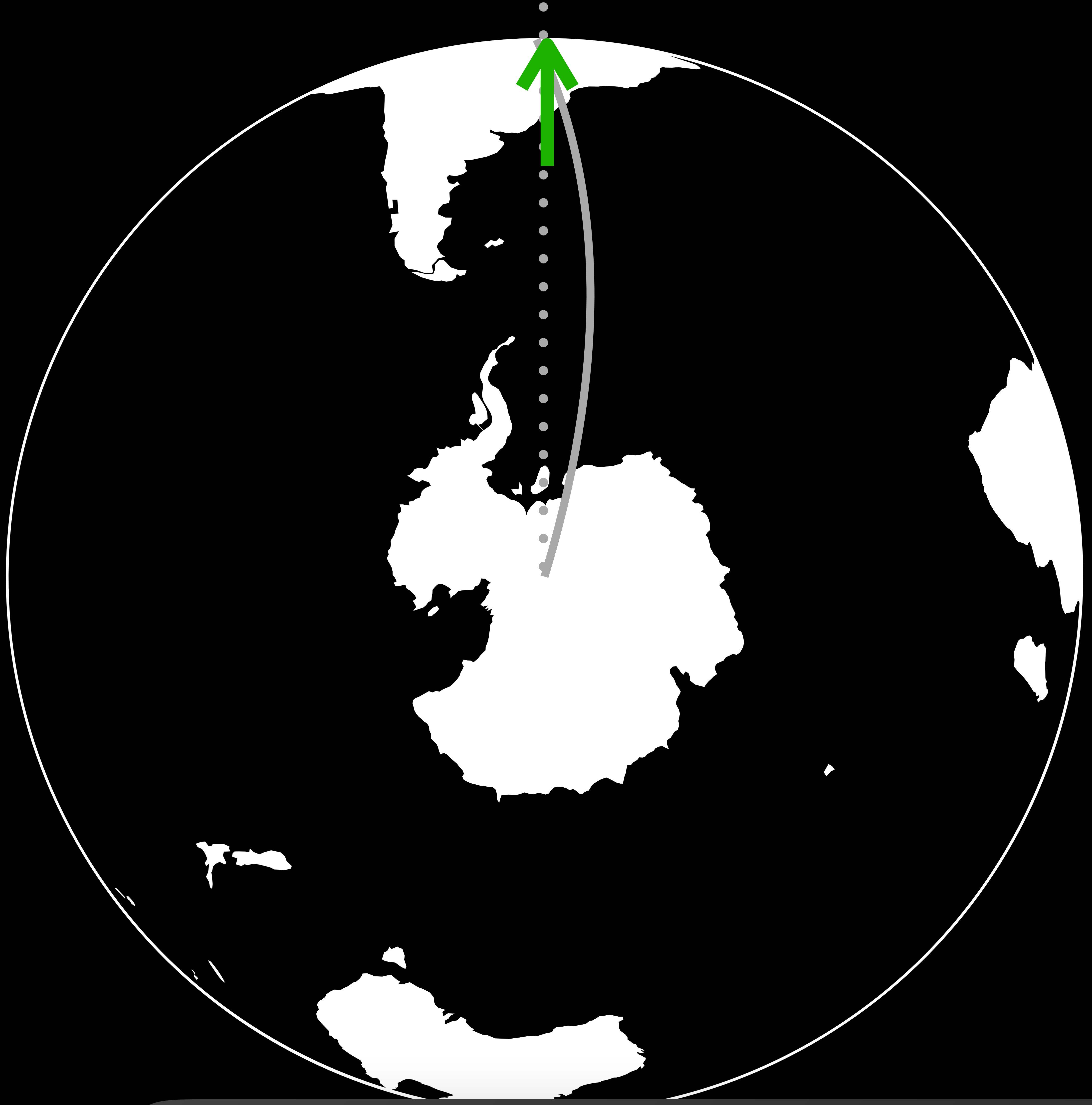
Ω – Earth's angular velocity
$\vec{v}$ – velocity of moving body
Φ – latitude of moving body

The Coriolis Force

(Coriolis, 1832 & 1835)

$$\overrightarrow{\textbf{Coriolis Force}} = 2\Omega\vec{v}\sin\Phi$$

Ω – Earth's angular velocity
$\vec{v}$ – velocity of moving body
Φ – latitude of moving body



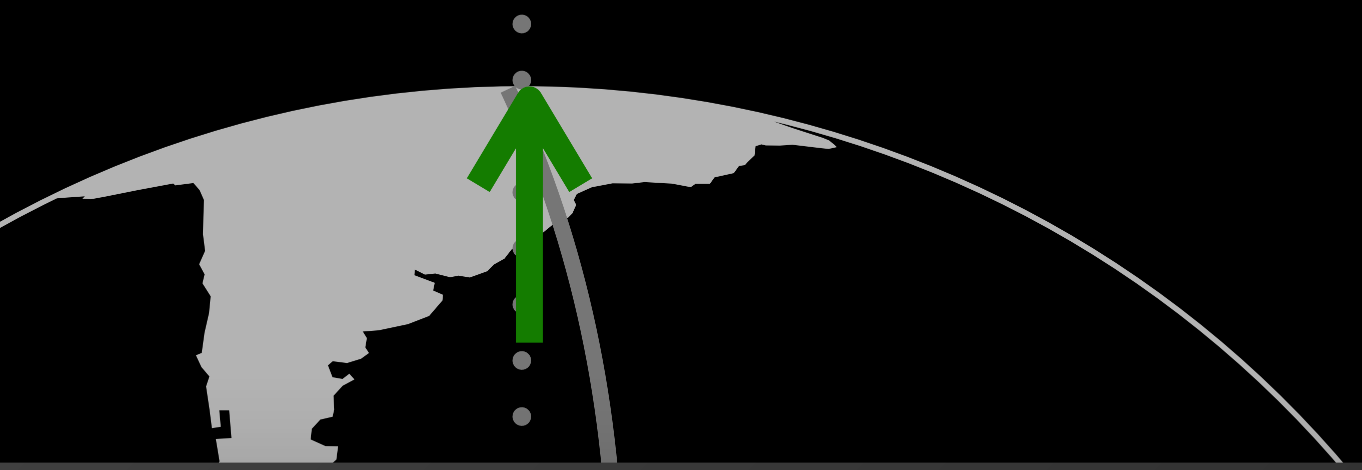
footnote

$$\overrightarrow{\textbf{Coriolis Force}} = 2\Omega\vec{v}\sin\Phi$$

Ω – **Earth's angular velocity**

$\vec{v}$ – **velocity of moving body**

Φ – **latitude of moving body**



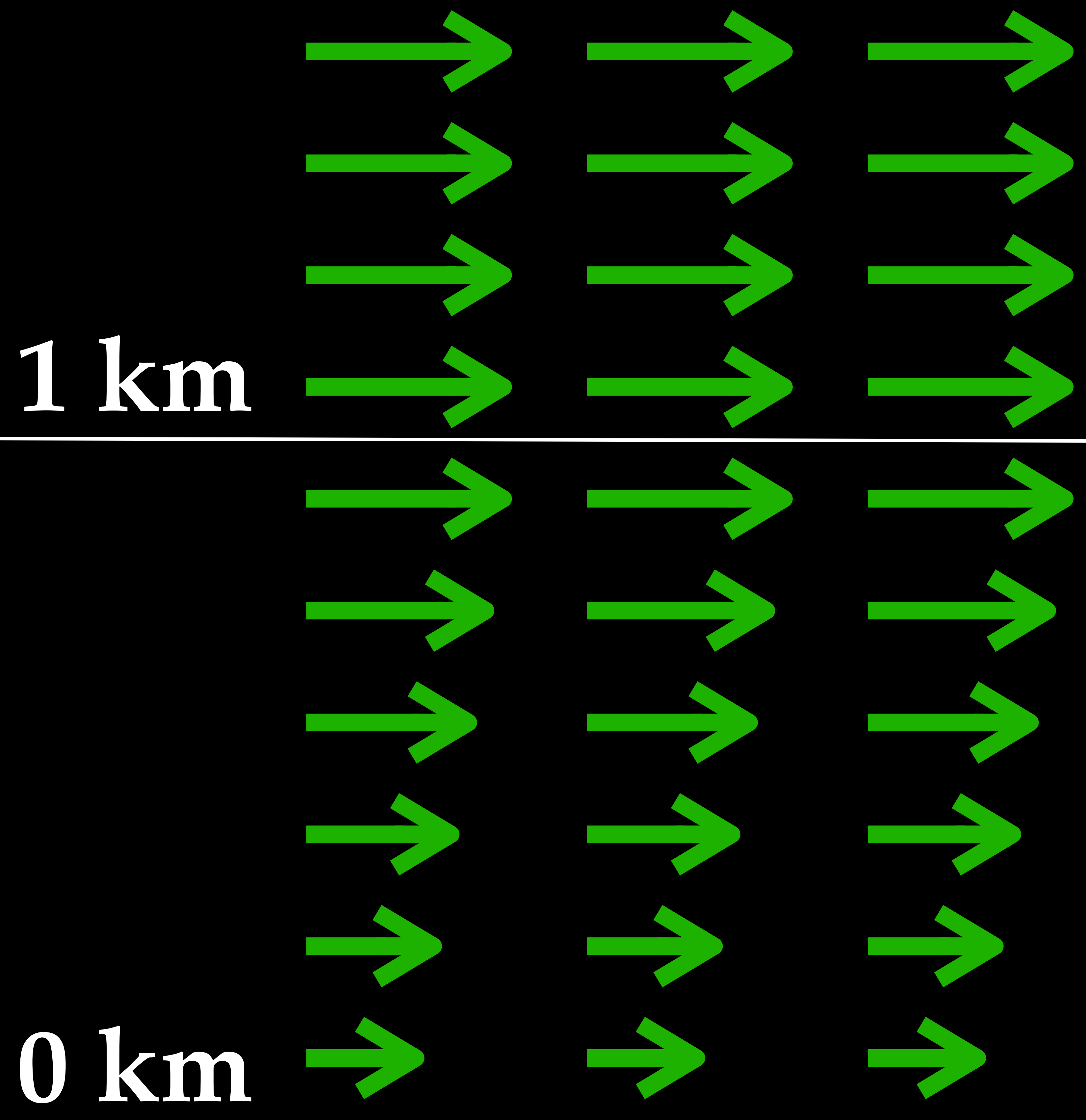
Bosch and Kleissl (2013) assumed **linear** C motion.

They used a network of **ground-mounted** irradiance sensors on the site of a PV plant.

$\vec{v}$ direction was obtained from edge detection of transient C, and $\vec{v}$ speed was computed by correlating time lags in PV panels **power output**.

The Frictional Force

(Petersen et al., 1998a)



$$\overrightarrow{\textbf{Frictional Force}} \leq \mu\vec{v}$$

The Frictional Force

(Petersen et al., 1998a)

The Centrifugal Force

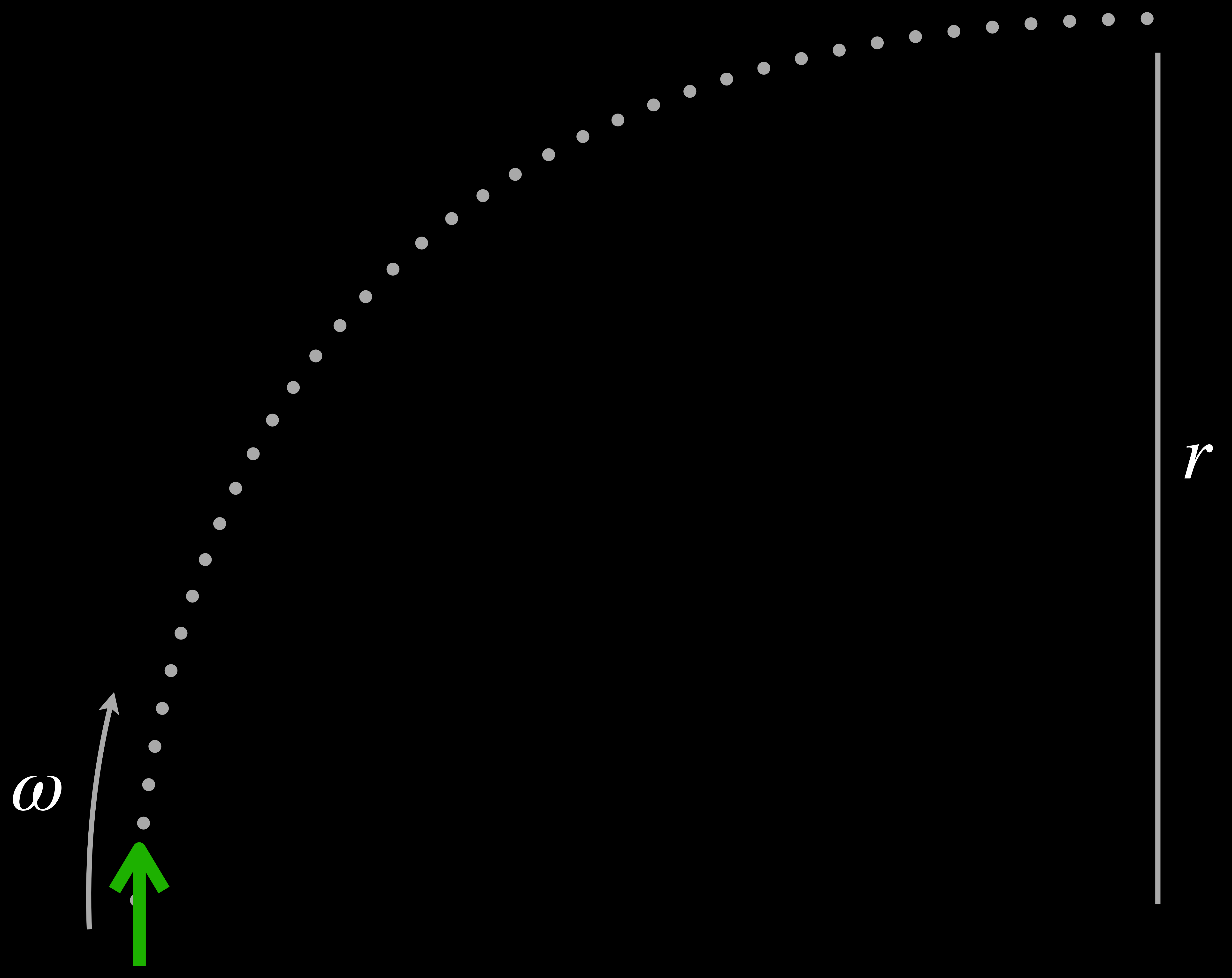
(Weidner and Sells, 1973)

$$\overrightarrow{\textbf{Centrifugal Force}} = m\omega^2 r$$

m – **air mass**

ω – W **angular velocity**

r – **curved path's radius**



The Centrifugal Force

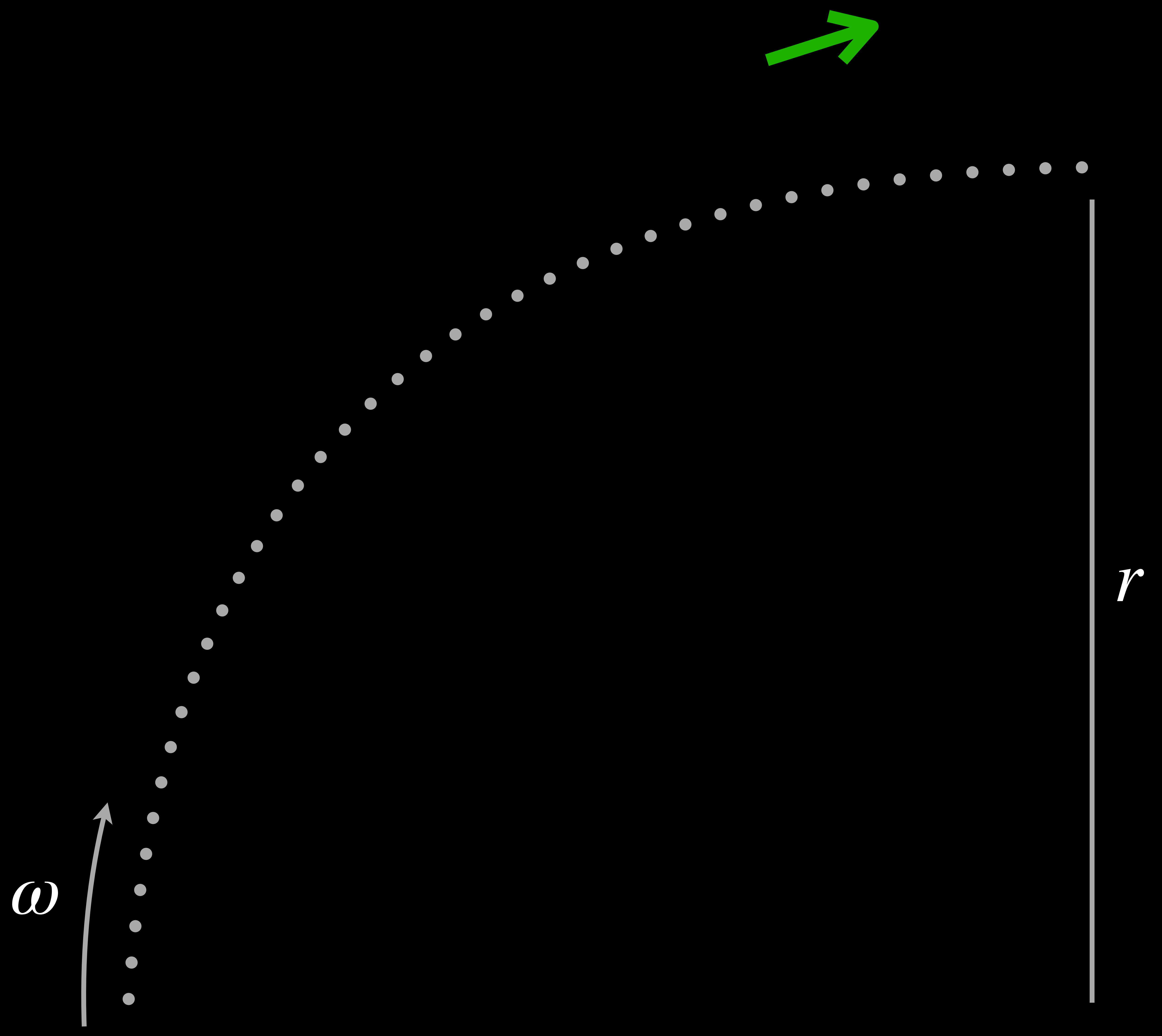
(*Weidner and Sells, 1973*)

$$\overrightarrow{\textbf{Centrifugal Force}} = m\omega^2 r$$

m – **air mass**

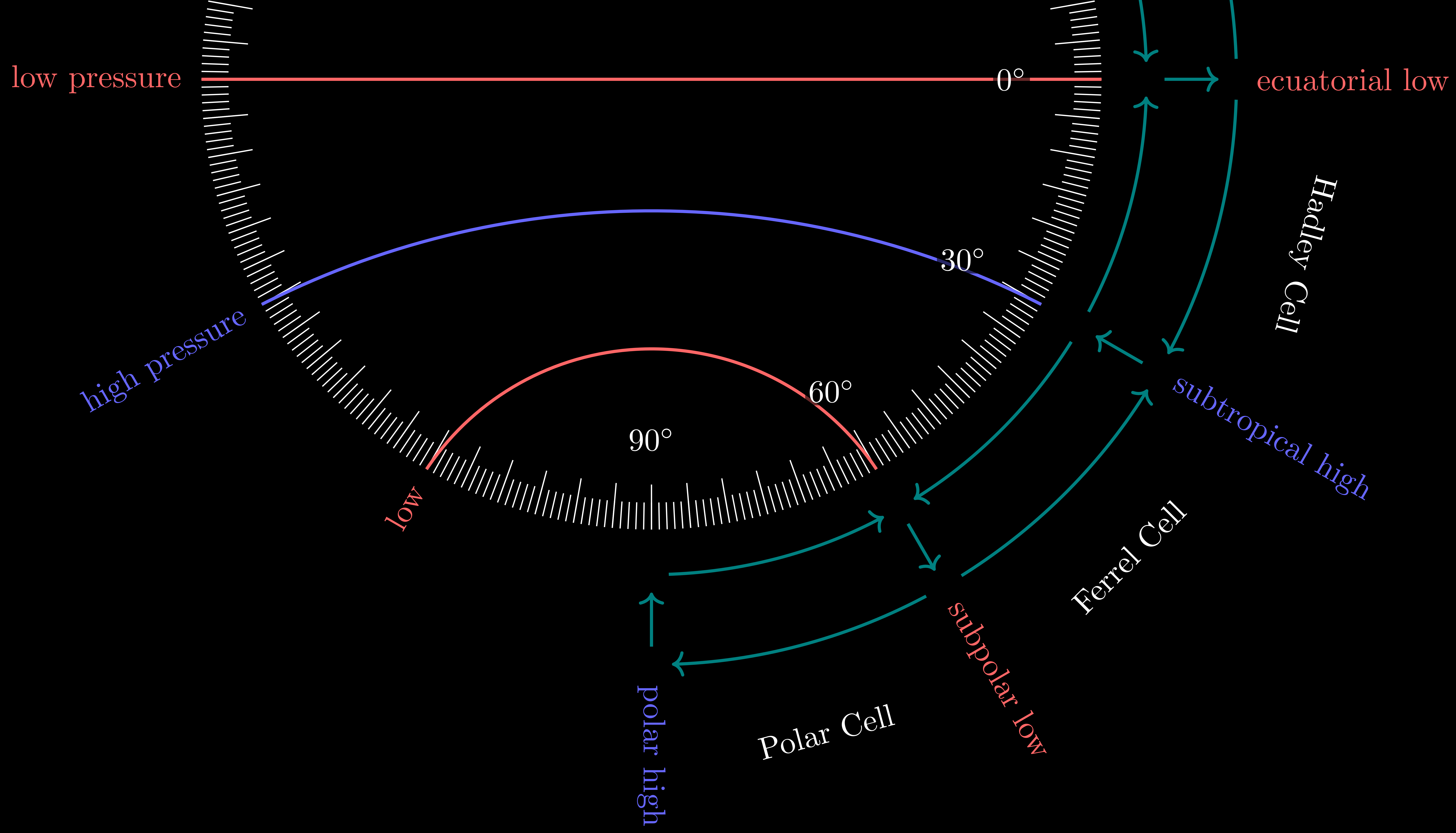
ω – W **angular velocity**

r – **curved path's radius**



The Centrifugal Force

(Cushman-Roisin and Beckers, 2011; Nanda, 2018)



low pressure
0°
ecuatorial low
Hadley Cell
30°
high pressure
subtropical high
60°
90°
low
Ferrel Cell
subpolar low
Polar Cell
polar high

Assumption 1

I assume W is a product of the global forces (except for the Frictional Force), but **not affected** by terrain roughness, topography, and tall structures.

Assumption 2

(Sun et al., 2018; Nouri et al., 2019)

I assume C are at the appropriate altitude to be transported by large-scale W, and C movement is **not influenced** by other phenomena except for W.

(Sun et al., 2018; Nouri et al., 2019)

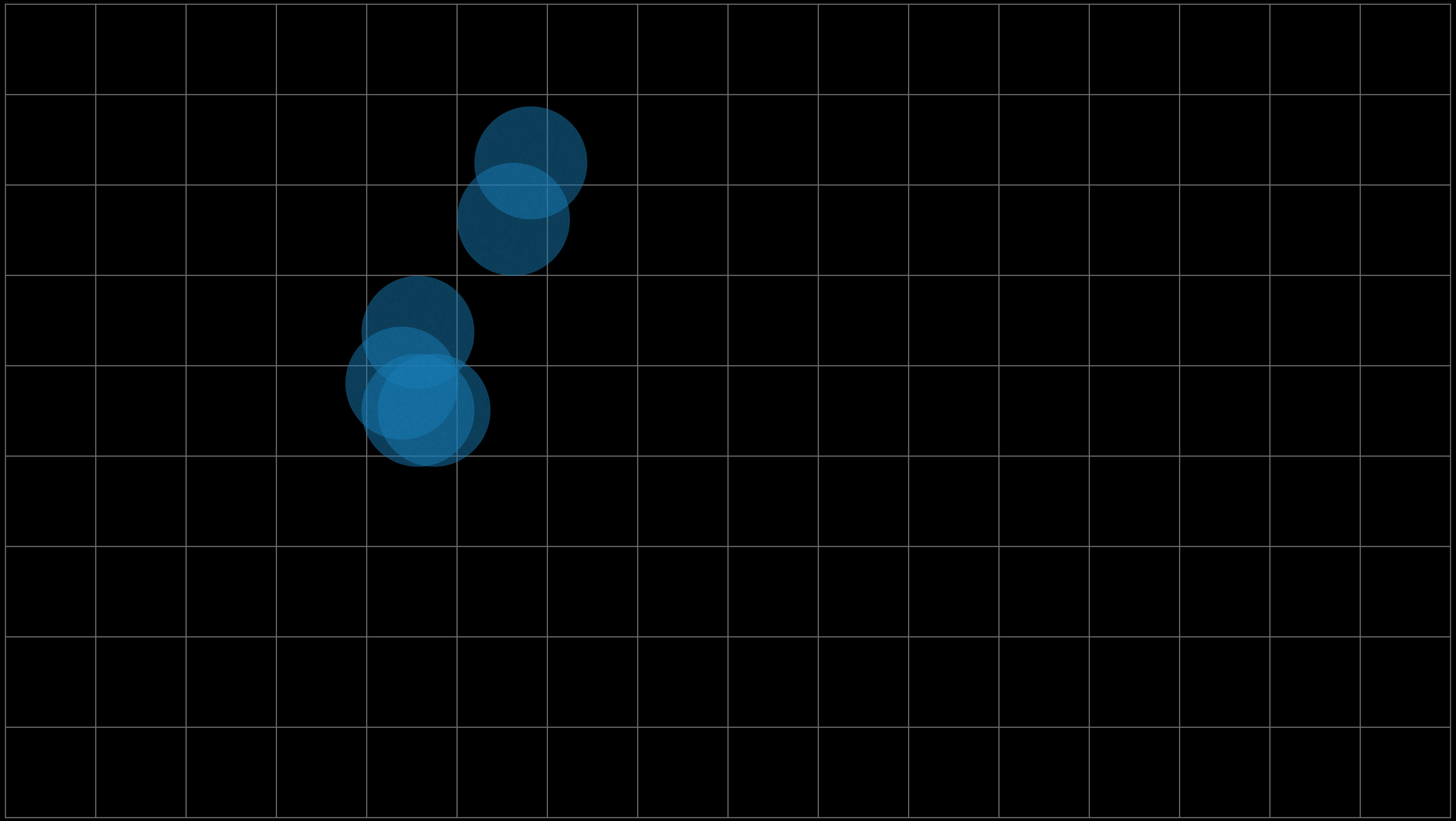
I assume C are at the appropriate altitude to be transported by large-scale W, and C movement is **not influenced** by other phenomena except for W.

(Farnebäck, 2003)

I assume W velocity and direction of movement are attainable from C motion by applying the **Optical Flow Procedure** to construct the W_{map} using $\vec{v}$.

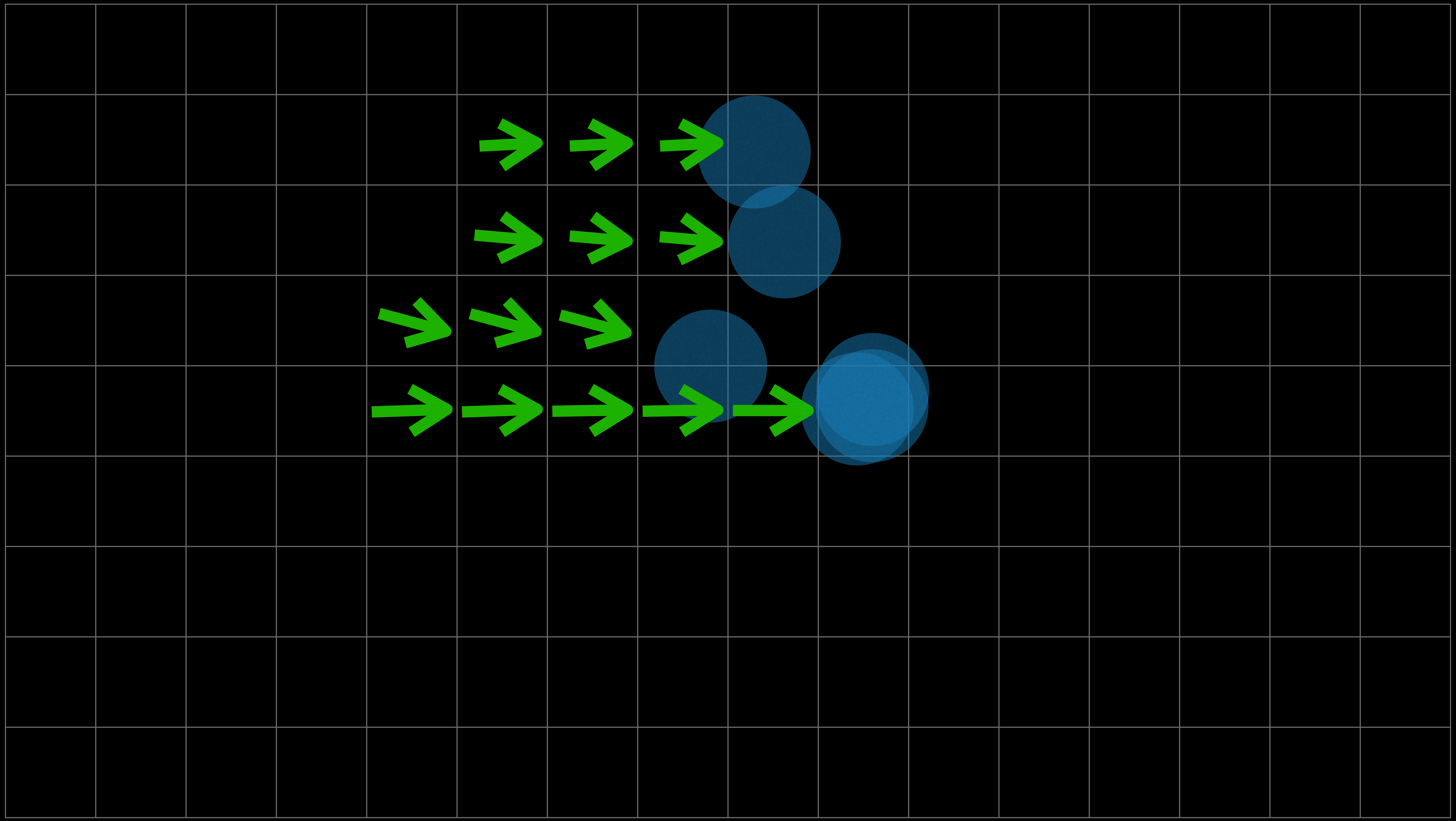
Assumption 3 • Optical Flow Example

(Farnebäck, 2003)



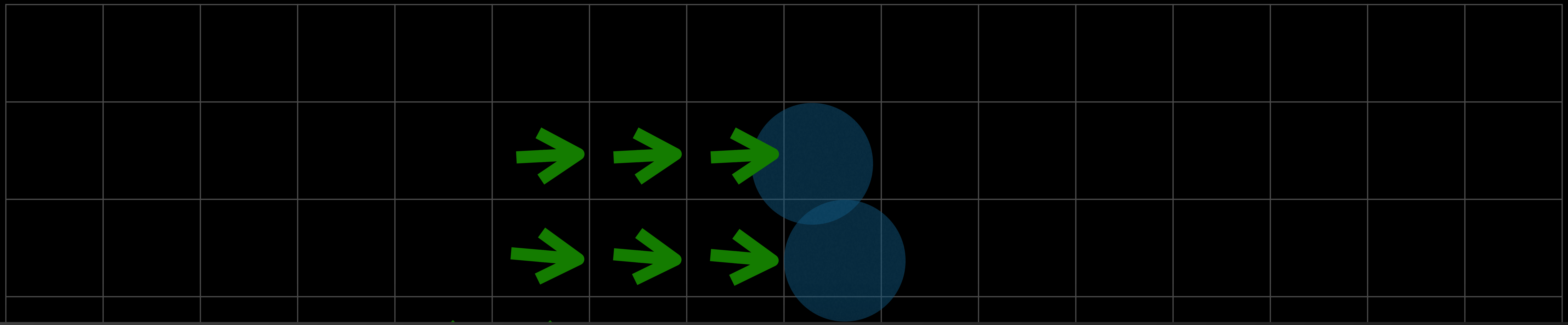
Assumption 3 • Optical Flow Example

(Farnebäck, 2003)



footnote

 (Farnebäck, 2003)



Marquez and Coimbra, (2013), and Zaher et al. (2017) believe $\vec{v}$ can be obtained from sky cameras or ground-mounted irradiance sensors.

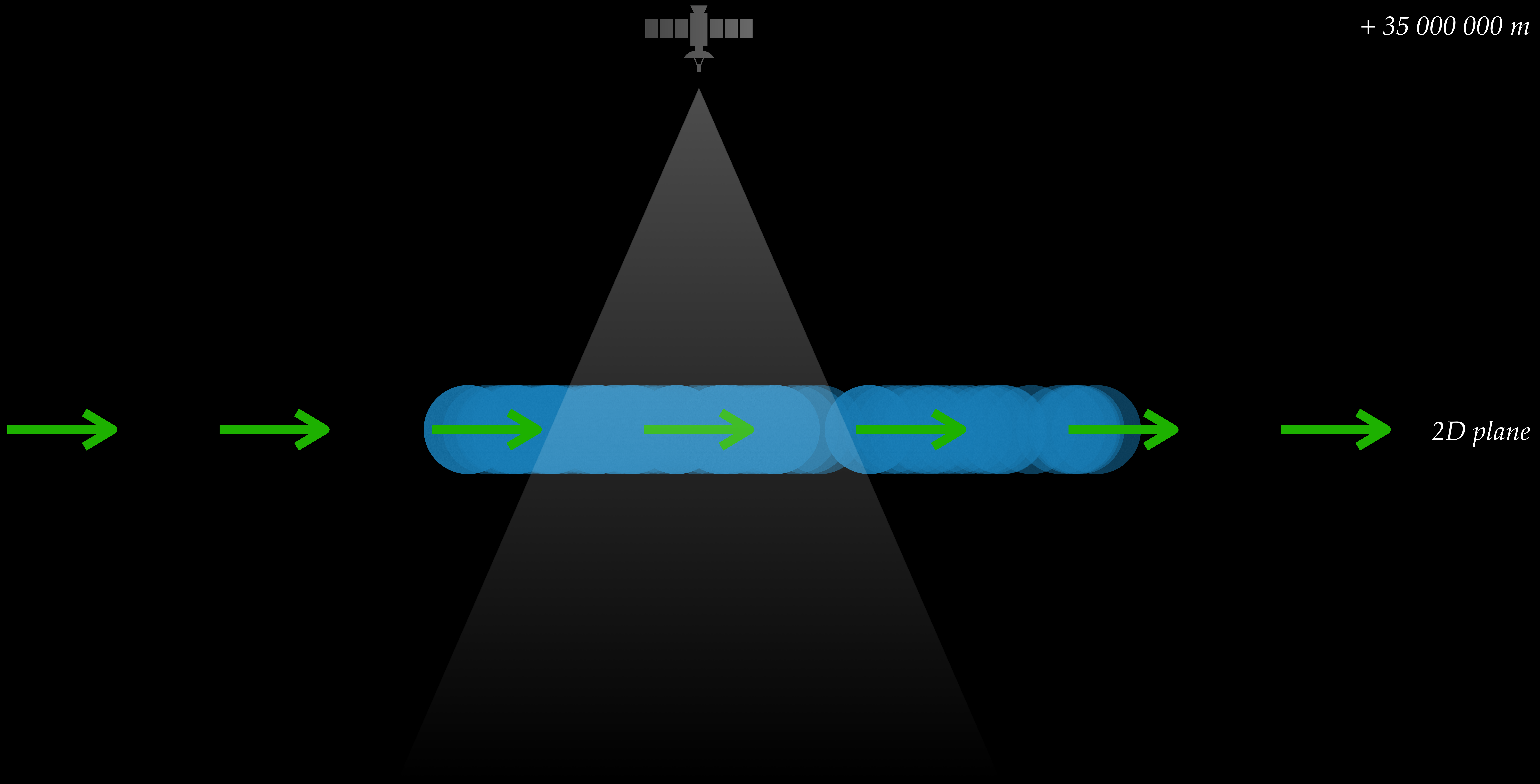
They **omitted** geostationary satellite imagery from their study.

Assumption 4

I assume W and C exist on the same **2D plane** from the perspective of the satellite's sensor.

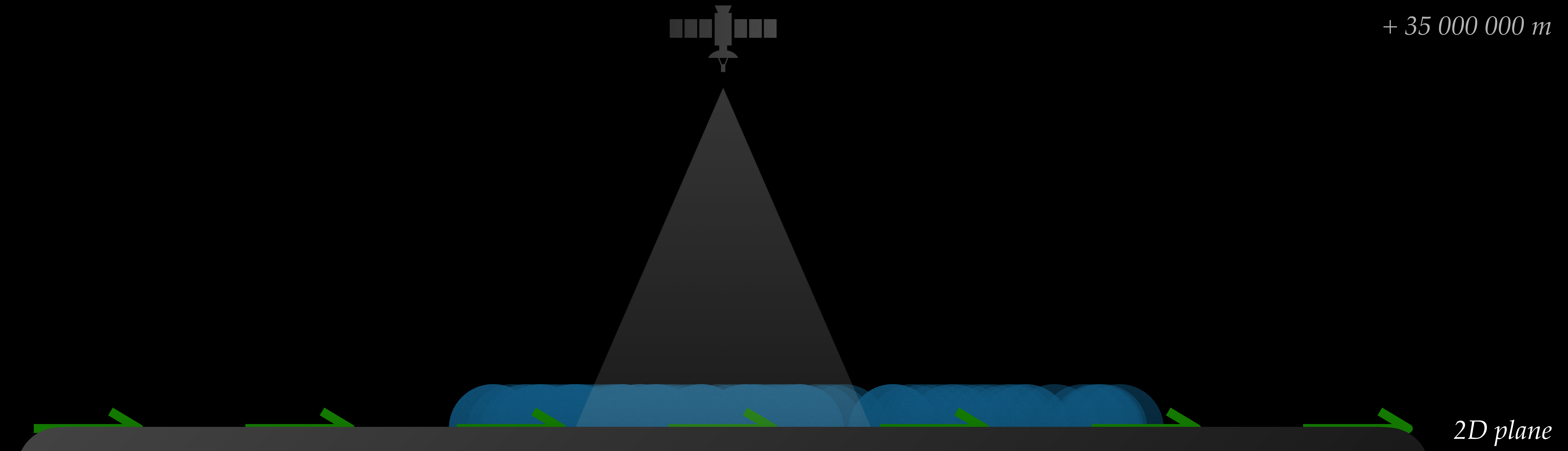
Assumption 4 • 2D Plane

Assumption 4 • 2D Plane



footnote

Assumption 4 • 2D Plane



Nouri et al. (2019) measured **average cloud heights** between 2 and 12 km altitude.

Nanda, (2018) writes about atmospheric processes claiming **wind is present** up to 16 km altitude.

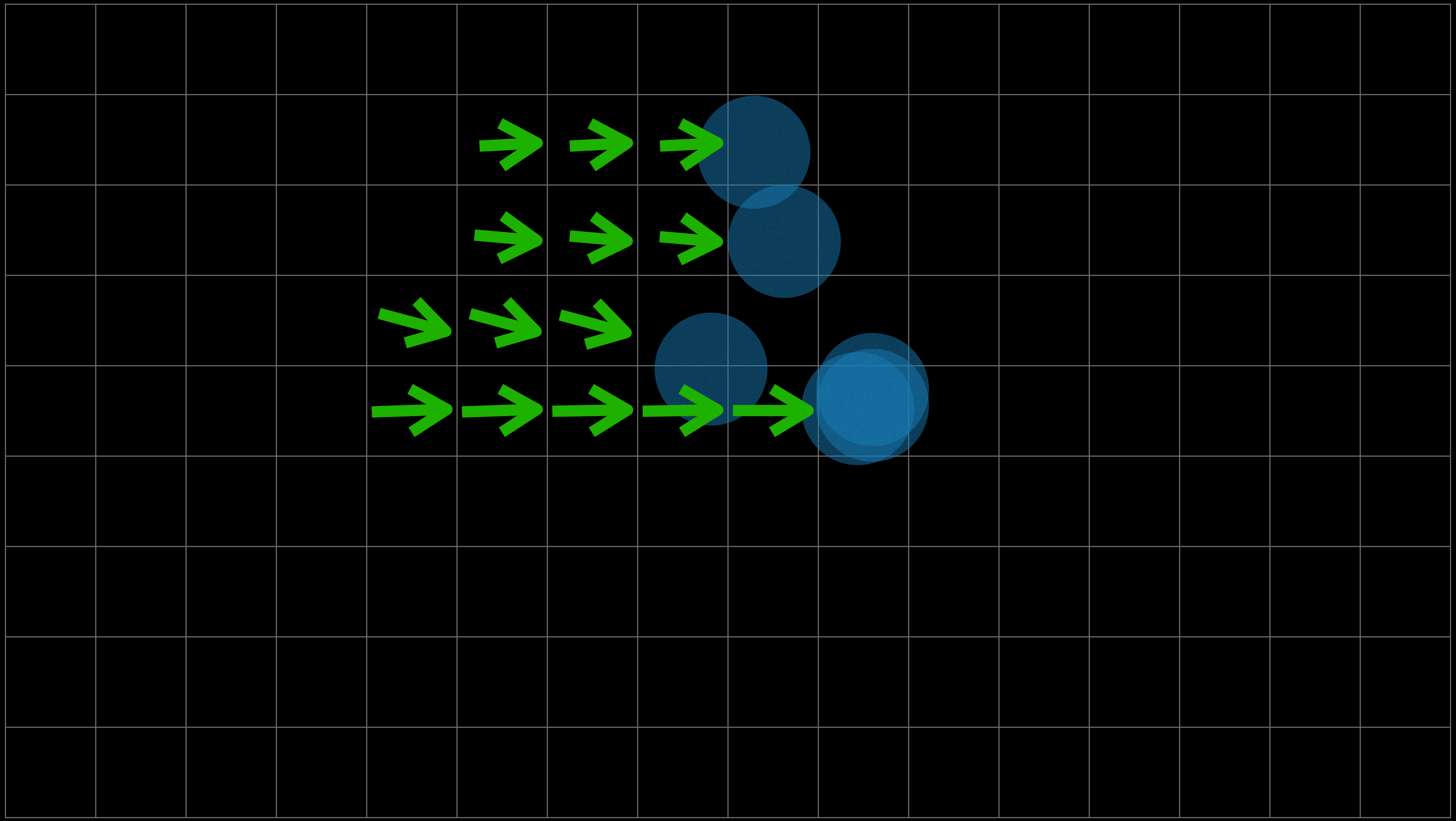
Assumption 5 *(Cushman-Roisin and Beckers, 2011; Nanda, 2018; Coriolis, 1832 & 1835)*

I assume W is constant and flows continuously in the **same directions** because of the aforementioned forces and the global air circulation system.

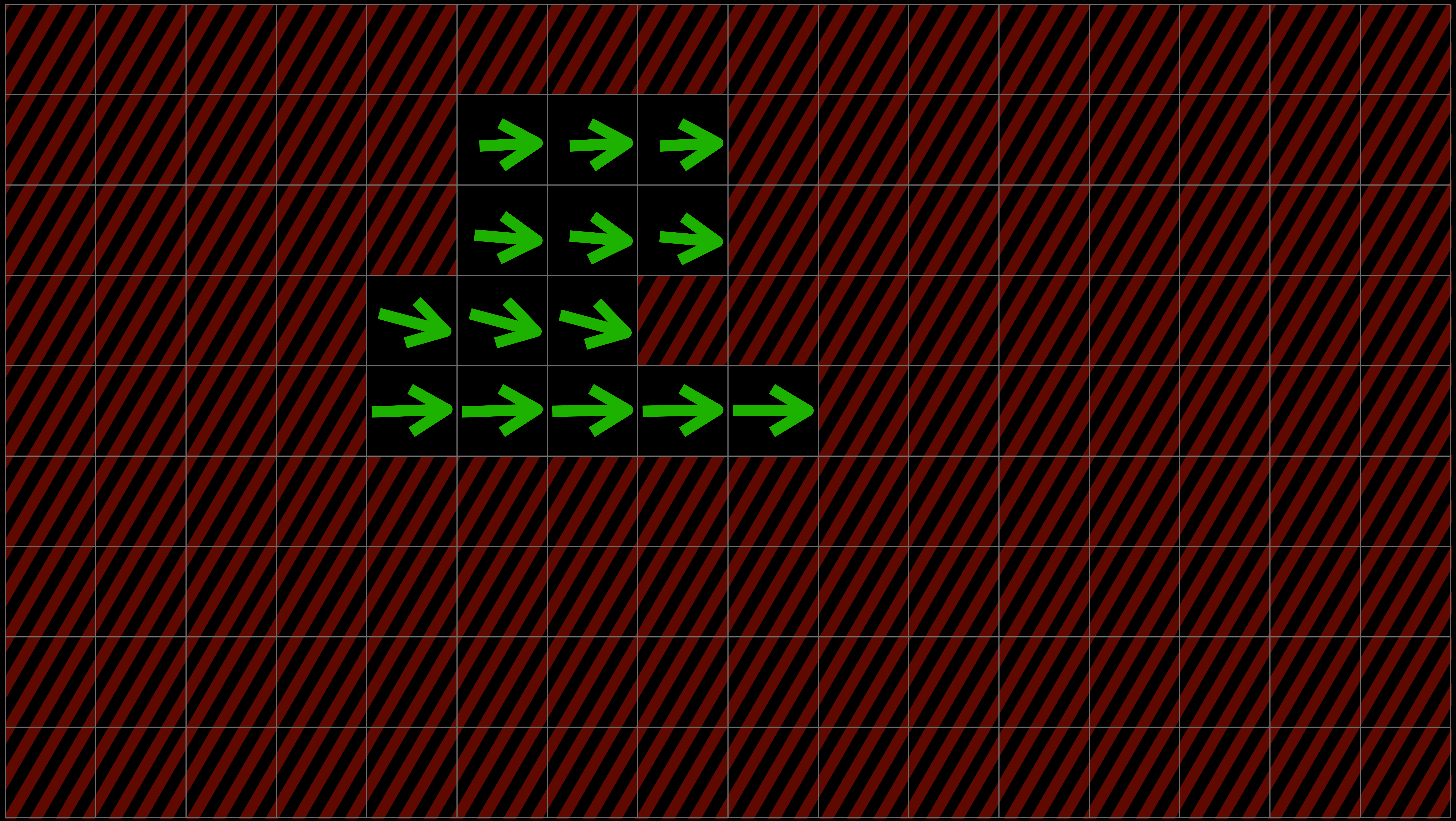
I assume W is smooth.
As pressures gradually come to an equilibrium, changes in W speed and direction over an amount of time is **smoother** rather than abrupt. The greater the pressure difference, the stronger the W is.

I assume W is present everywhere on the W_{map} even if it does not contain $\vec{v}$ in p. The gaps in the map can be **filled in** using $\vec{v}$ found in other regions of S.

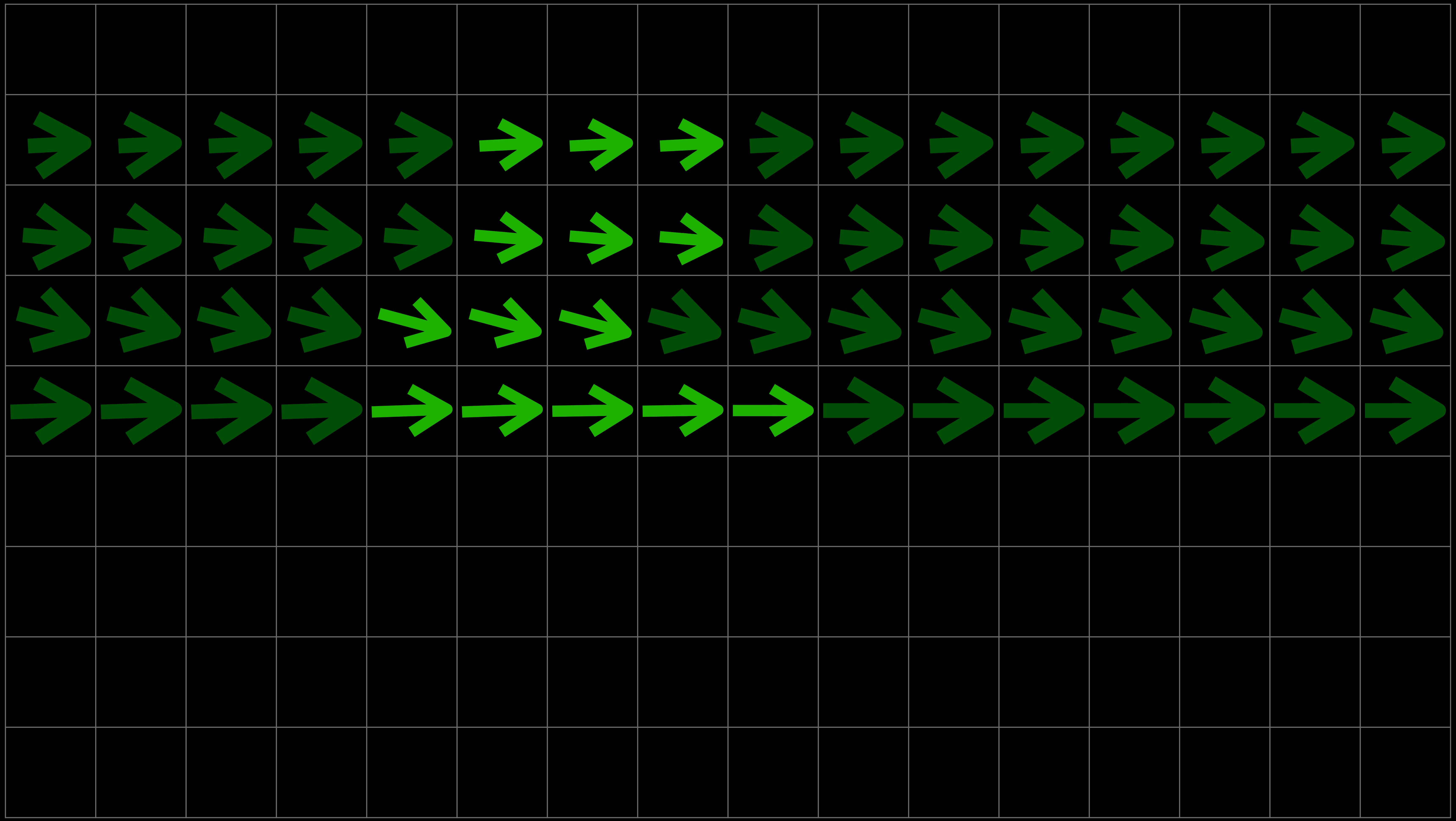
Assumption 7 • Wind Map Example



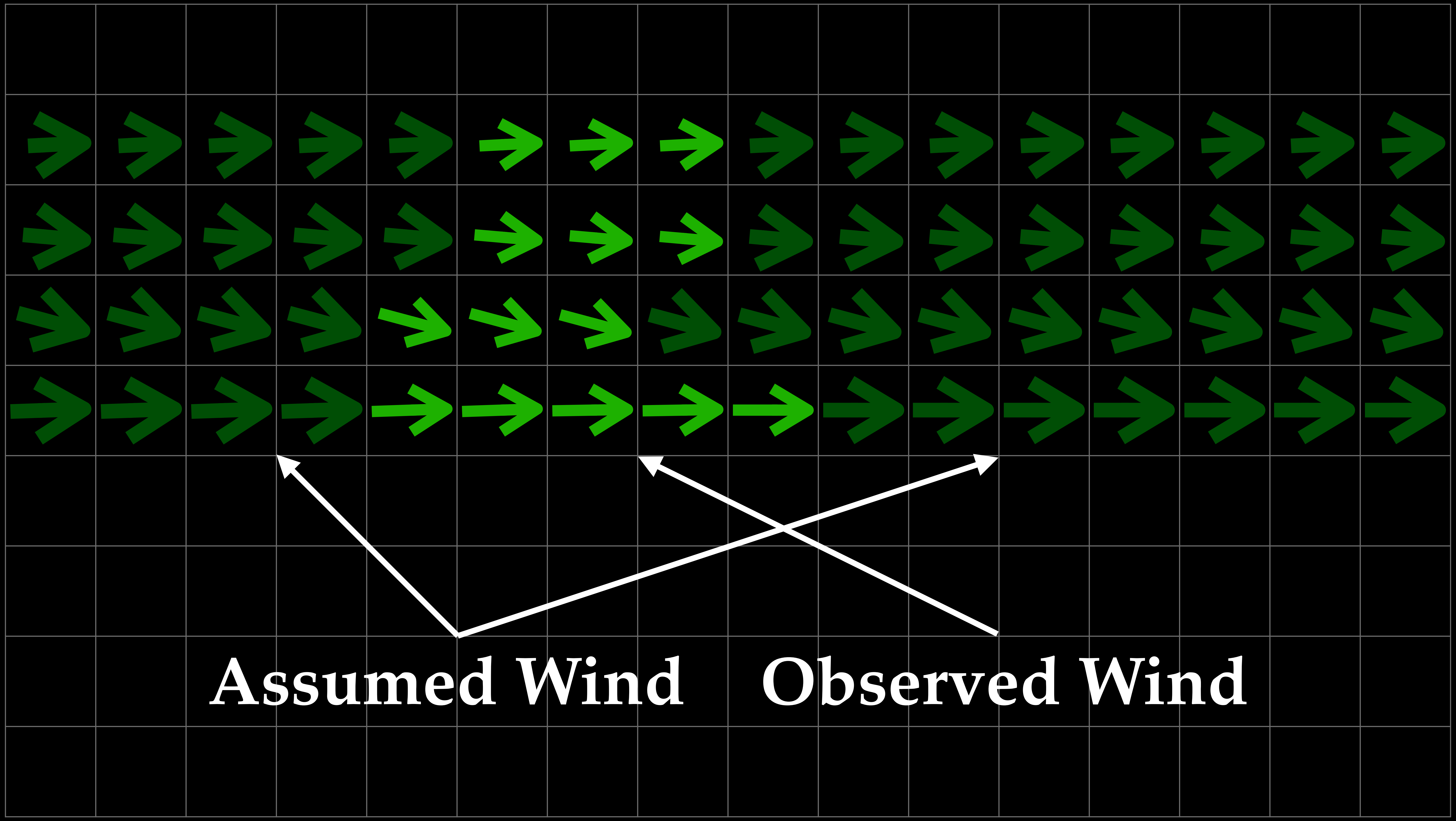
Assumption 7 • Wind Map Example



Assumption 7 • Wind Map Example



Assumption 7 • Wind Map Example



Contents

Simulate groups of birds, fish, and animals;

The Flocking Behavior

(Reynolds, 1987)

Simulate groups of birds, fish, and animals;

Has origins in computer graphics;

Simulate groups of birds, fish, and animals;

Has origins in computer graphics;

Evolved from particle systems;

The Flocking Behavior

(Reynolds, 1987)

Simulate groups of birds, fish, and animals;

Has origins in computer graphics;

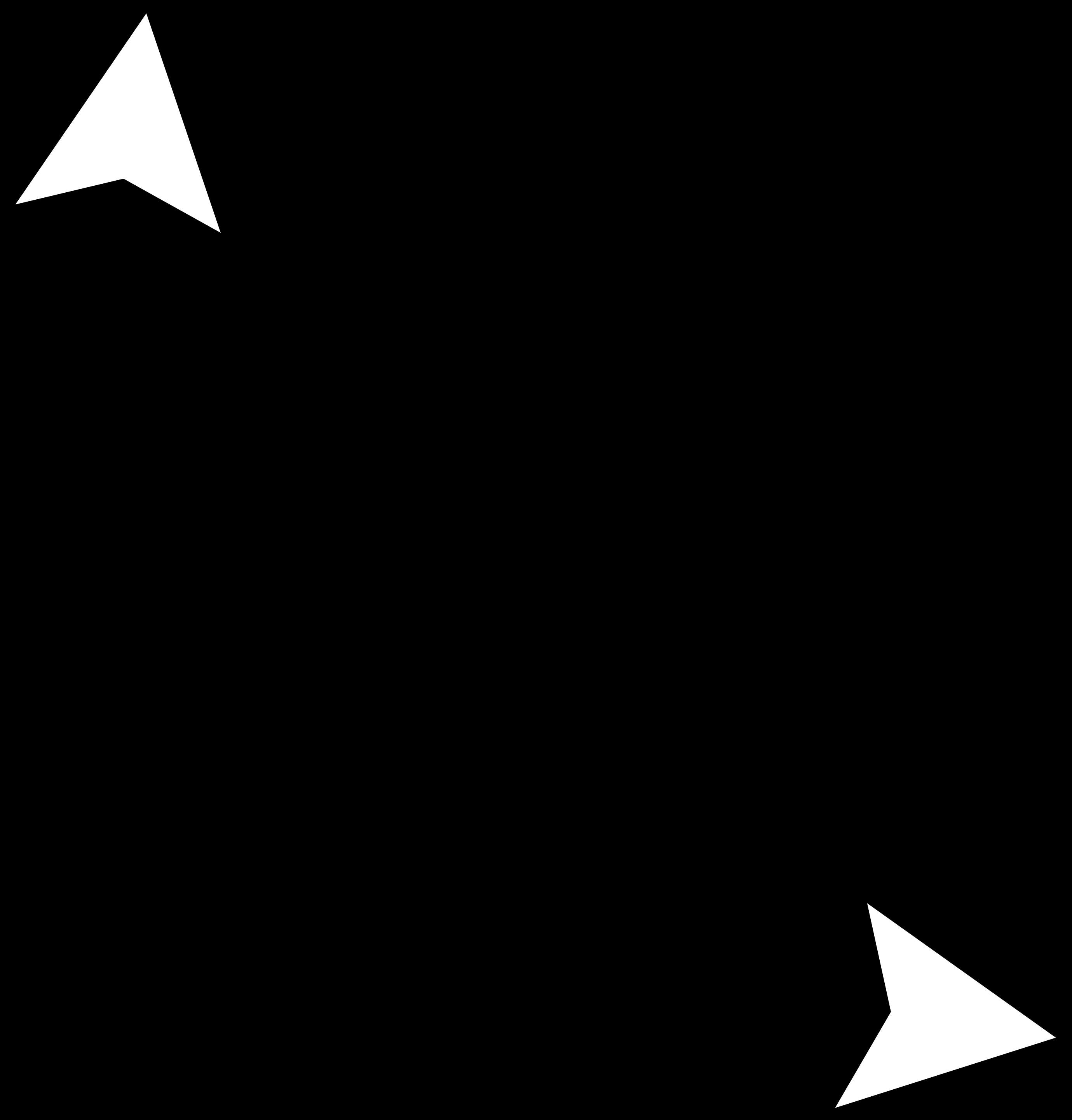
Evolved from particle systems;

Nature-inspired.

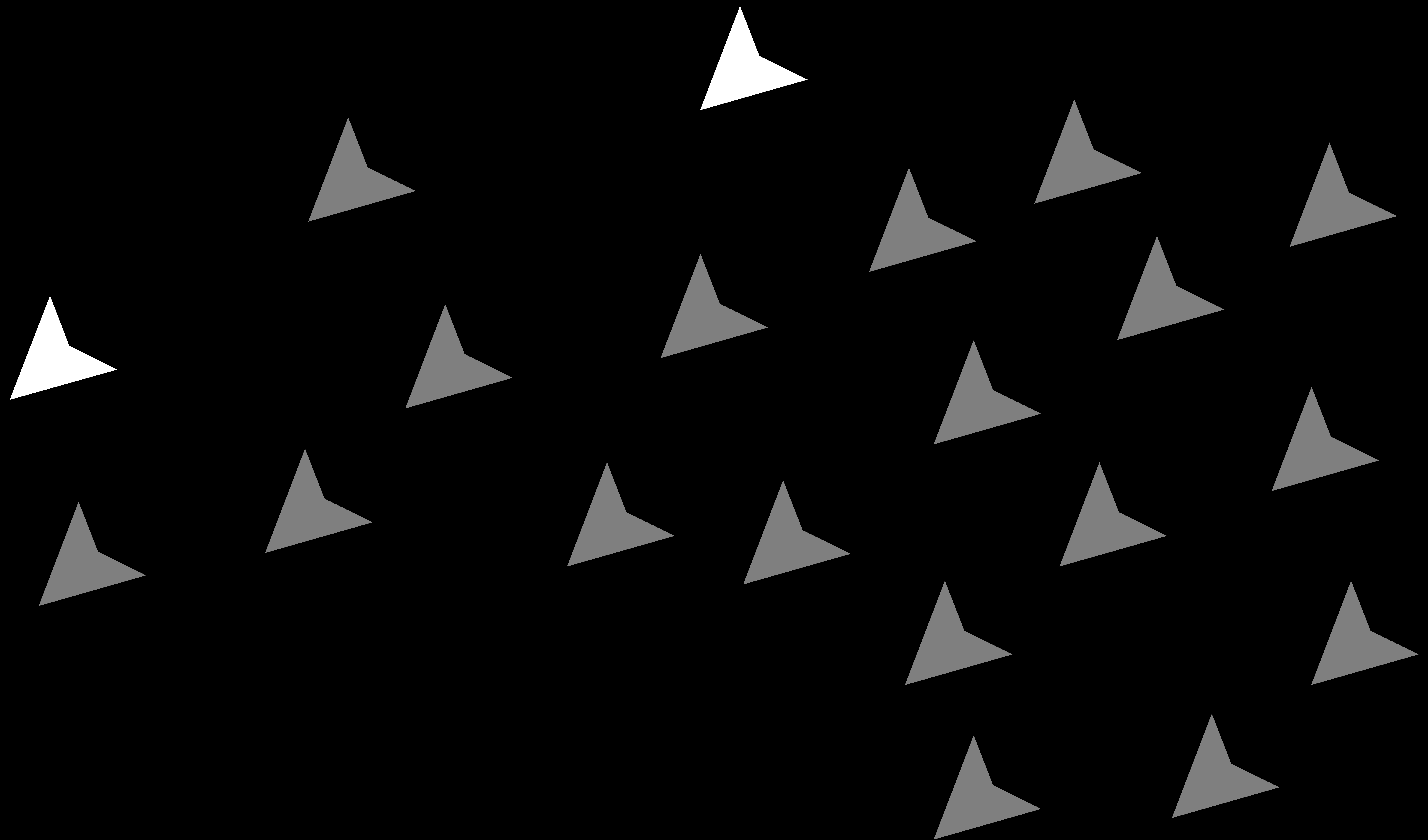
The Flocking Behavior

(Reynolds, 1987)

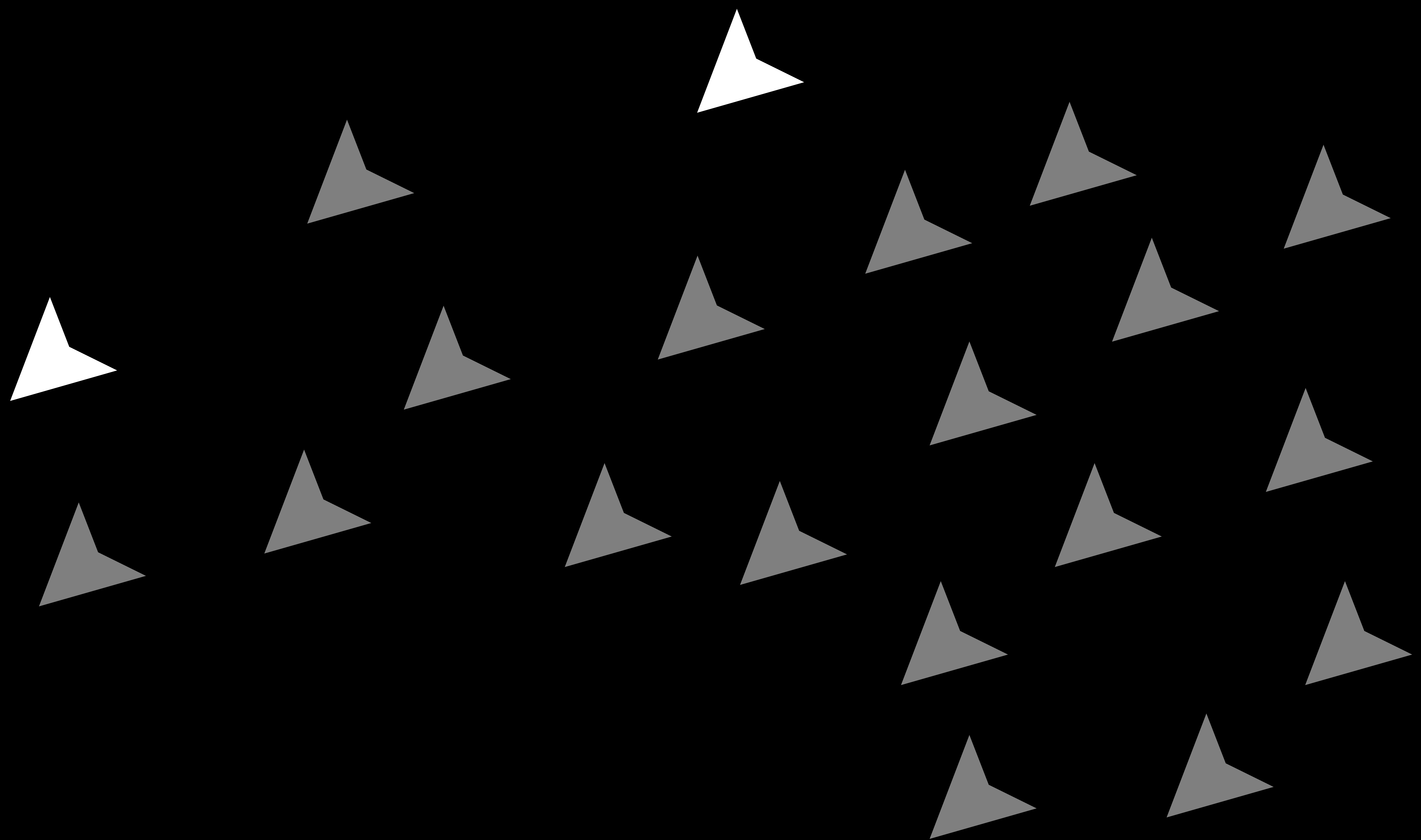
The Flocking Behavior • Collision Avoidance



The Flocking Behavior • Velocity Matching



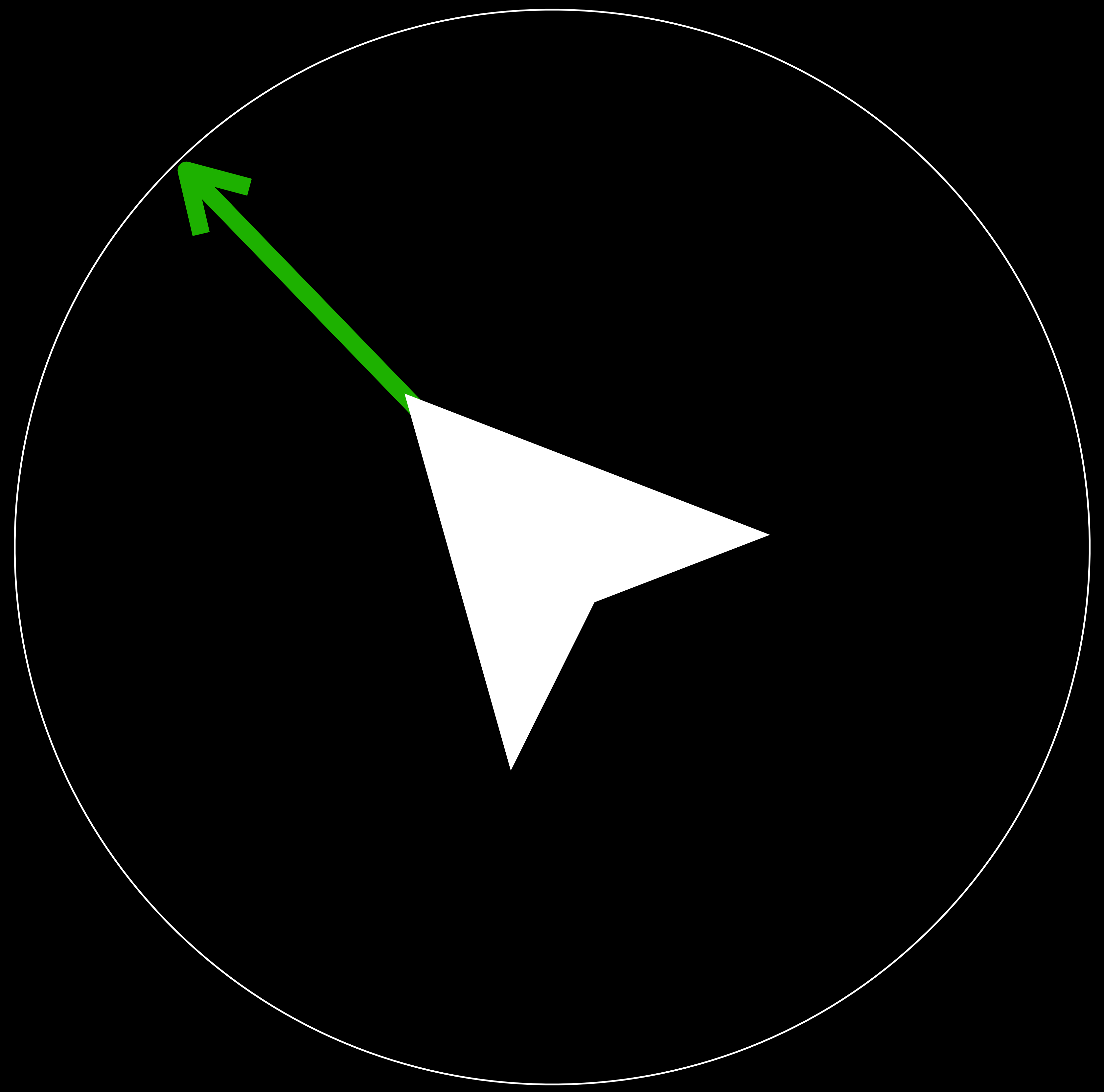
The Flocking Behavior • Flock Centering



Fixed radius

Minimum distance

Avoid impact

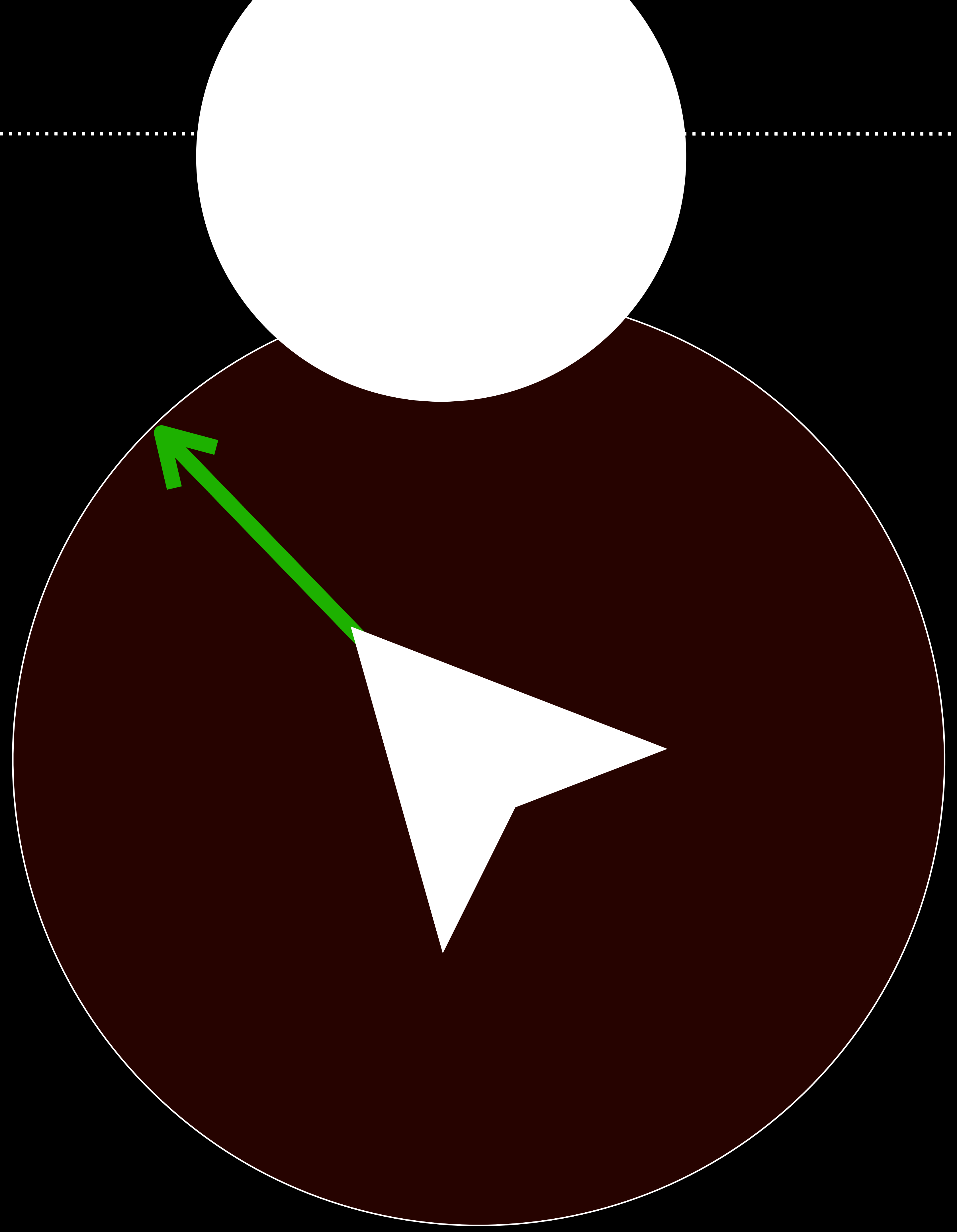


Collision Avoidance

Fixed radius

Minimum distance

Avoid impact

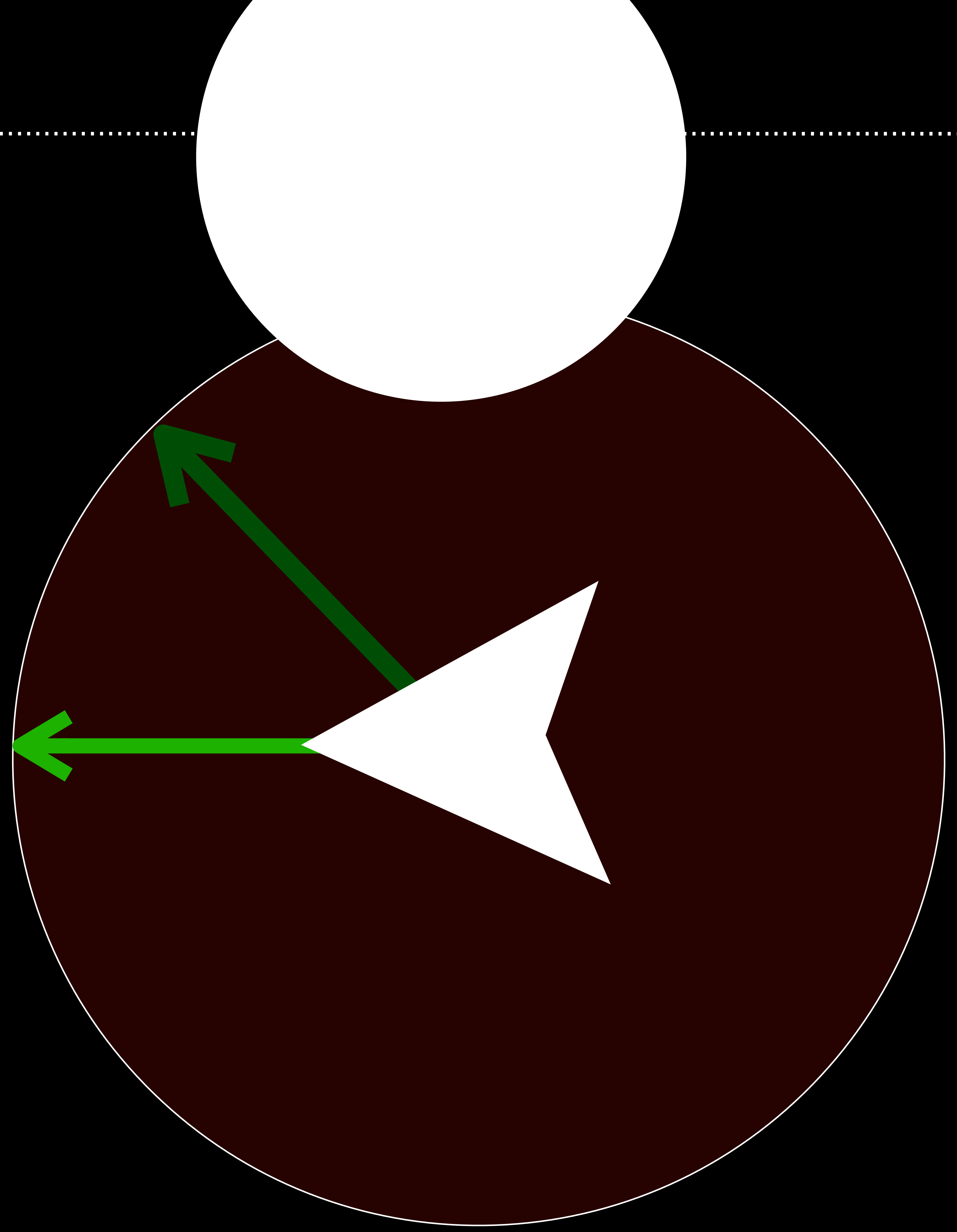


Collision Avoidance

Fixed radius

Minimum distance

Avoid impact

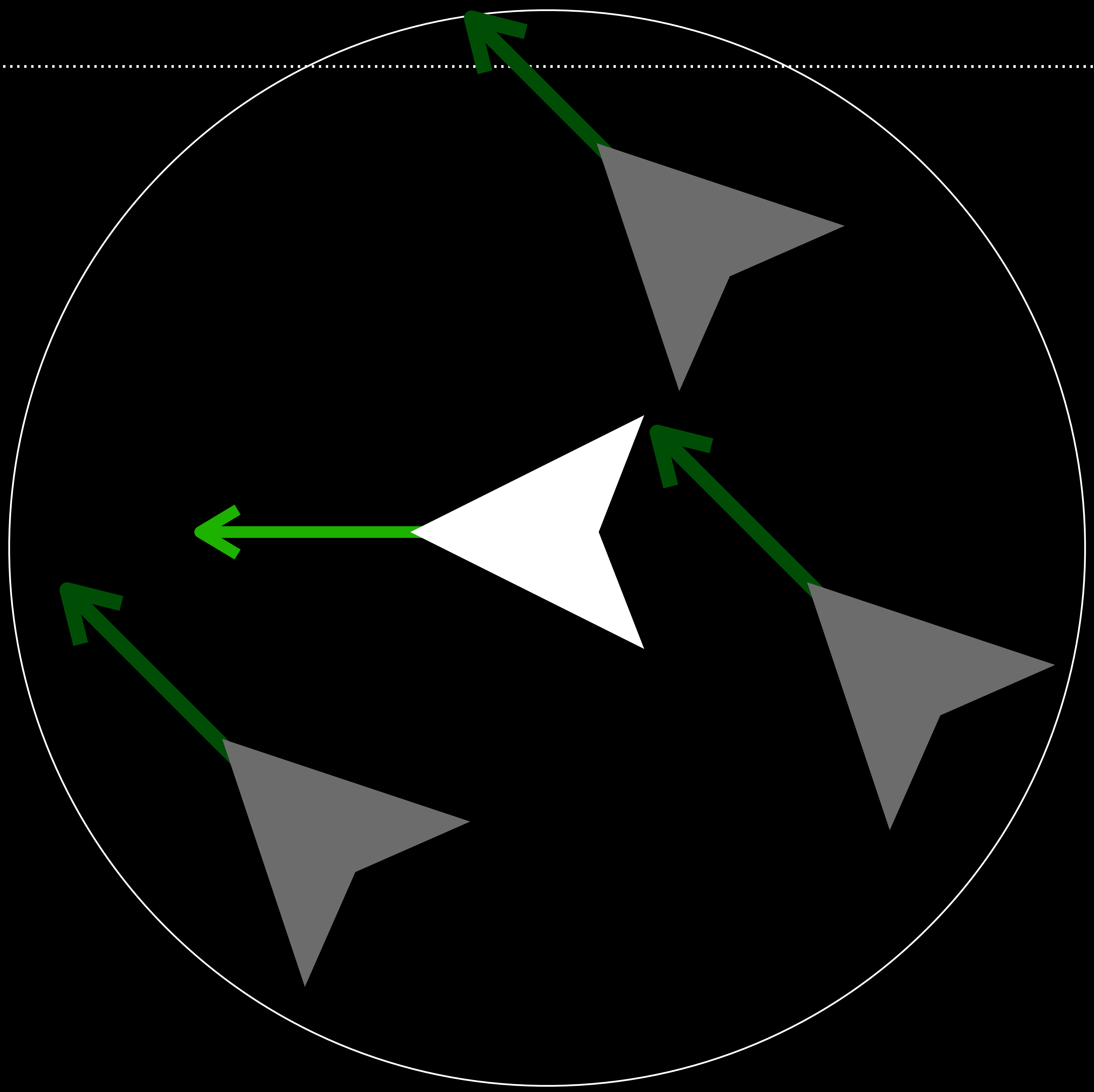


Velocity Matching

Neighborhood radius

Analyze neighbors

Match velocity

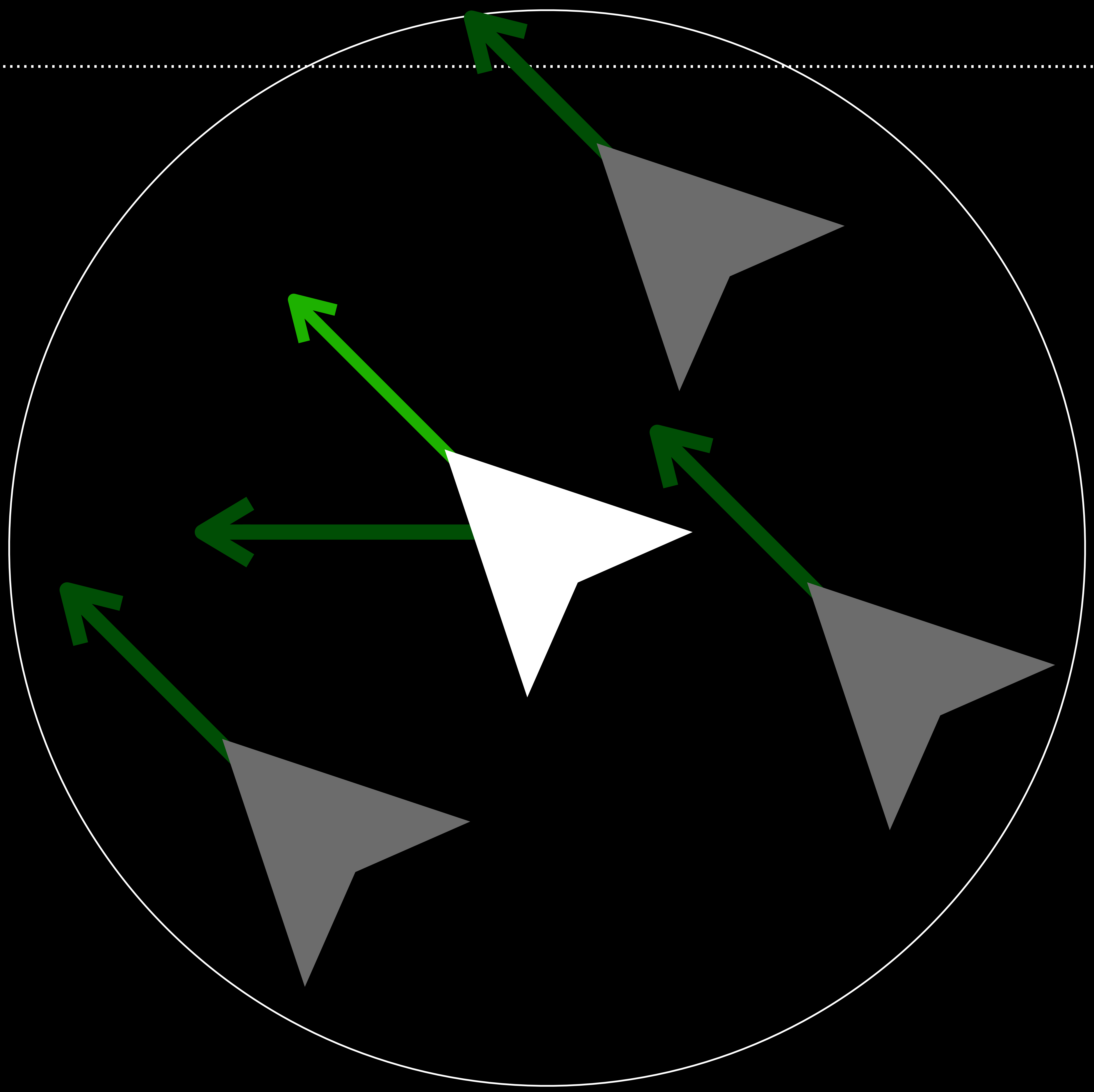


Velocity Matching

Neighborhood radius

Analyze neighbors

Match velocity

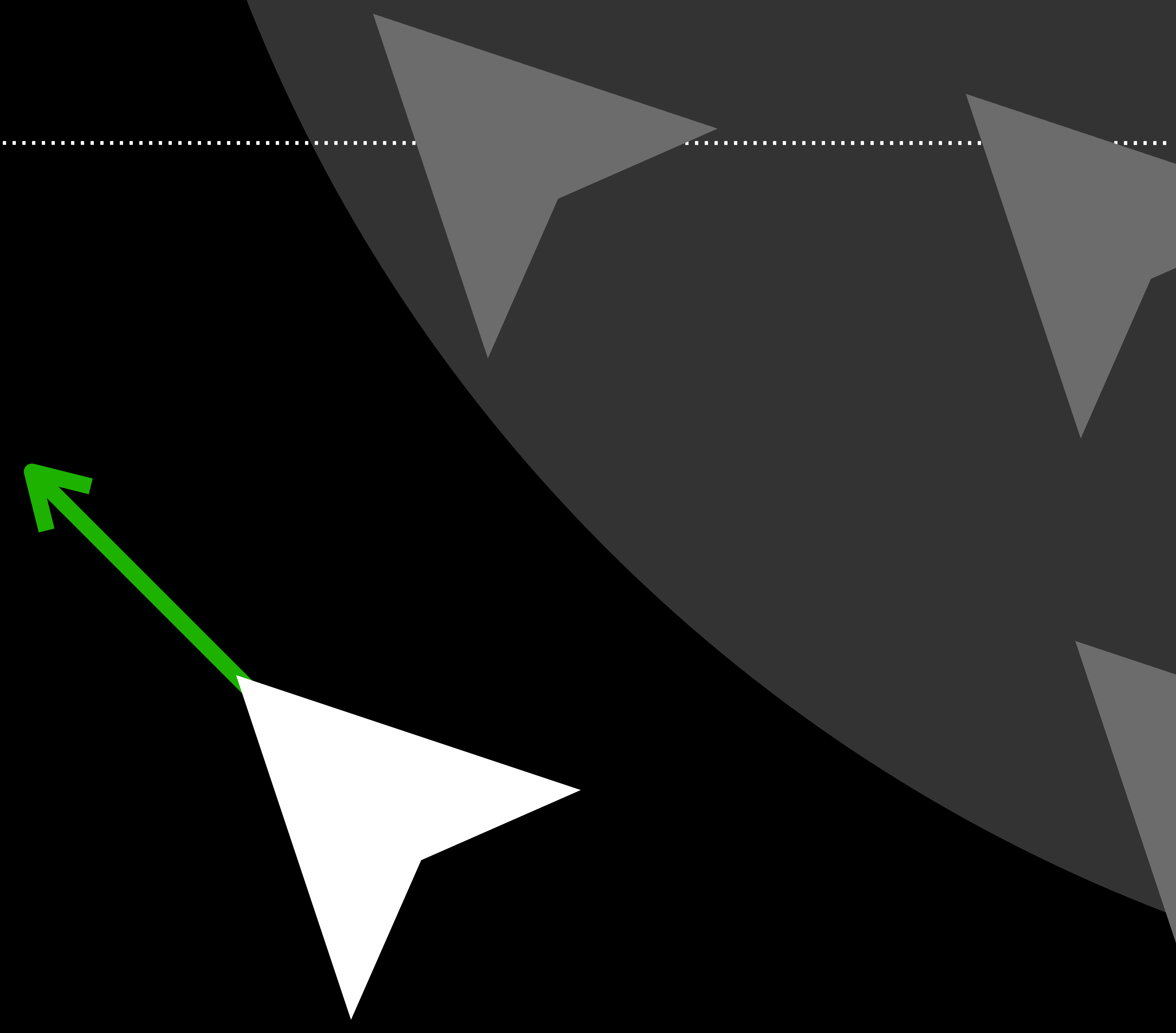


Flock Centering

Flock boundary

Center of mass

Move closer

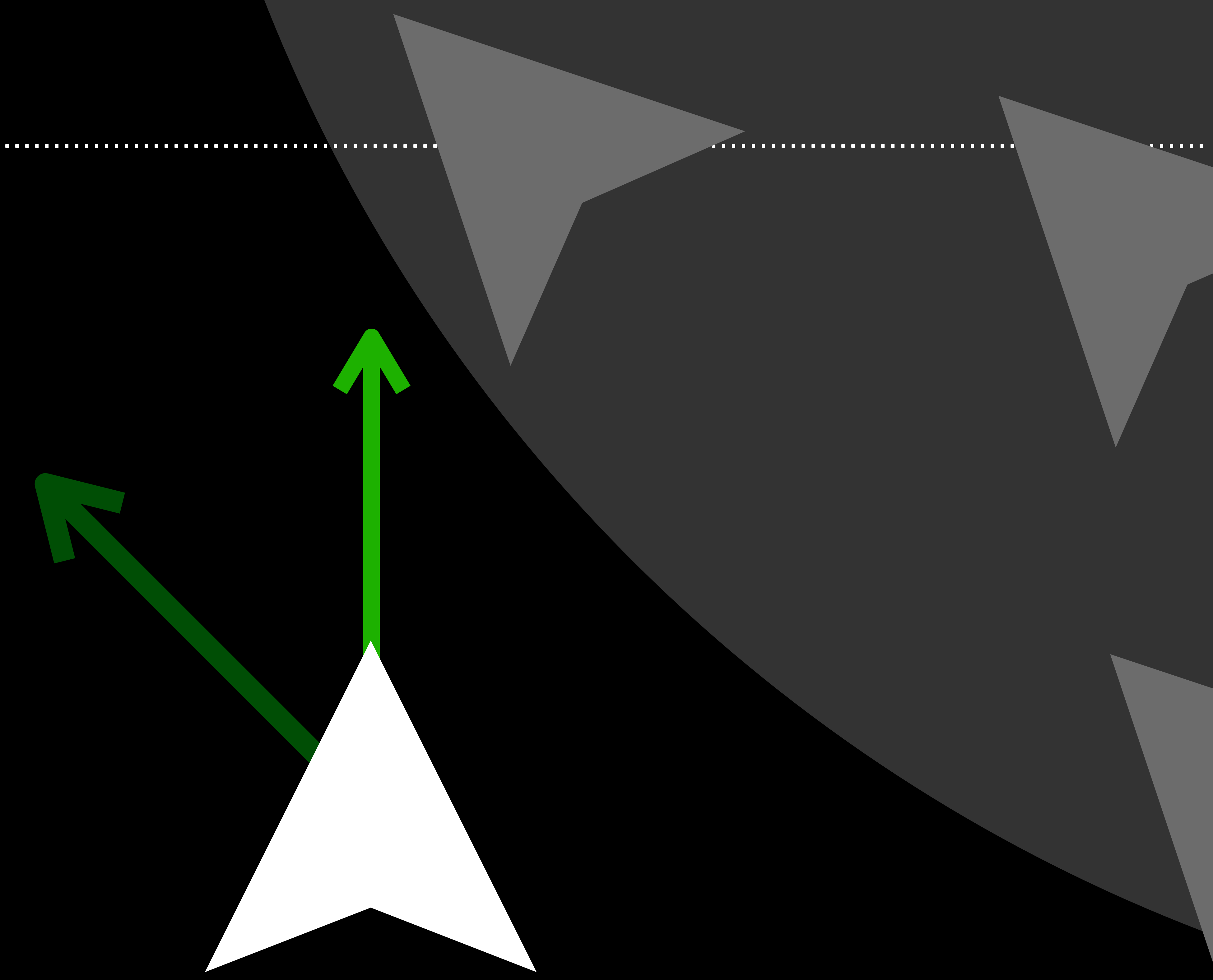


Flock Centering

Flock boundary

Center of mass

Move closer

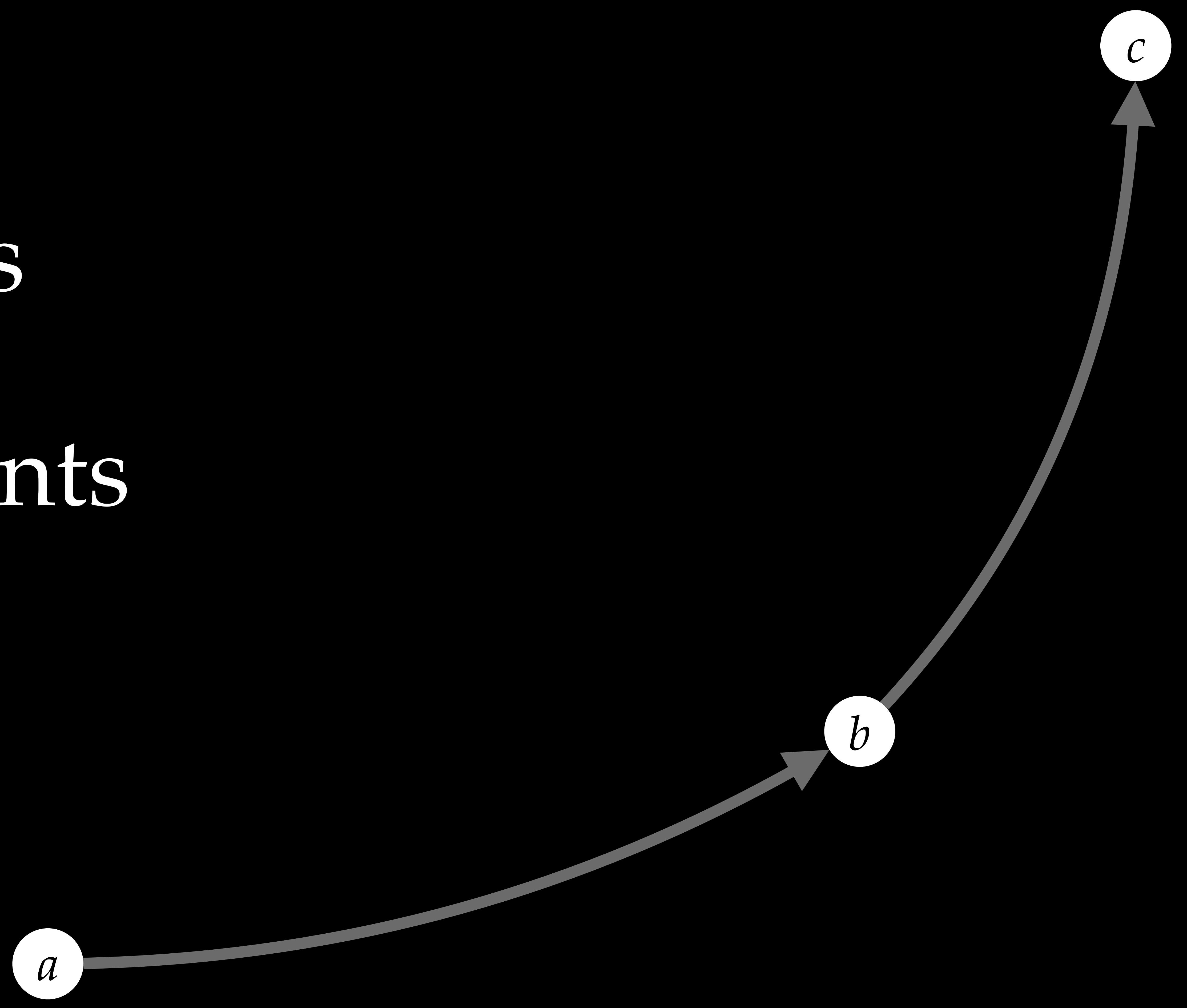


Migratory Behavior

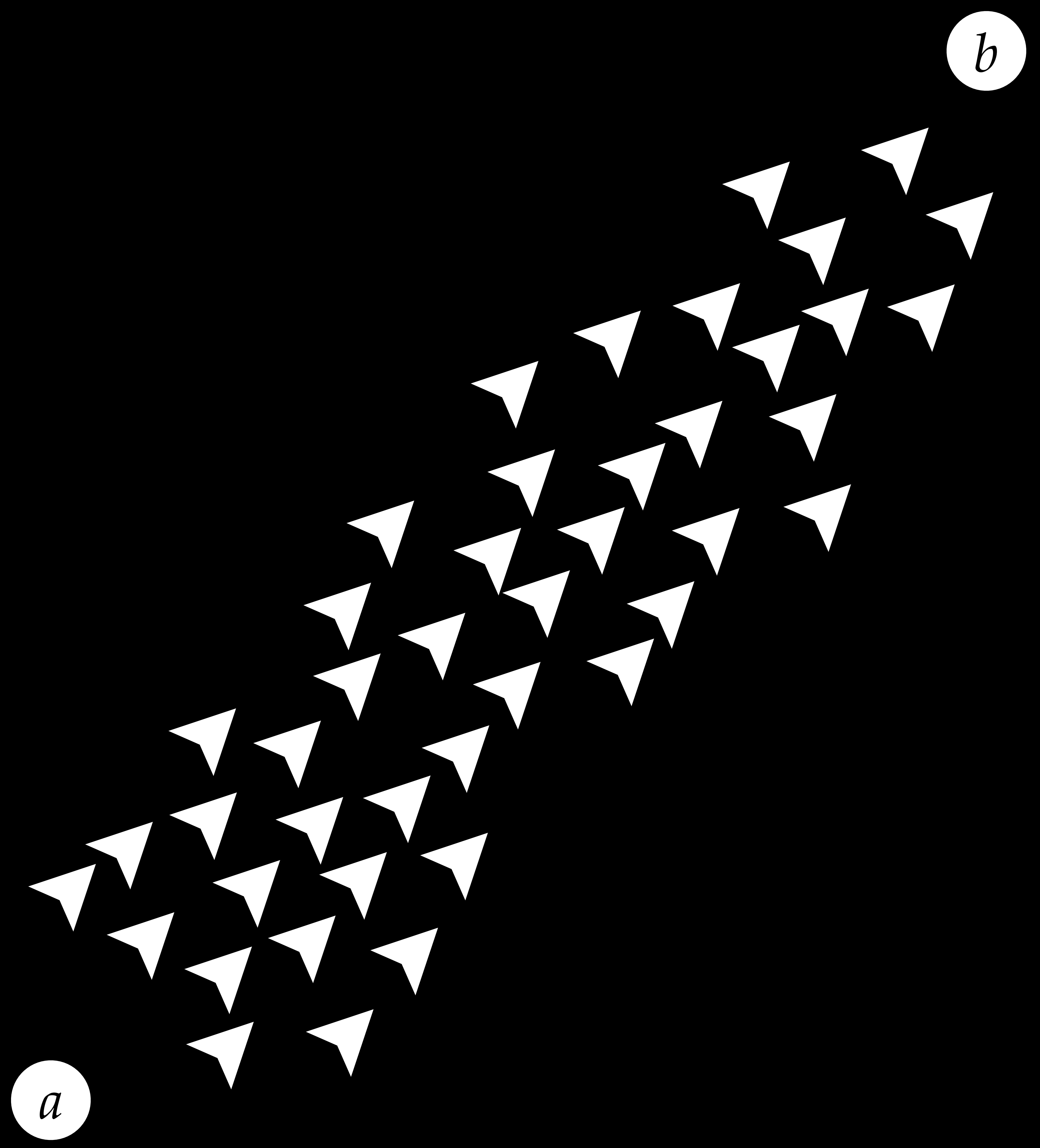
Global movement targets

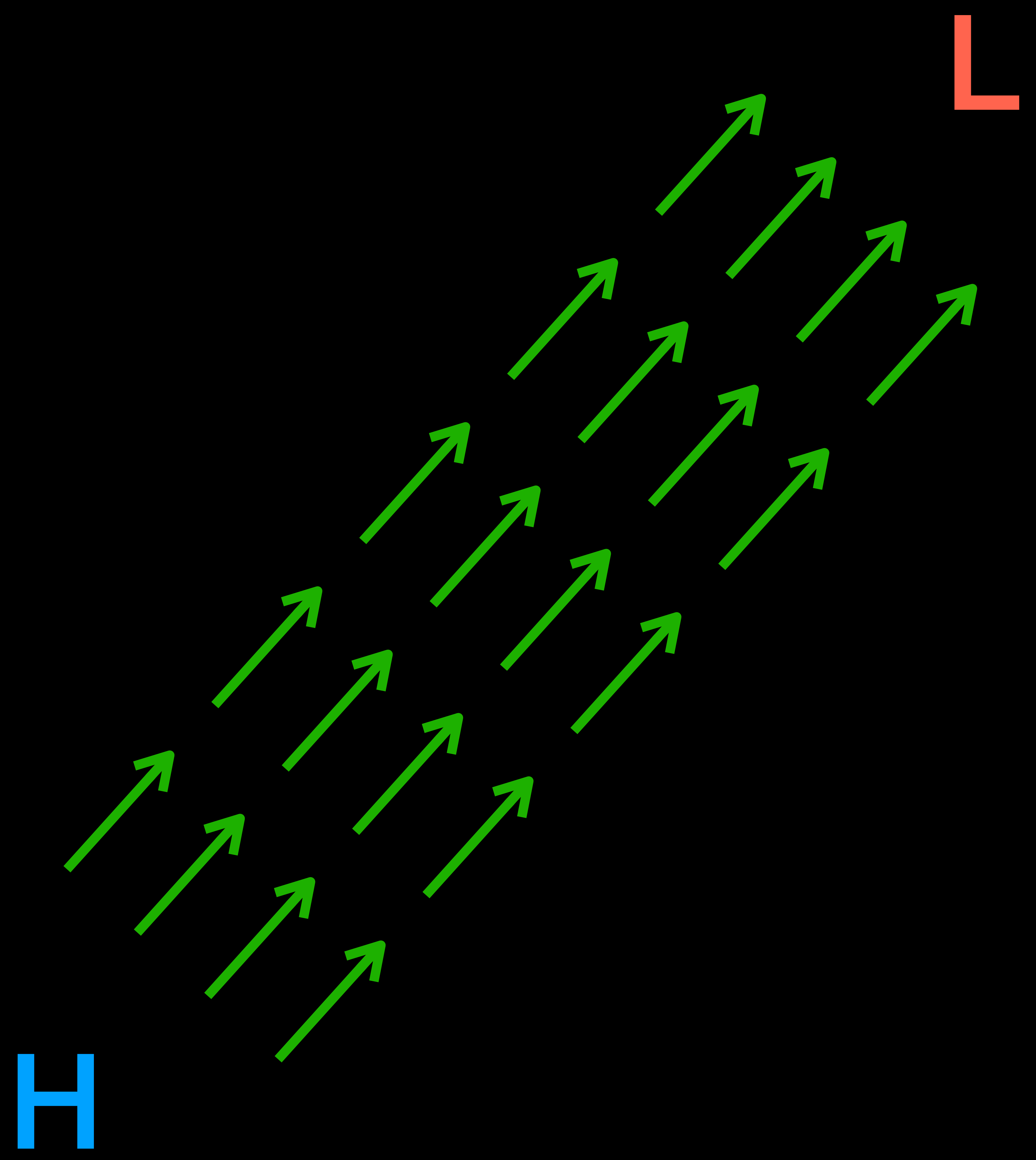
Set coordinates checkpoints

Affects all boids



Similarity





Modified Boids Algorithm

Collision Avoidance

Flock Centering

Velocity Matching

Modified Boids Algorithm

~~Collision Avoidance~~ *(Asm. 1)*

Flock Centering

Velocity Matching

Modified Boids Algorithm

~~Collision Avoidance~~ *(Asm. 1)*

~~Flock Centering~~ *(Asm. 5)*

Velocity Matching

Modified Boids Algorithm

~~Collision Avoidance~~ *(Asm. 1)*

~~Flock Centering~~ *(Asm. 5)*

Velocity Matching

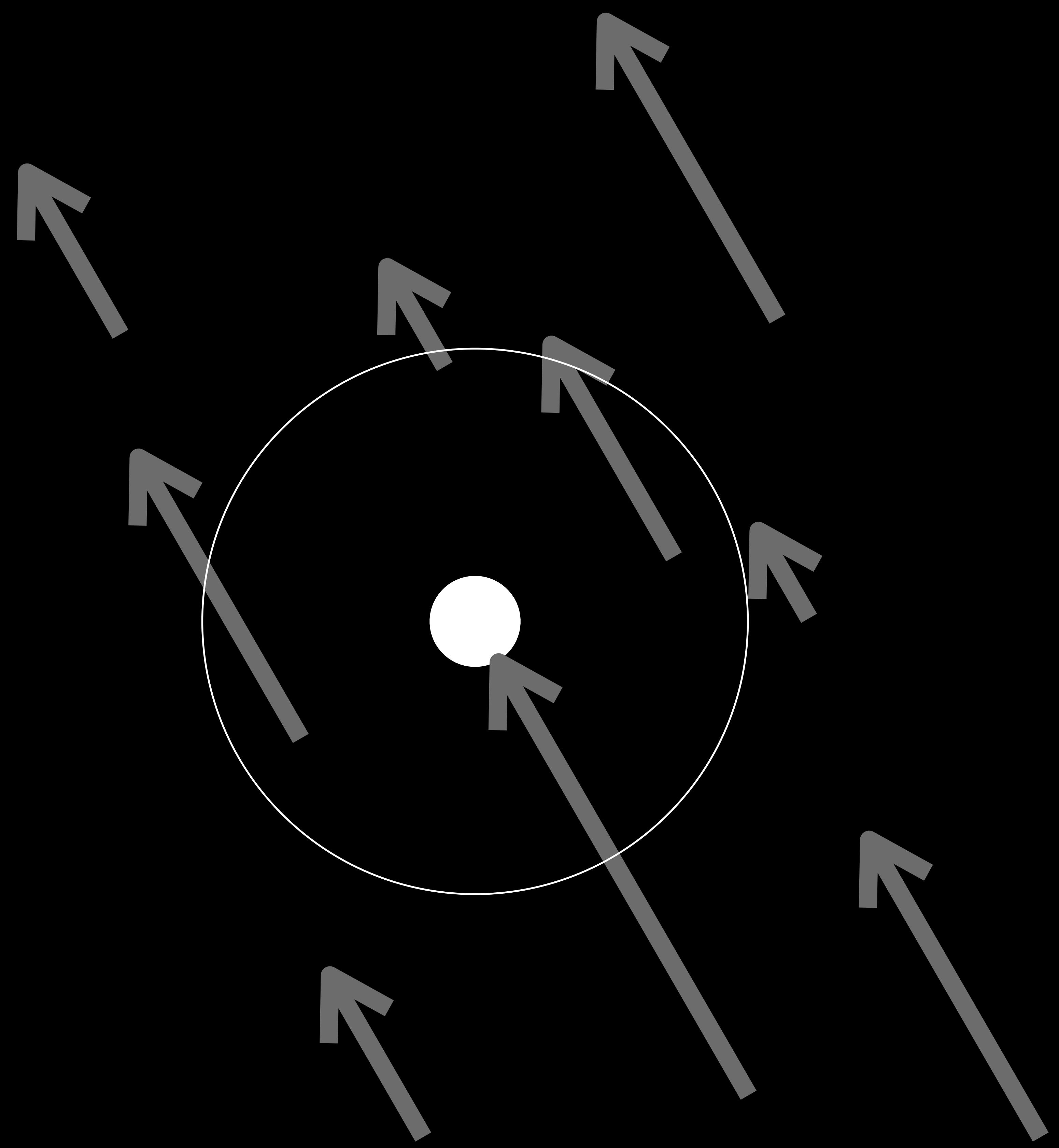
Modified Boids Algorithm

~~Collision Avoidance~~ *(Asm. 1)*

~~Flock Centering~~ *(Asm. 5)*

≈ Velocity Matching

\+ Define neighborhood radius



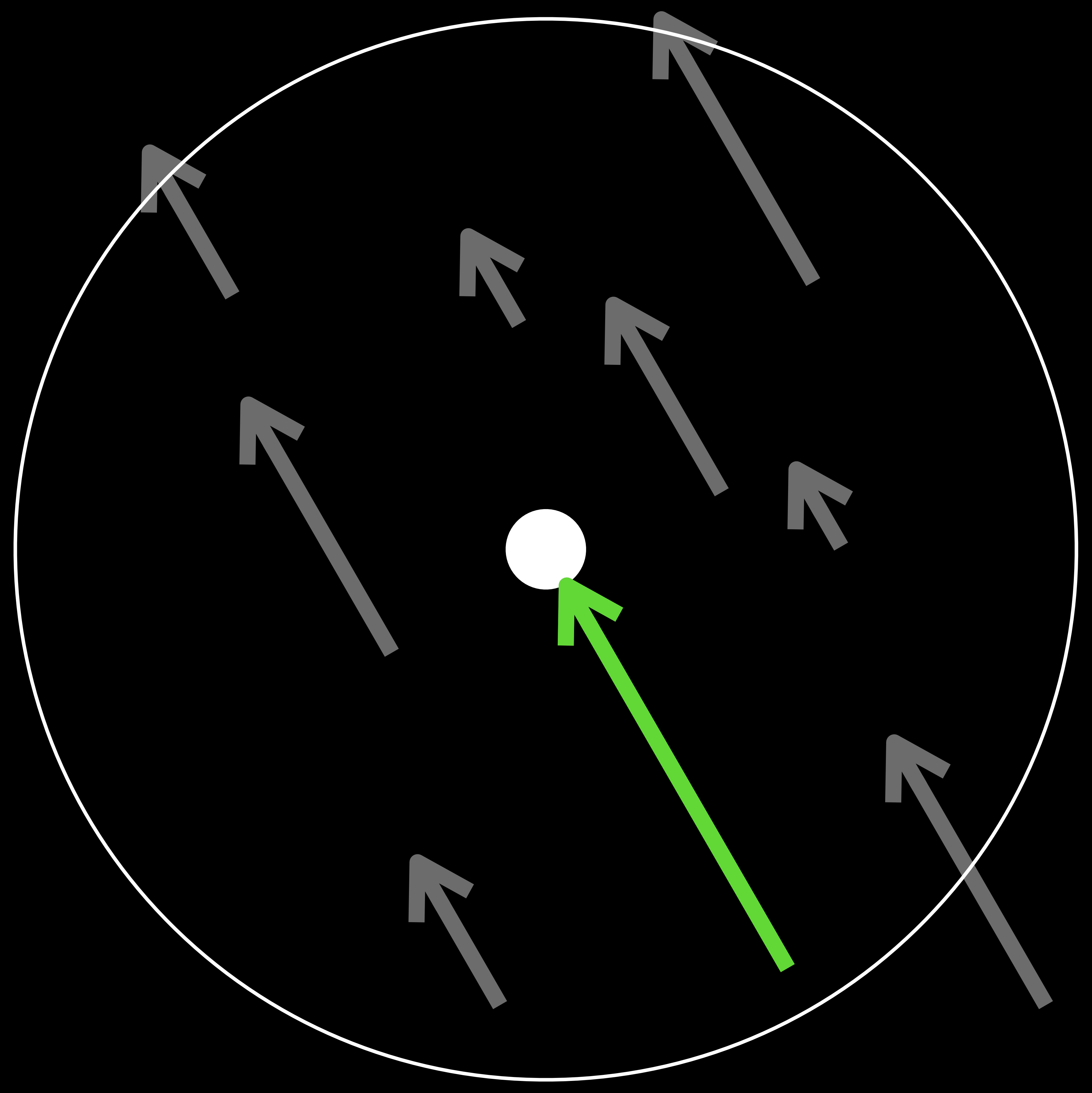
Modified Boids Algorithm

~~Collision Avoidance~~ *(Asm. 1)*

~~Flock Centering~~ *(Asm. 5)*

≈ Velocity Matching

\+ Define neighborhood radius



Modified Boids Algorithm

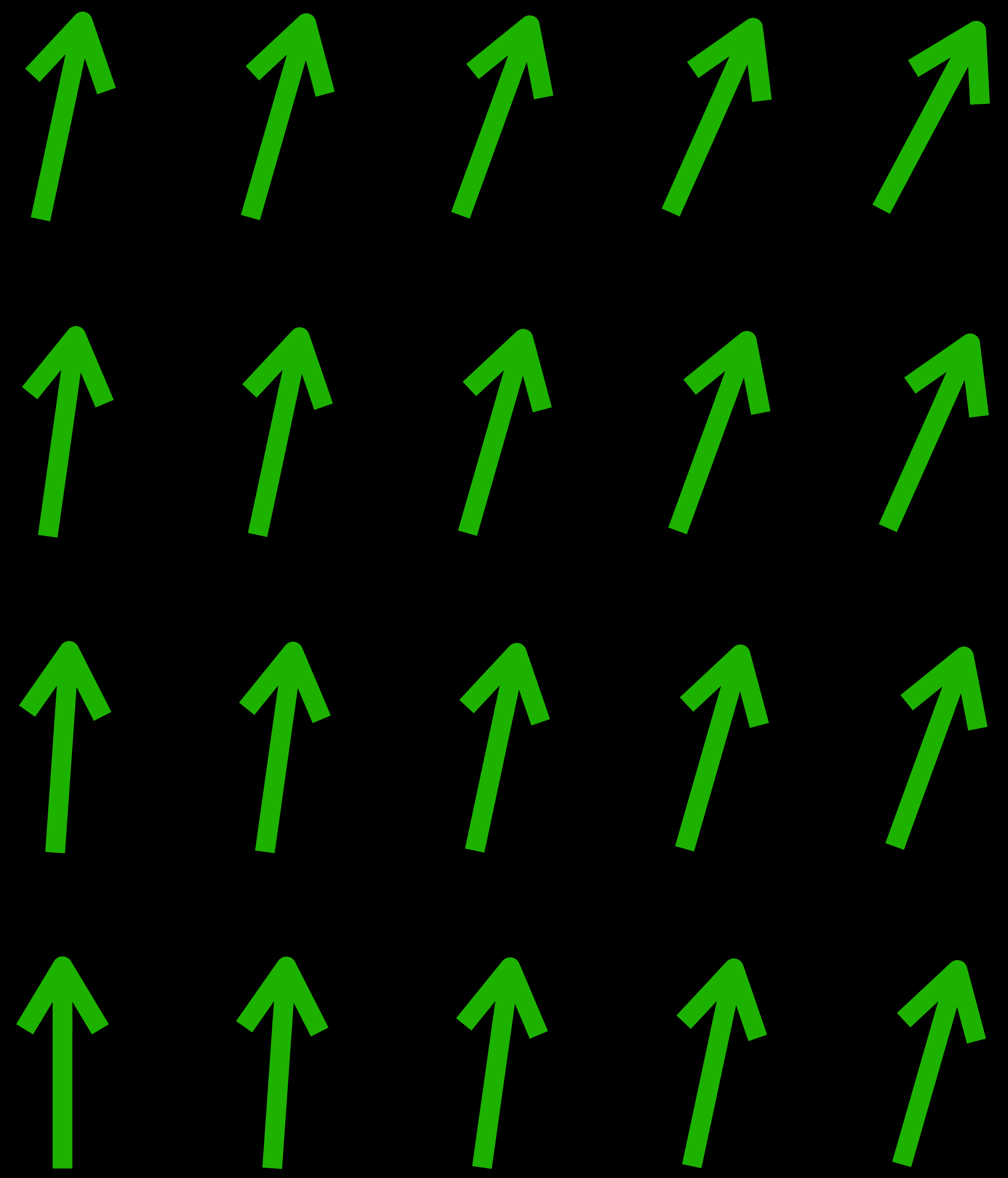
~~Collision Avoidance~~ *(Asm. 1)*

~~Flock Centering~~ *(Asm. 5)*

≈ Velocity Matching

+ Define neighborhood radius

+ Existence of a W_{map}



Modified Boids Algorithm

~~Collision Avoidance~~ *(Asm. 1)*

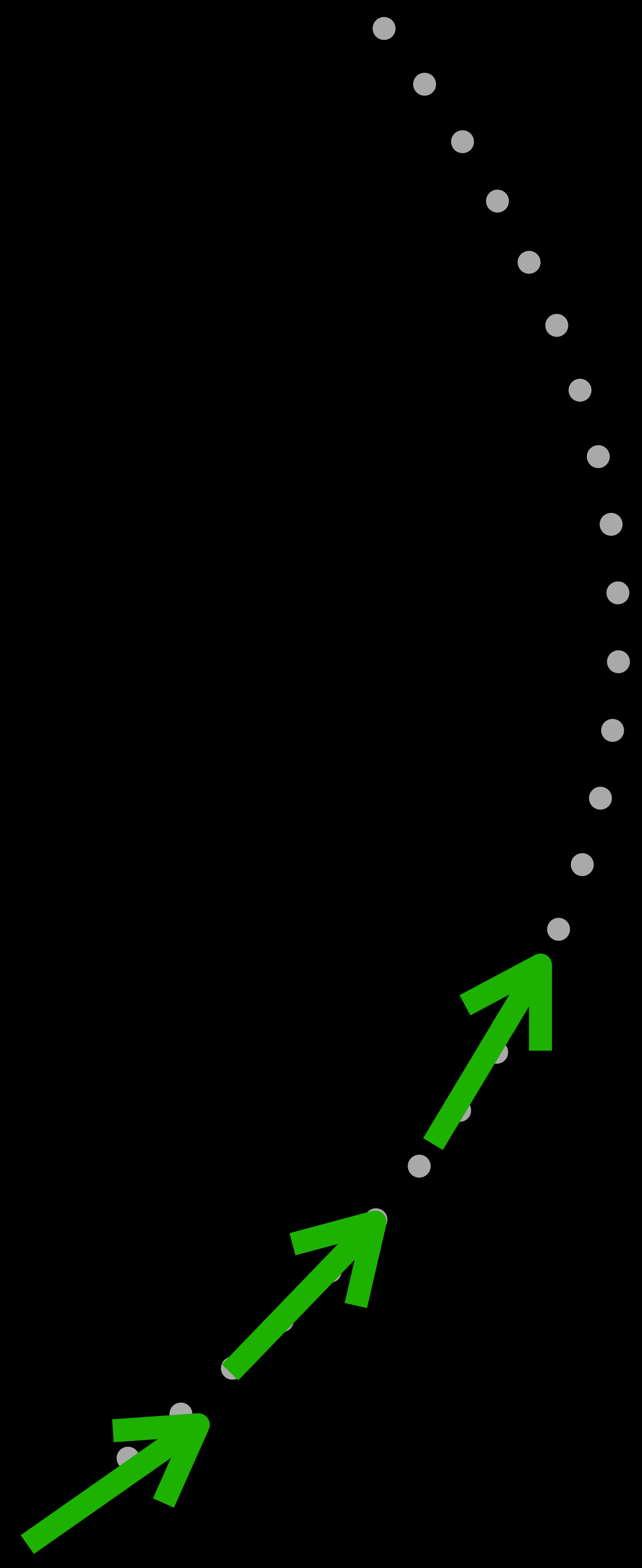
~~Flock Centering~~ *(Asm. 5)*

≈ Velocity Matching

+ Define neighborhood radius

+ Existence of a W_{map}

+ $\overrightarrow{\text{Centrifugal Force}}$



Modified Boids Algorithm

~~Collision Avoidance~~ *(Asm. 1)*

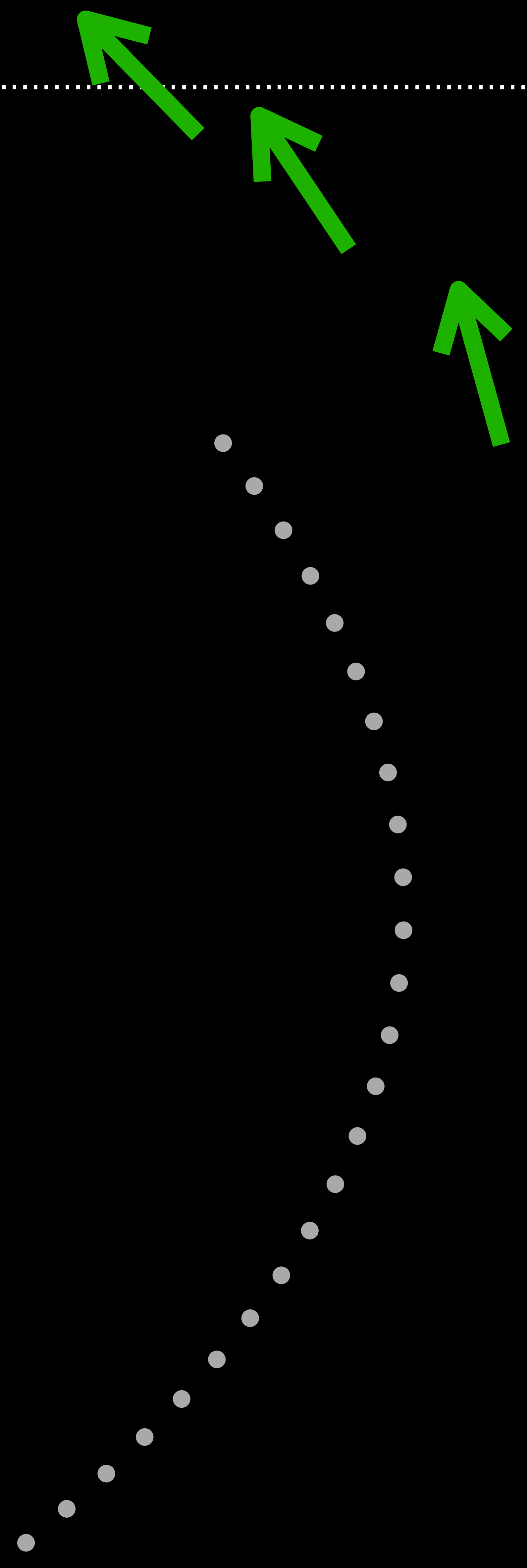
~~Flock Centering~~ *(Asm. 5)*

≈ Velocity Matching

\+ Define neighborhood radius

\+ Existence of a W_{map}

\+ $\overrightarrow{\text{Centrifugal Force}}$



Construct S' W_{map}.

Detect motion in each $i \in I$.

Save $\vec{v} \in i$ as *optical flows*.

Aggregate *flows* into a general W_{map}.

Insert C *particles* in S.

Filter C *pixels* and isolate them from S' background.

Initialize C *particles* in S.

Forecast W and C.

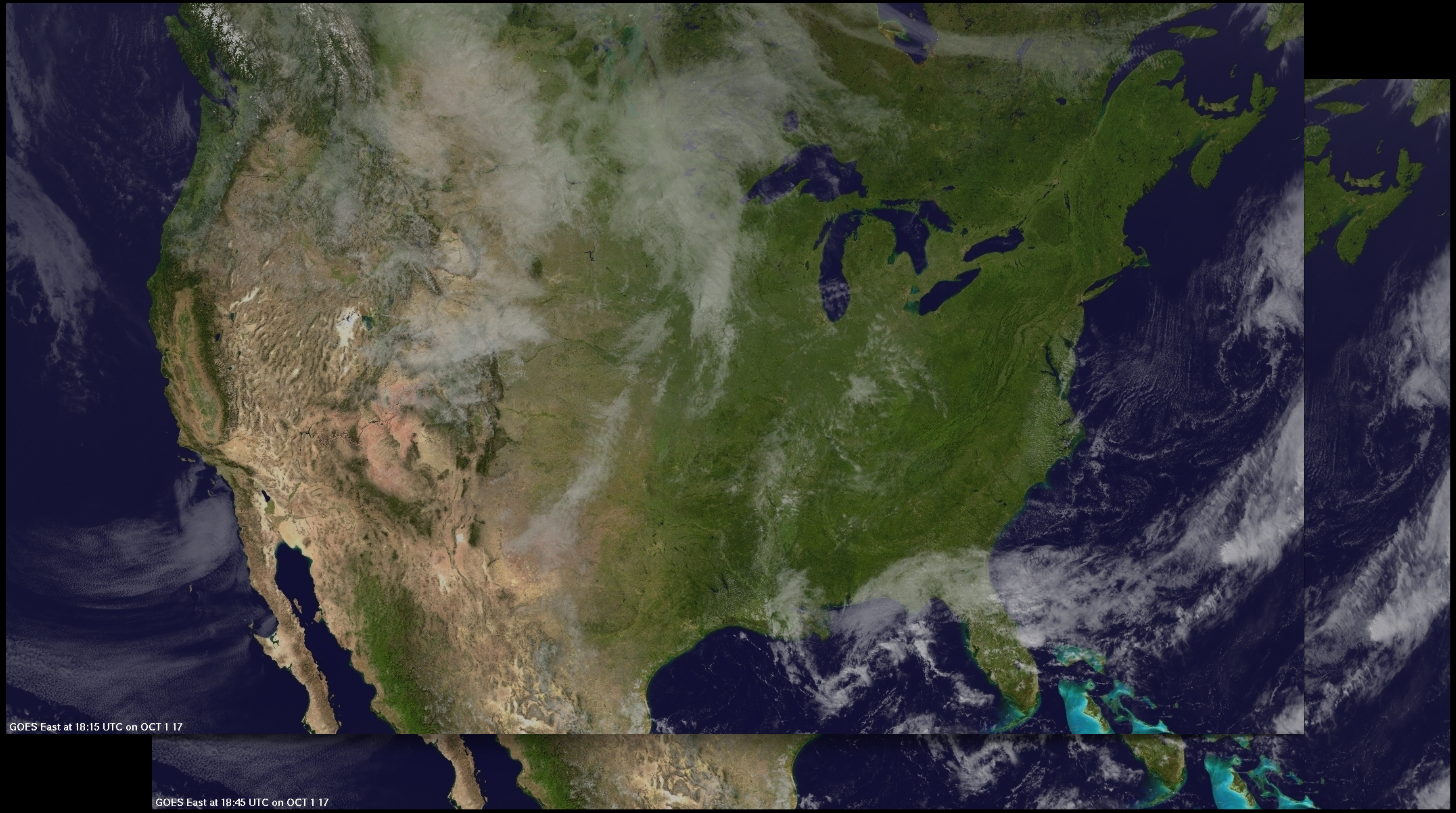
Update the W_{map} using the set of rules.

Update the C *particles'* positions according to the new W_{map}.

Workflow

(NOAA GOES Archive)

Workflow


GOES East at 18:15 UTC on OCT 1 17
GOES East at 18:45 UTC on OCT 1 17

Workflow

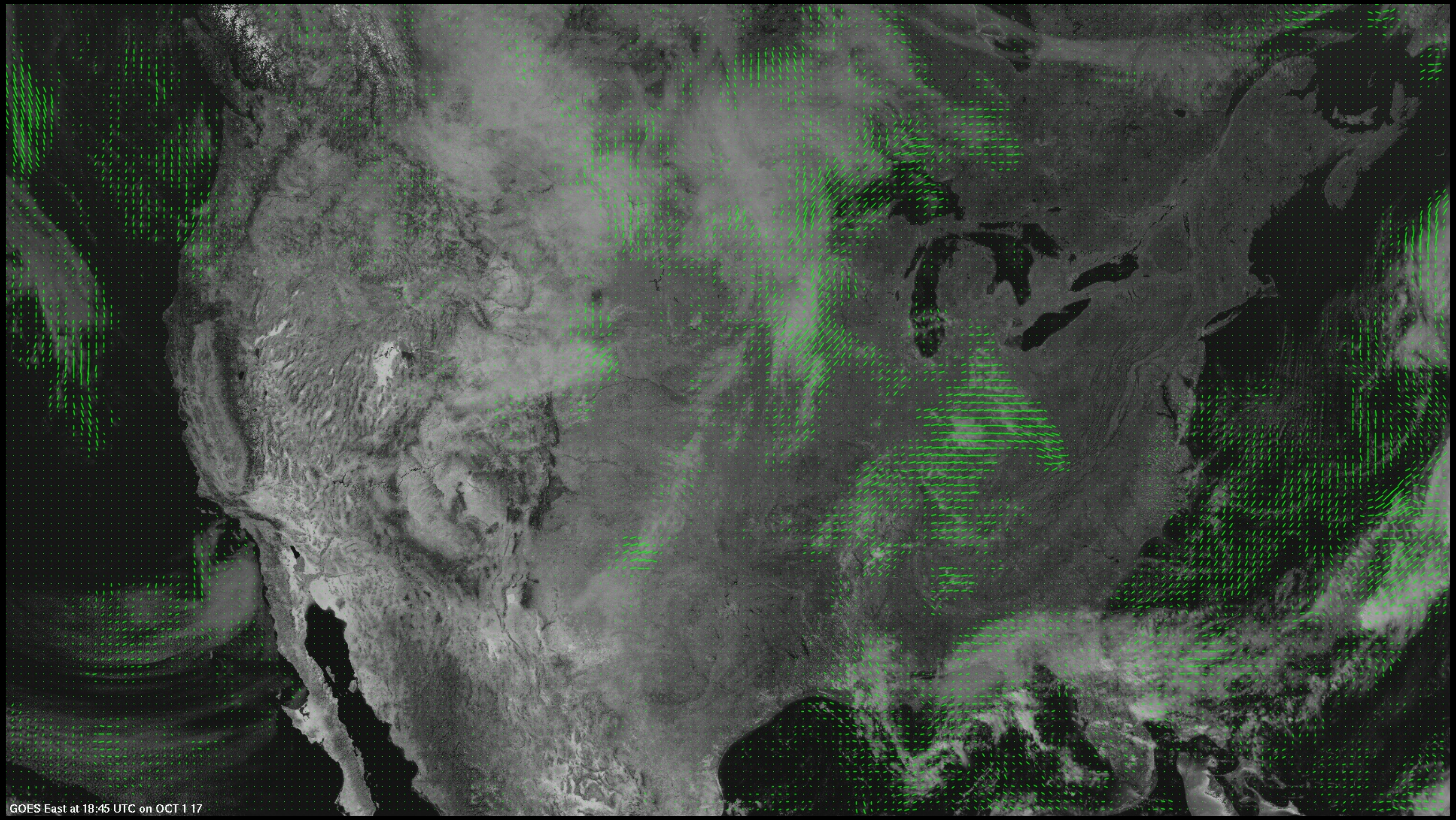

GOES East at 18:15 UTC on OCT 1 17
GOES East at 18:45 UTC on OCT 1 17
GOES East at 18:15 UTC on OCT 1 17
GOES East at 18:45 UTC on OCT 1 17

Workflow

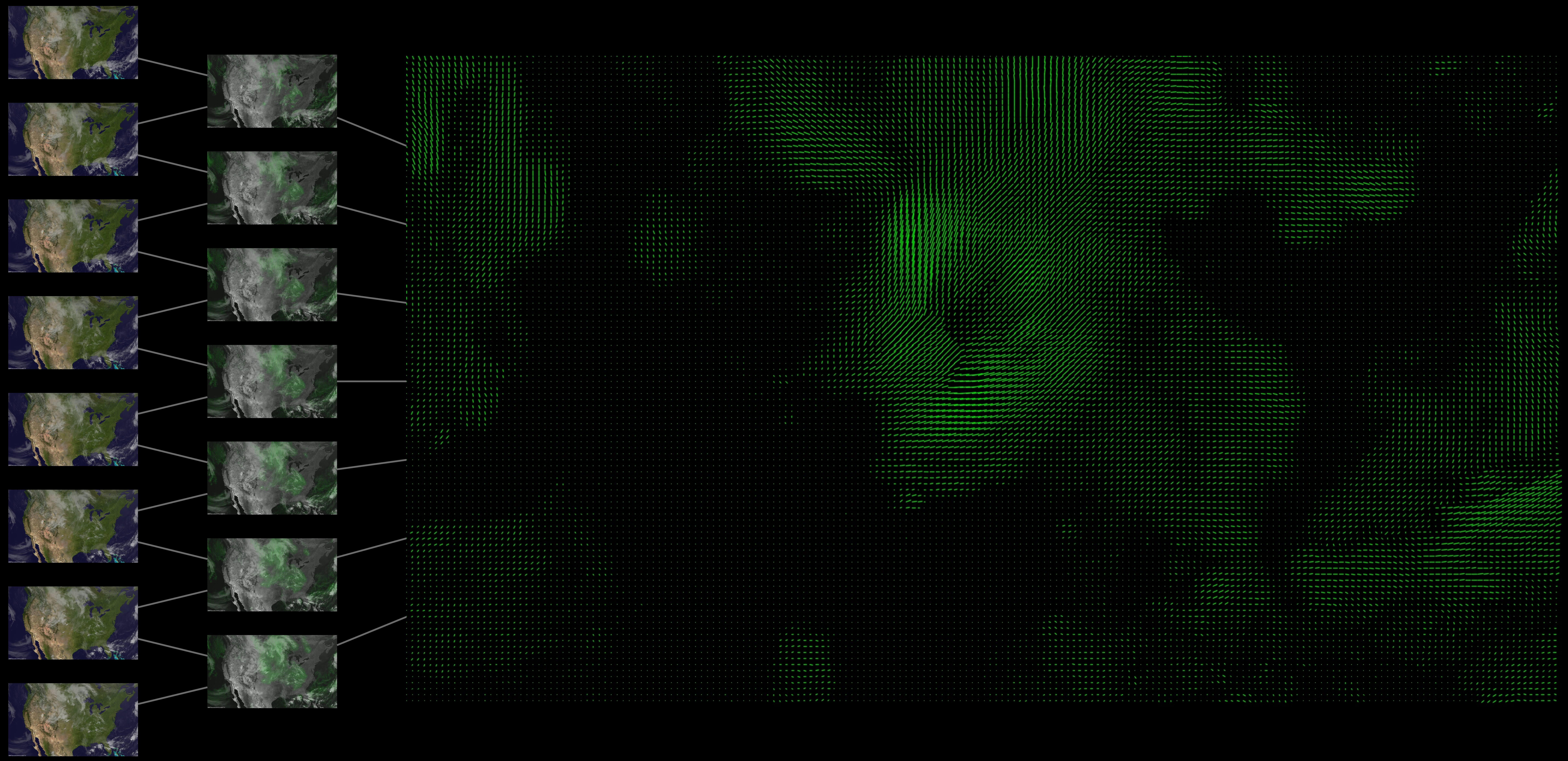
(Farnebäck, 2003)

Workflow

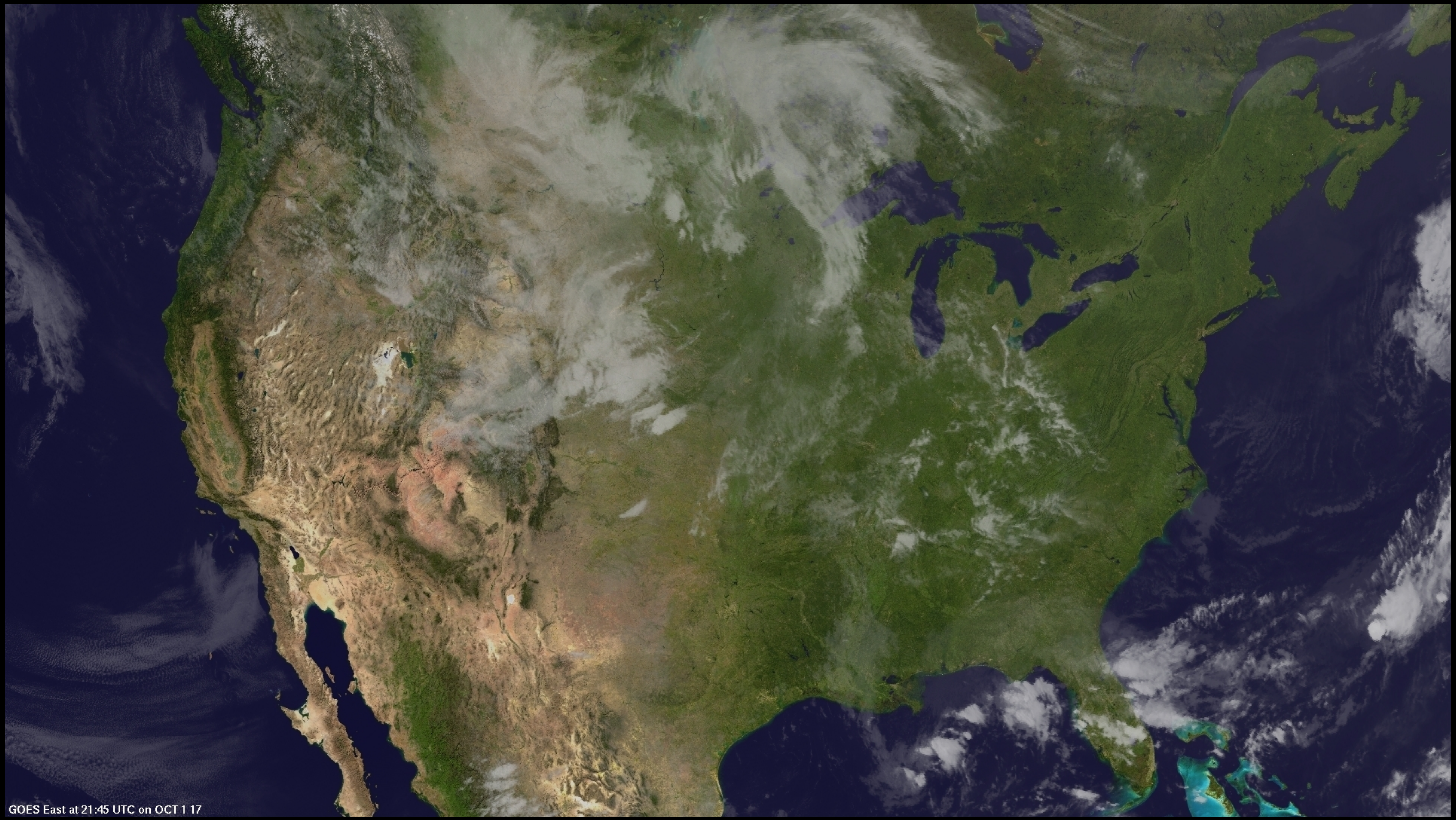
(Farnebäck, 2003)



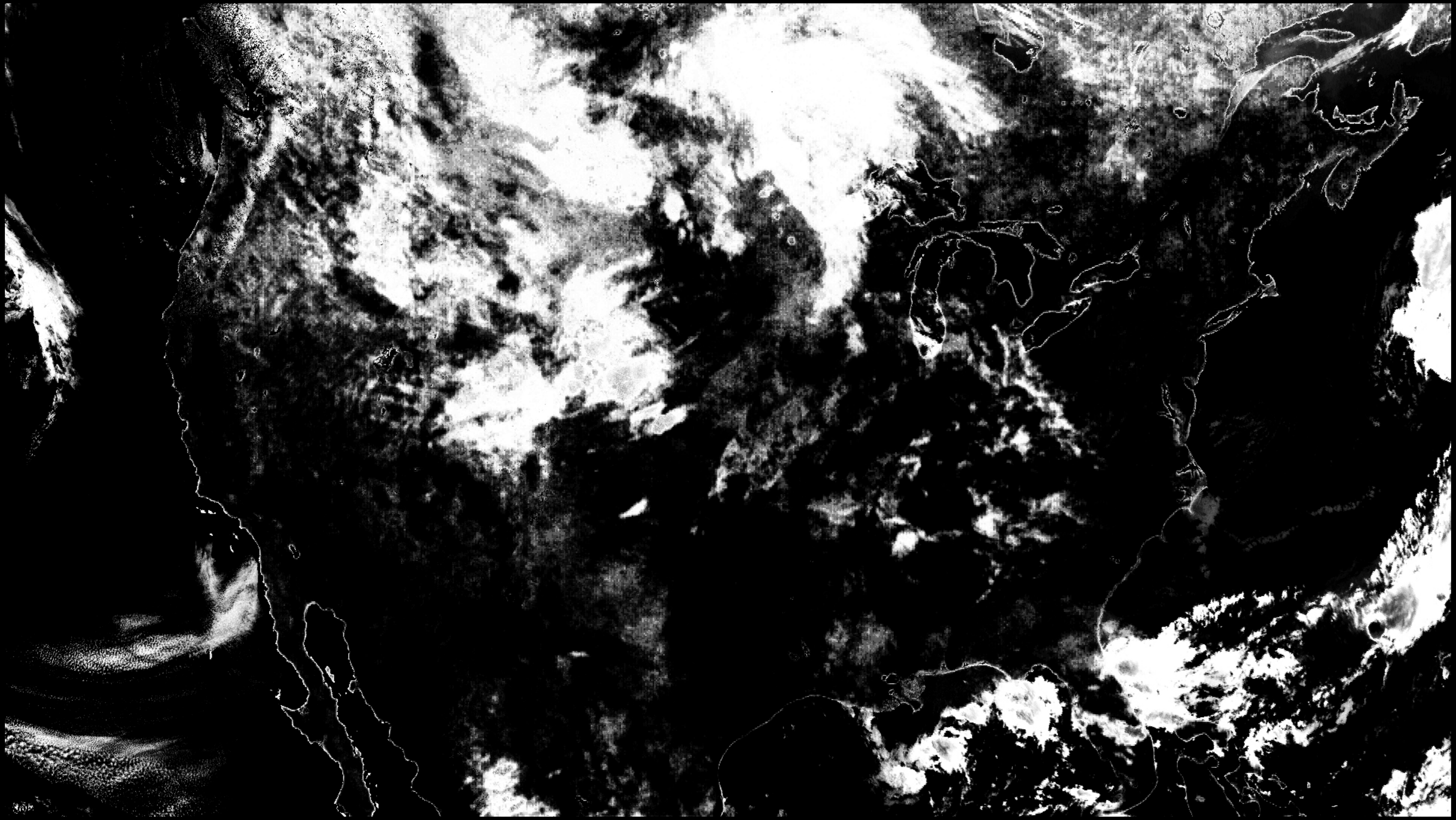
Workflow



Workflow



Workflow



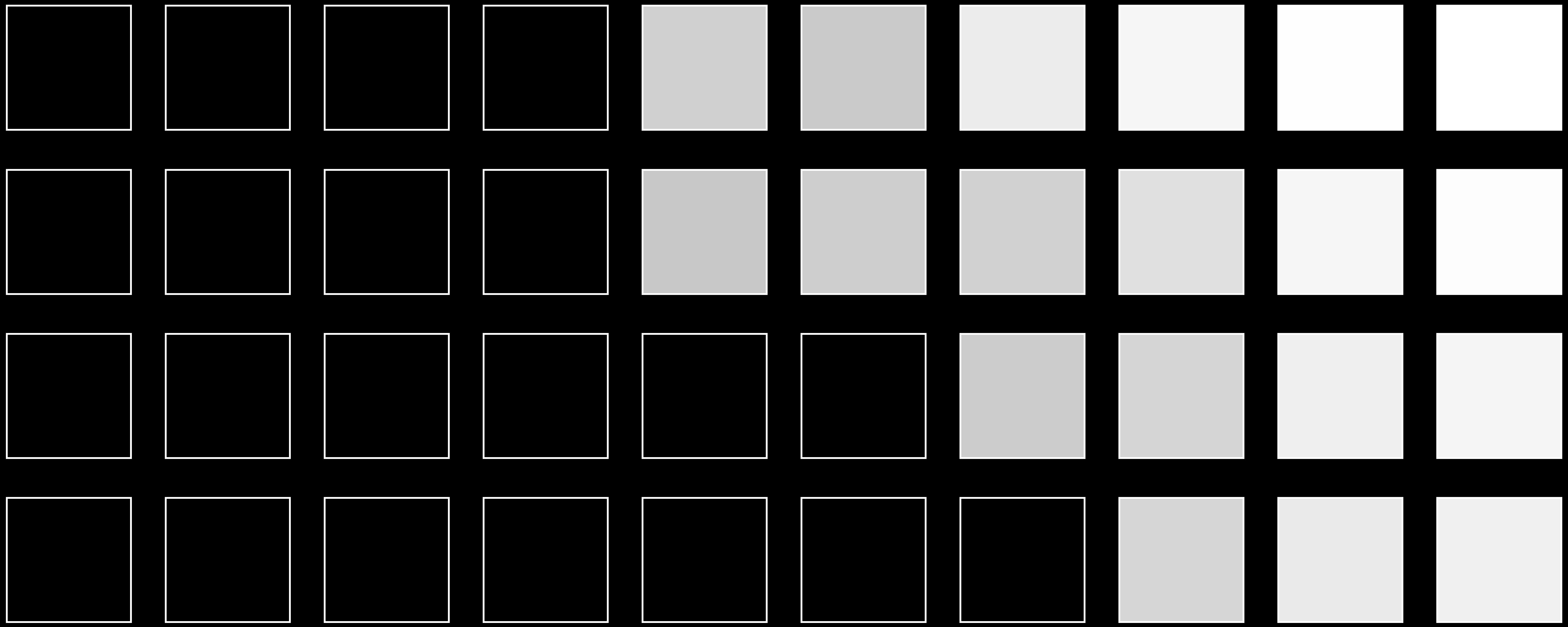
Saturation Filtering

67	64	53	39	17	15	14	12	7	0
76	67	55	48	20	18	17	15	12	7
90	78	76	54	26	23	19	16	14	12
88	90	48	43	37	30	24	16	14	14

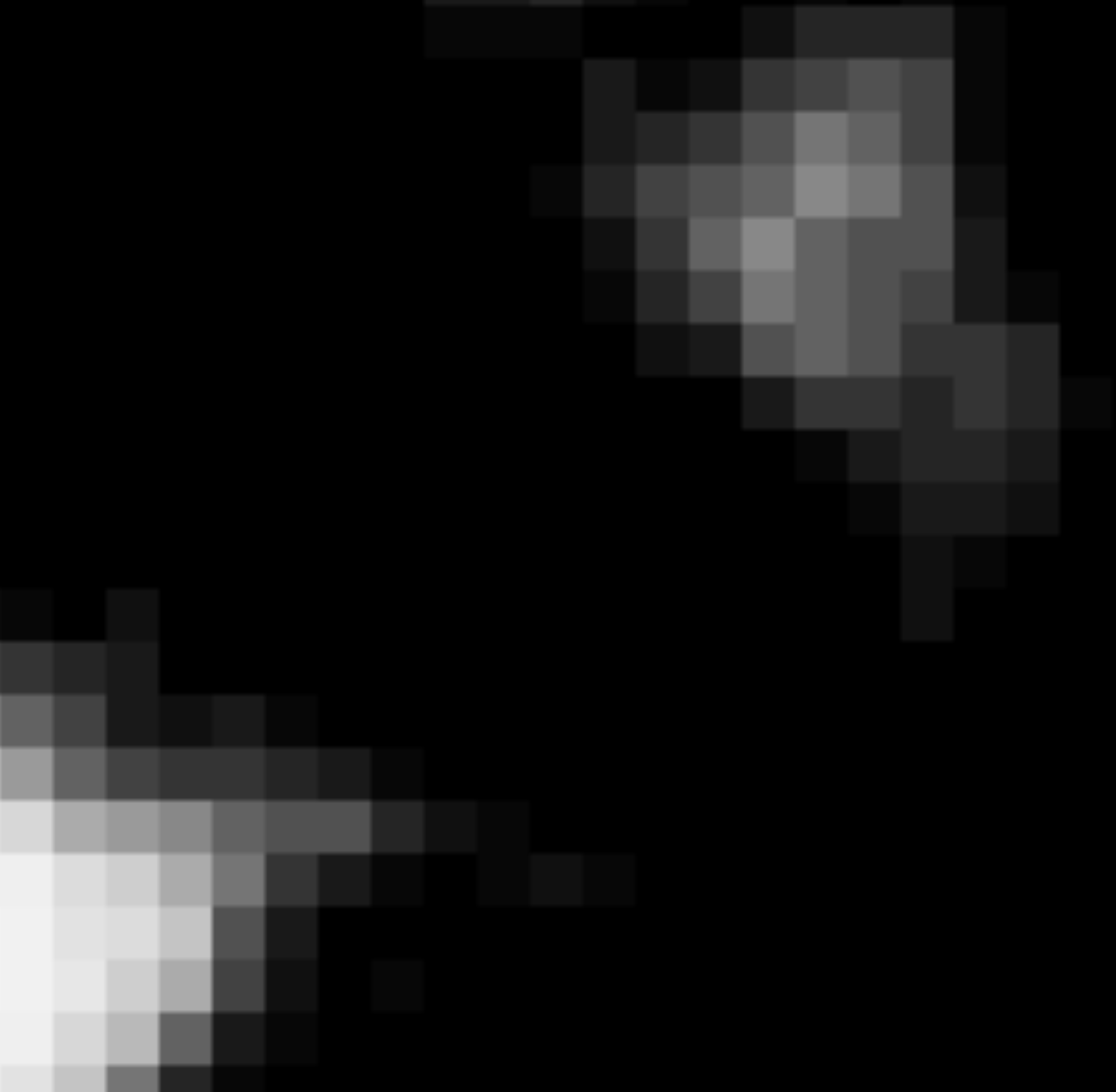
Saturation Filtering

67	64	53	39	17	15	14	12	7	0
76	67	55	48	20	18	17	15	12	7
90	78	76	54	26	23	19	16	14	12
88	90	48	43	37	30	24	16	14	14

Saturation Filtering



Optical Flow Process



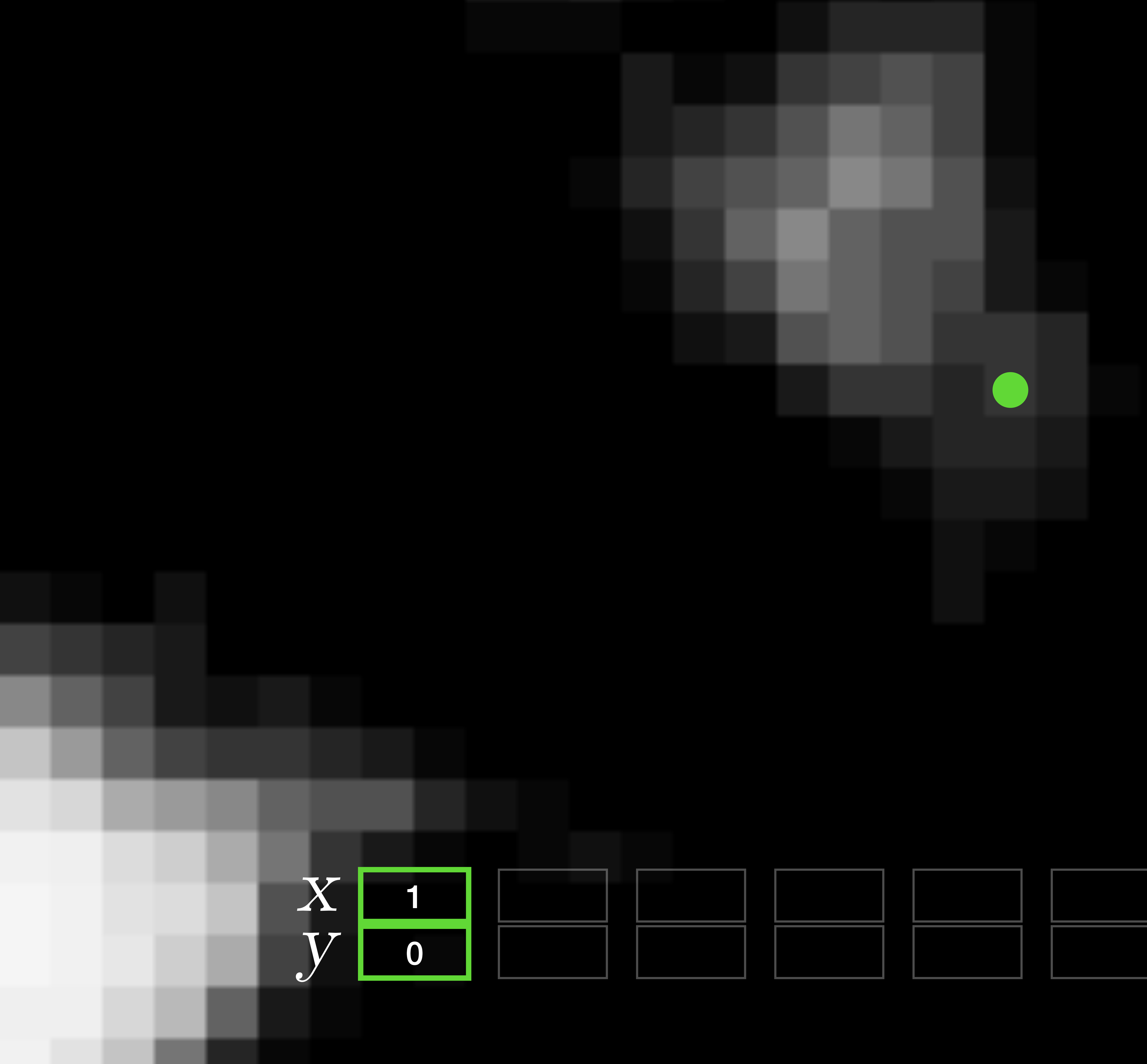
Optical Flow Process

Optical Flow Process

Optical Flow Process

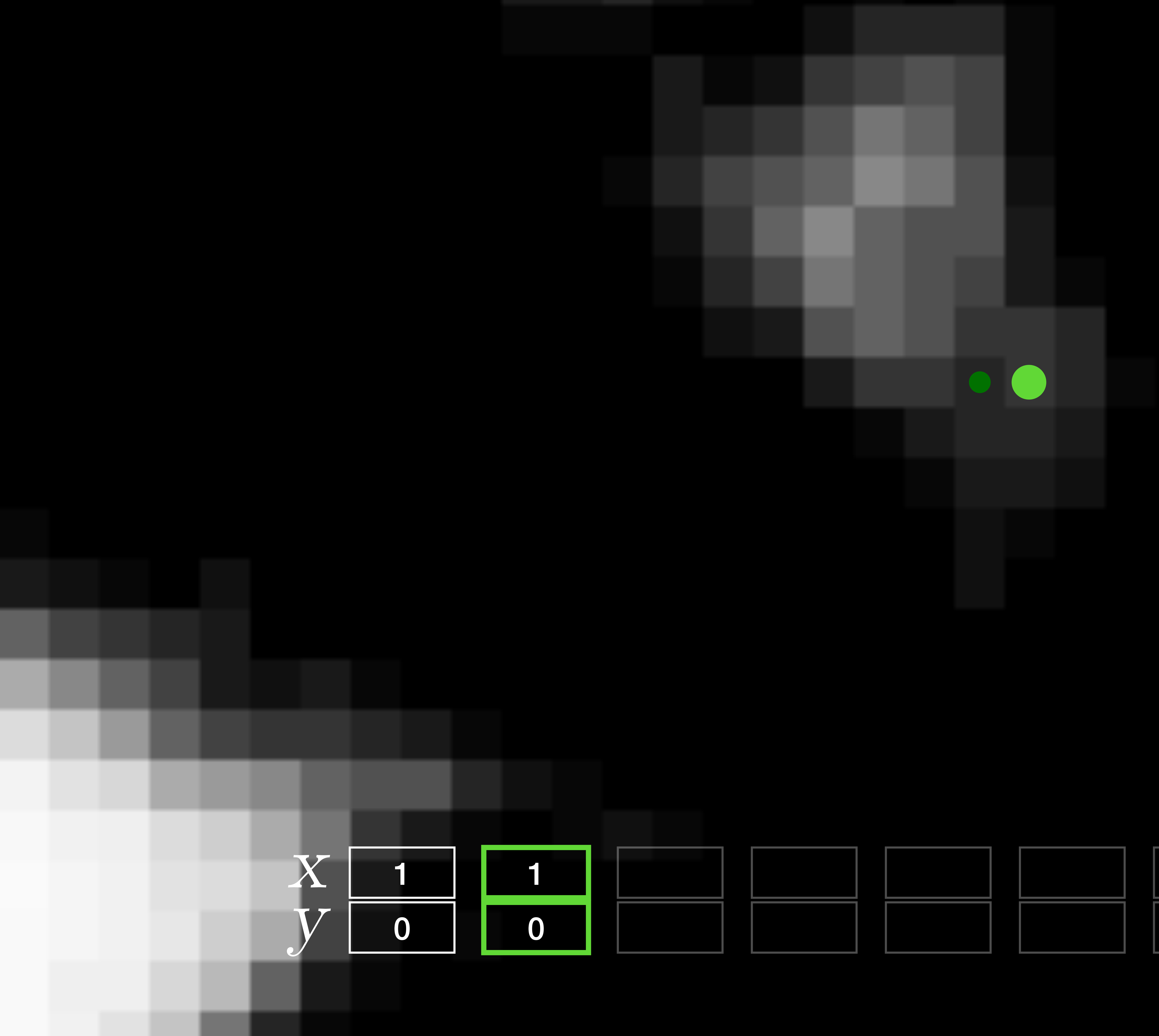
x	1									
y	0									

Optical Flow Process



x	1
y	0

Optical Flow Process



x	1	1								
y	0	0								

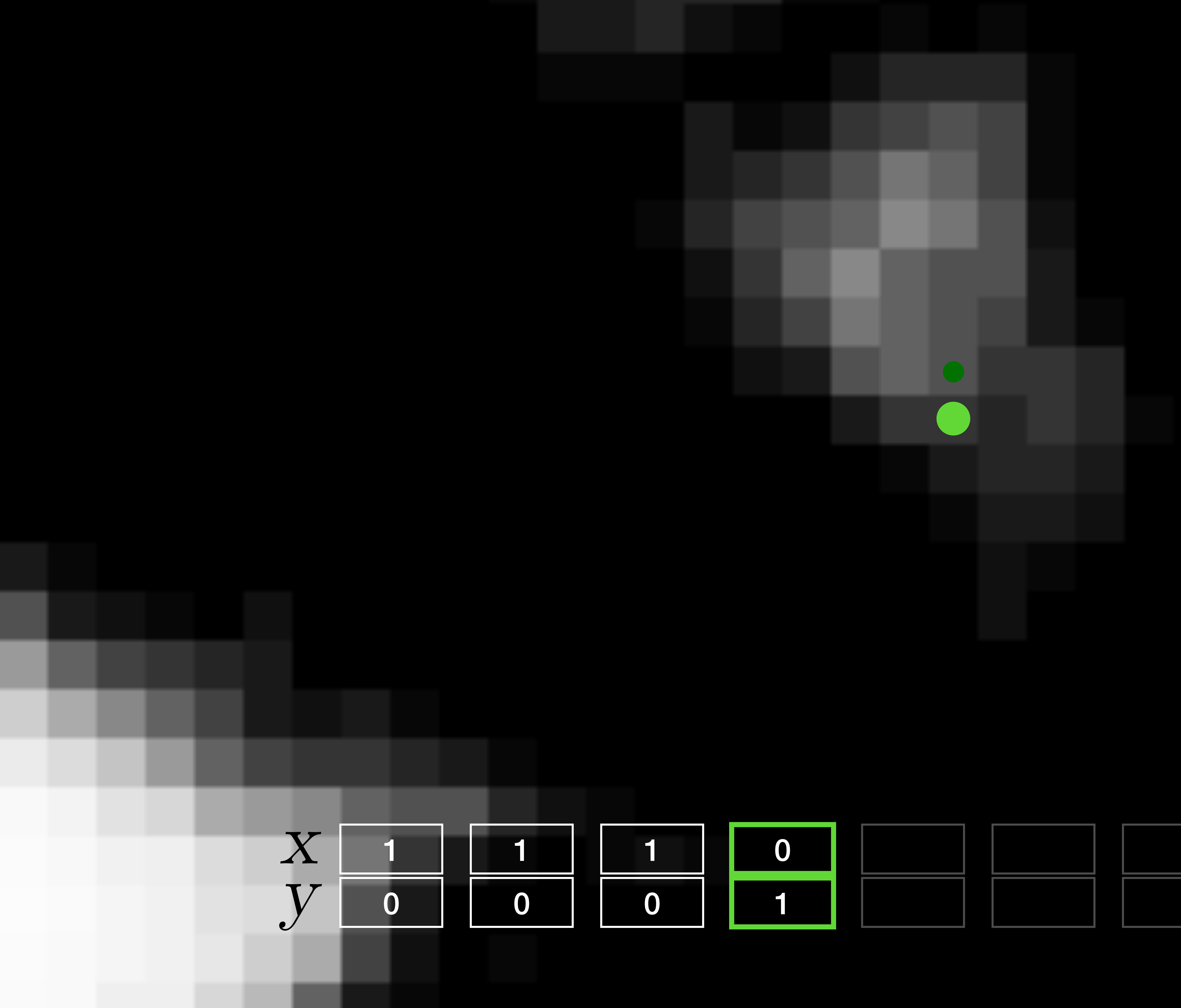
Optical Flow Process

Optical Flow Process

x	1	1	1							
y	0	0	0							

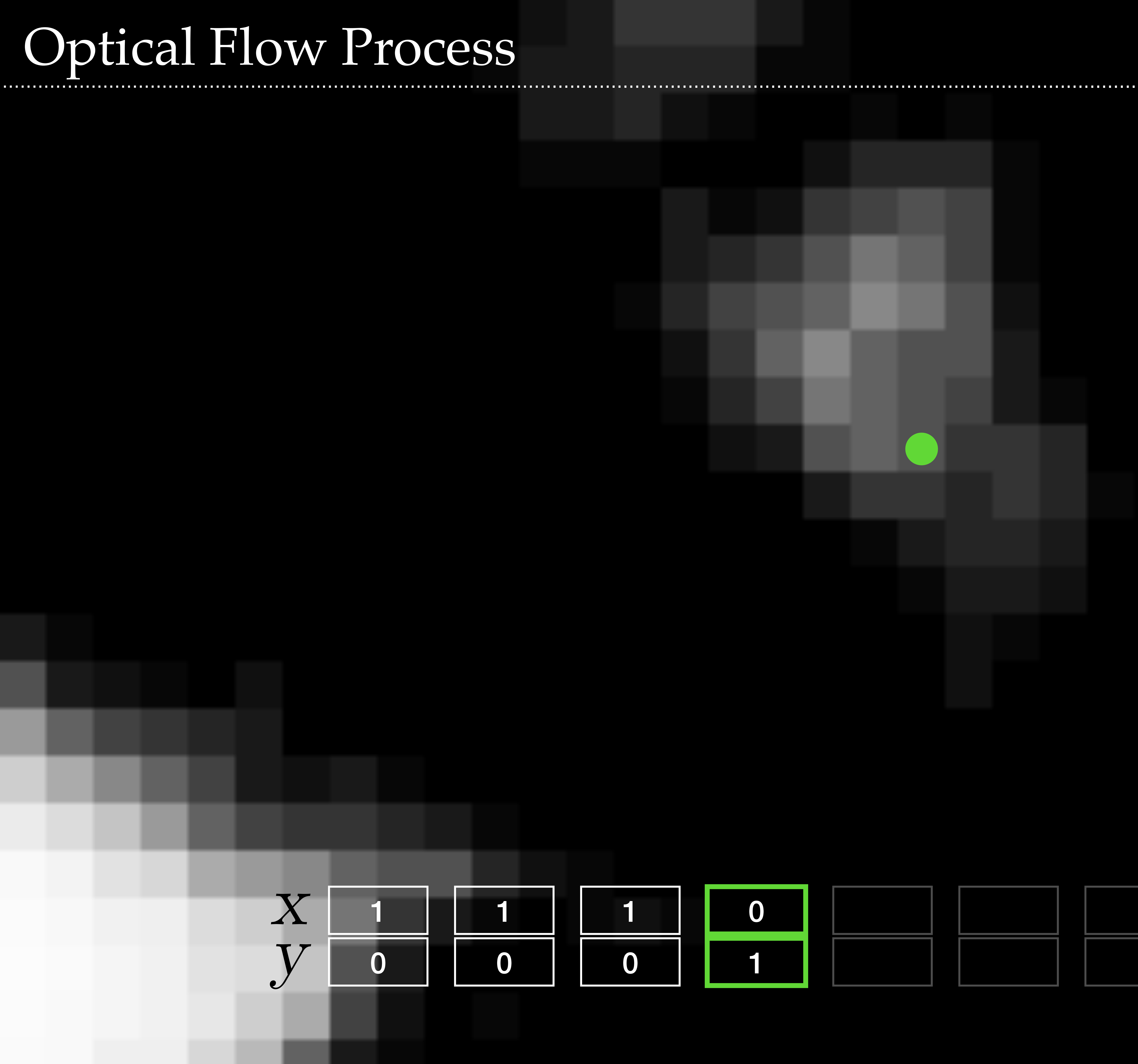
Optical Flow Process

Optical Flow Process



x	1	1	1	0						
y	0	0	0	1						

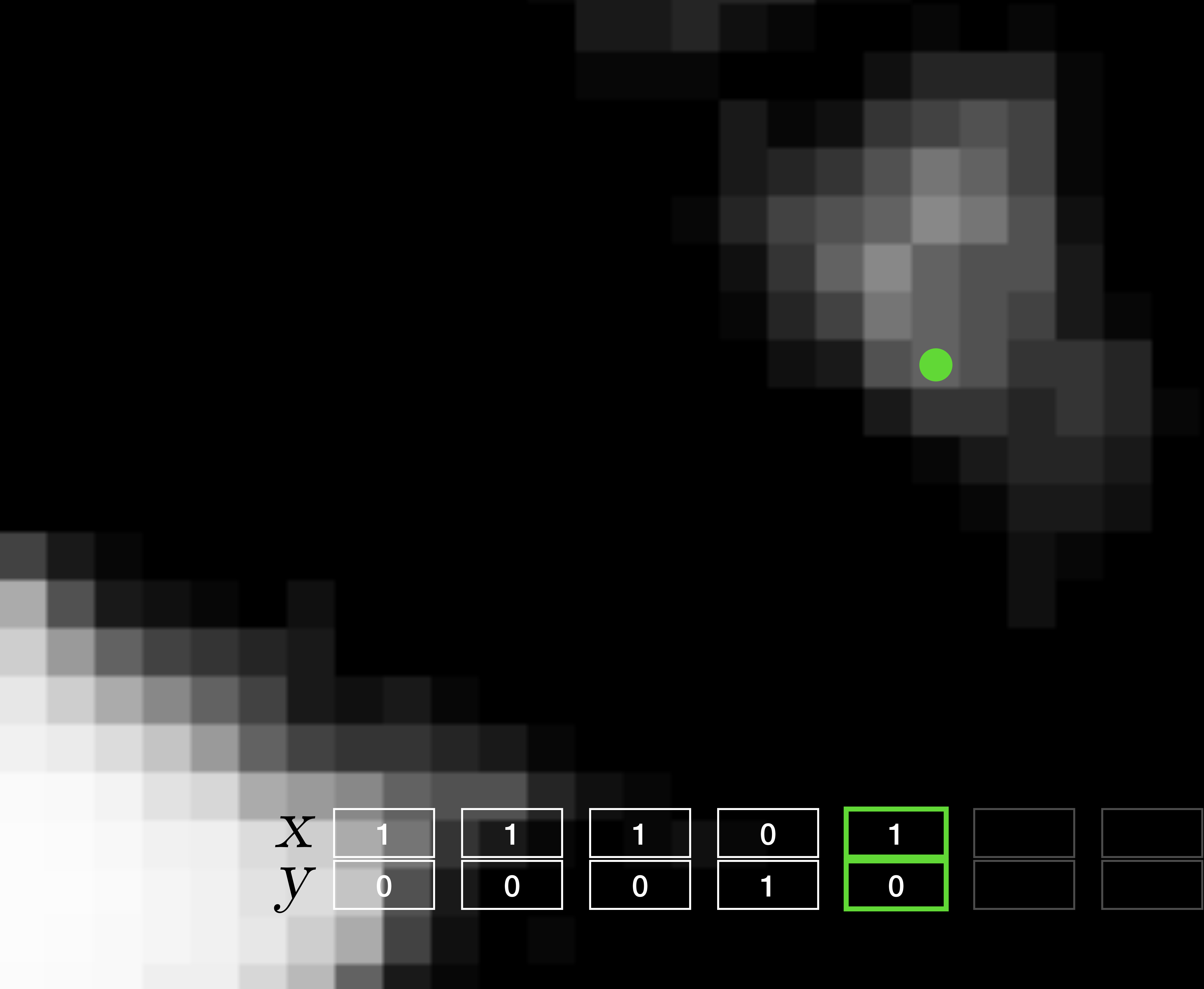
Optical Flow Process



x	1	1	1	0						
y	0	0	0	1						

Optical Flow Process

Optical Flow Process



x	1	1	1	0	1					
y	0	0	0	1	0					

Optical Flow Process

x	1	1	1	0	1	1				
y	0	0	0	1	0	0				

Optical Flow Process

x	1	1	1	0	1	1				
y	0	0	0	1	0	0				

Optical Flow Process

x	1	1	1	0	1	1	1			
y	0	0	0	1	0	0	0			

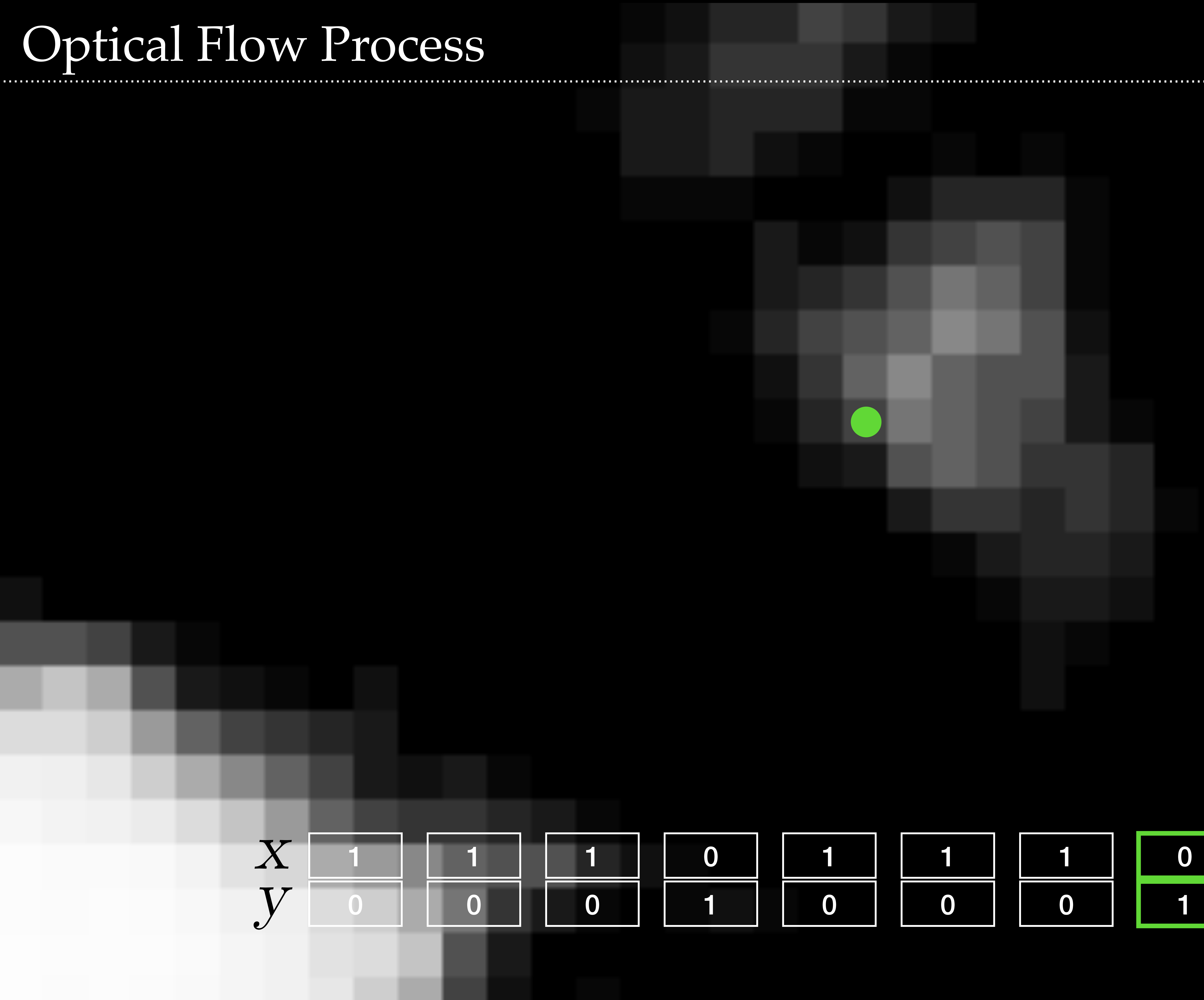
Optical Flow Process

x	1	1	1	0	1	1	1			
y	0	0	0	1	0	0	0			

Optical Flow Process

x	1	1	1	0	1	1	1	0		
y	0	0	0	1	0	0	0	1		

Optical Flow Process



x	1	1	1	0	1	1	1	0		
y	0	0	0	1	0	0	0	1		

Optical Flow Process

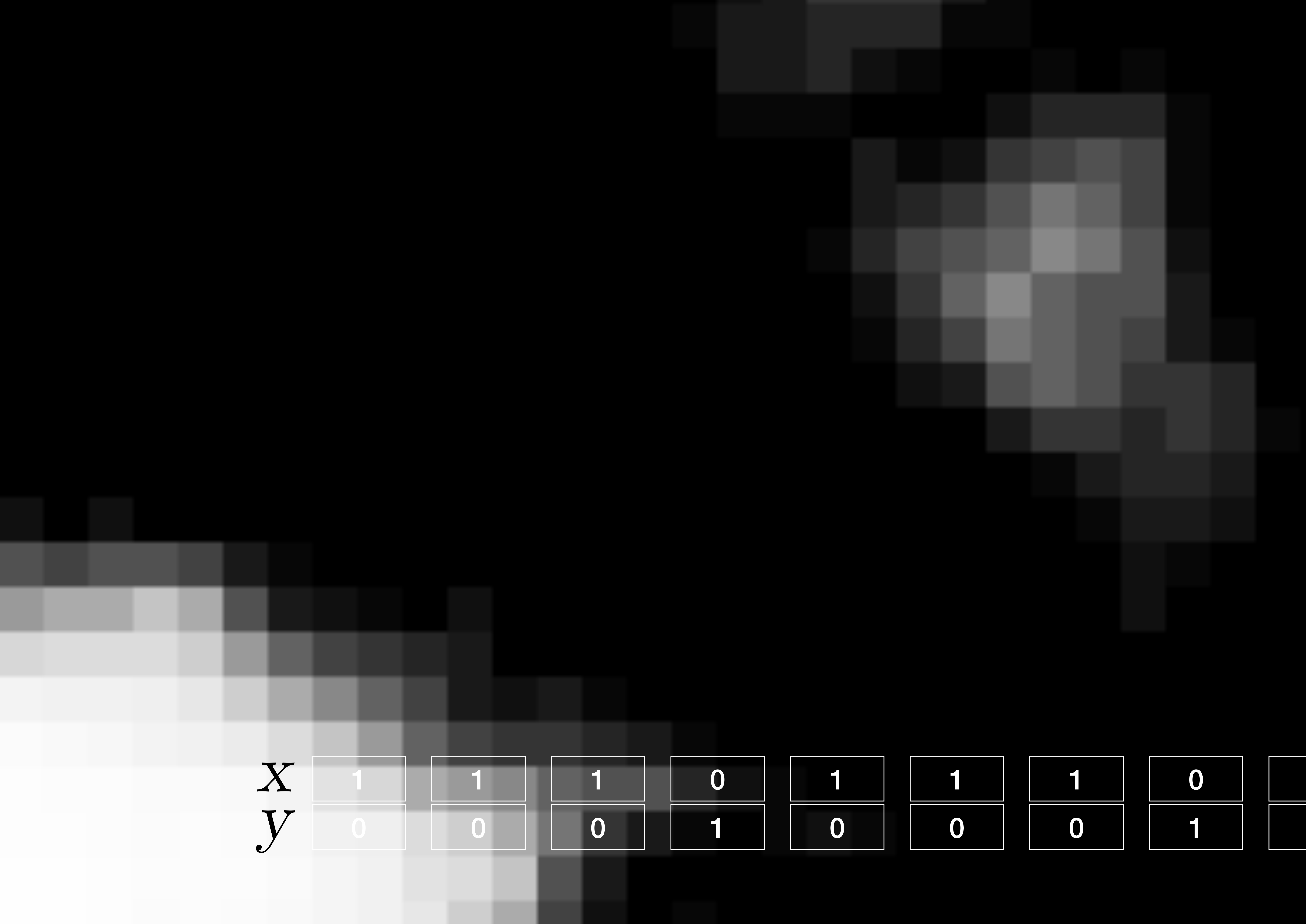
x	1	1	1	0	1	1	1	0	1	
y	0	0	0	1	0	0	0	1	0	

Optical Flow Process

x	1	1	1	0	1	1	1	0	1	
y	0	0	0	1	0	0	0	1	0	

Optical Flow Process

Optical Flow Process



x	1	1	1	0	1	1	1	0	1	1
y	0	0	0	1	0	0	0	1	0	0

Optical Flow Process

x	1	1	1	0	1	1	1	0	1	1
y	0	0	0	1	0	0	0	1	0	0

Pixel motion data

1	1	1	0	1	1	1	0	1	1
0	0	0	1	0	0	0	1	0	0
1 → 2	2 → 3	3 → 4	4 → 5	5 → 6	6 → 7	7 → 8	8 → 9	9 → 10	10 → 11

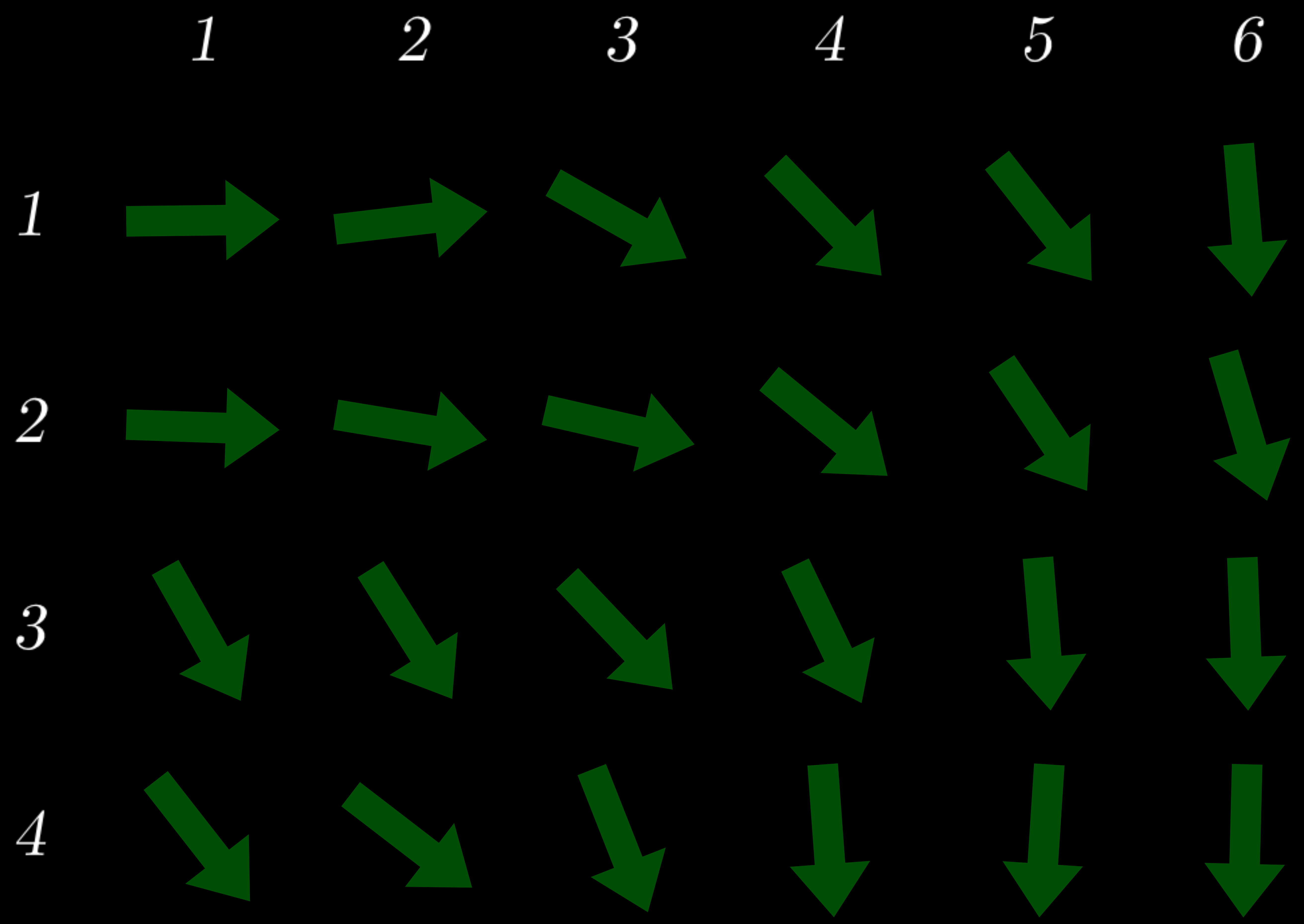
	1	2	3	4	5	6
1	1	1	1	1	1	1
	0	0	0	1	1	1
2	1	1	1	1	1	1
	0	0	0	0	1	1
3	1	1	1	0	0	0
	1	1	1	1	1	1
4	1	0	0	0	0	0
	1	1	1	1	1	1

1 → 2

	1	2	3	4	5	6
1	1.02246	1.02576	1.01616	1.04350	1.74782	1.06209
	-0.01143	-0.4596	0.3804	1.0055	1.64463	1.65841
2	1.0232	0.9910	0.9932	1.1044	1.5785	0.7234
	0.0032	0.0334	0.1890	0.3045	0.7001	1.0033
3	0.5693	0.7203	0.8993	0.4323	0.2003	0.0234
	0.7134	0.8024	0.9203	1.2304	1.4403	1.5774
4	0.5435	0.2232	0.1334	0.0021	-0.0043	-0.0123
	0.5463	0.5343	1.0102	1.0324	1.1084	1.2192

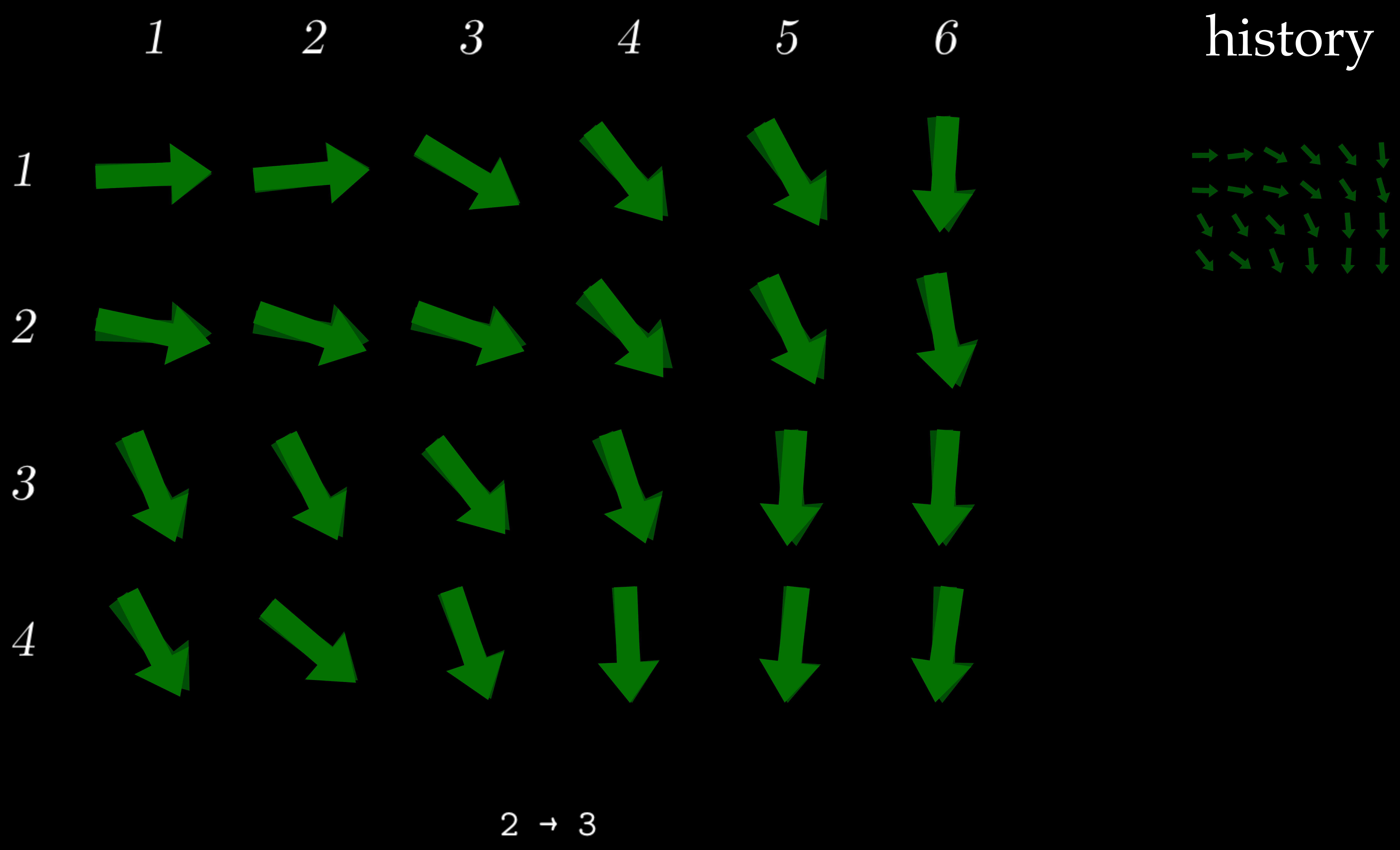
1 → 2

Optical Flow Process

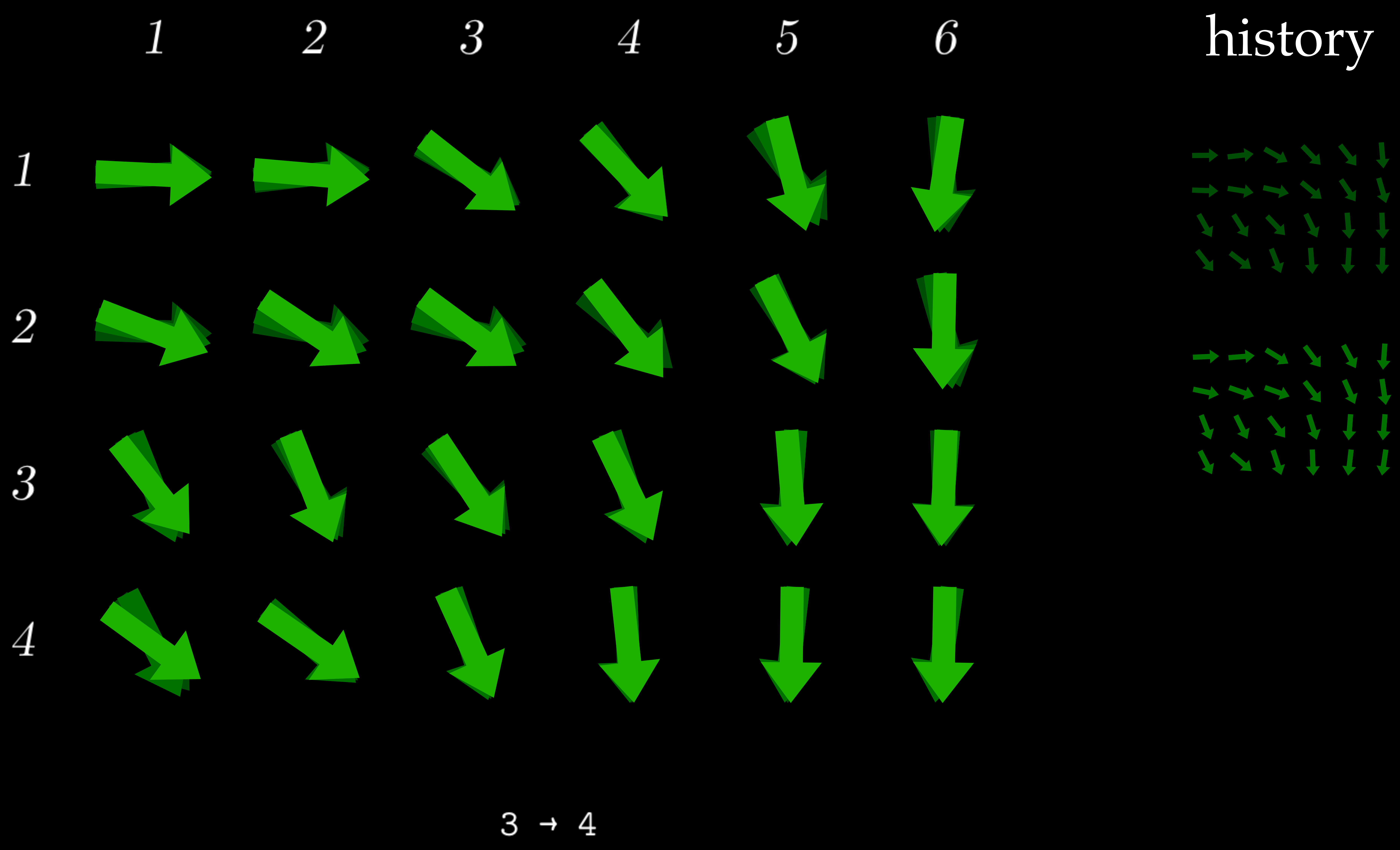


1 → 2

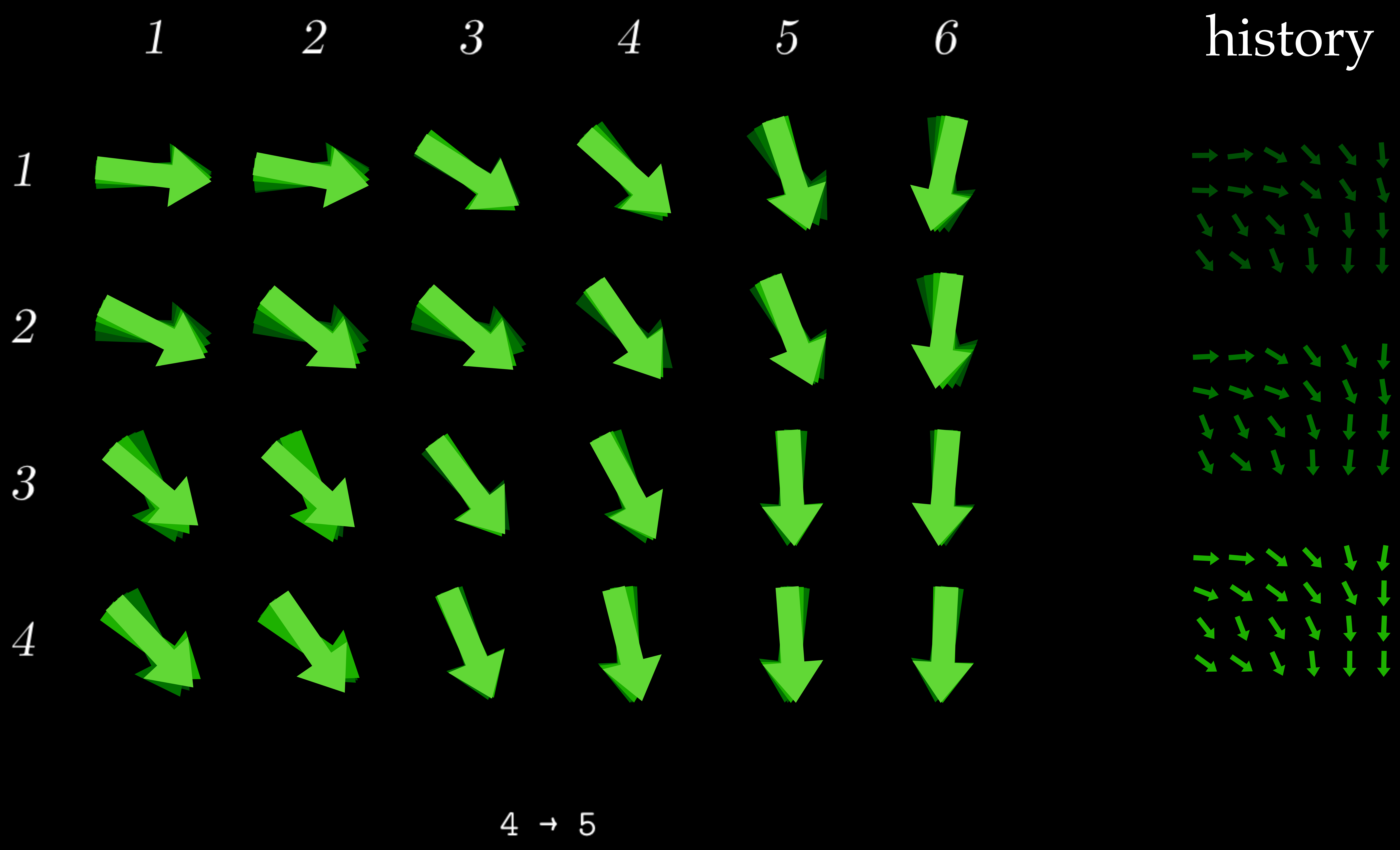
Optical Flow Process



Optical Flow Process



Optical Flow Process



```
#pragma omp parallel for shared(files)
    private(i) schedule(dynamic, 3)

for i ← 0 to range(files) do
    drawOpticalFlowMap()
    saveOverlays()
    averageFlows()
end for
```

Parallelisations • OpenMP

```
#pragma omp parallel for shared(files)
    private(i) schedule(dynamic, 3)

for i ← 0 to range(files) do
    #pragma omp sind                                        drawOpticalFlowMap()
    for y ← 0 to rows do
        for x ← 0 to cols do
            drawLine()
            drawCircle()
        end for
    end for
    saveOverlays()
    averageFlows()
end for
```

Parallelisations • OpenMP

```
#pragma omp parallel for shared(files)
    private(i) schedule(dynamic, 3)

for i ← 0 to range(files) do
    drawOpticalFlowMap()
    saveOverlays()
    averageFlows()
end for
```

Parallelisations • OpenMP

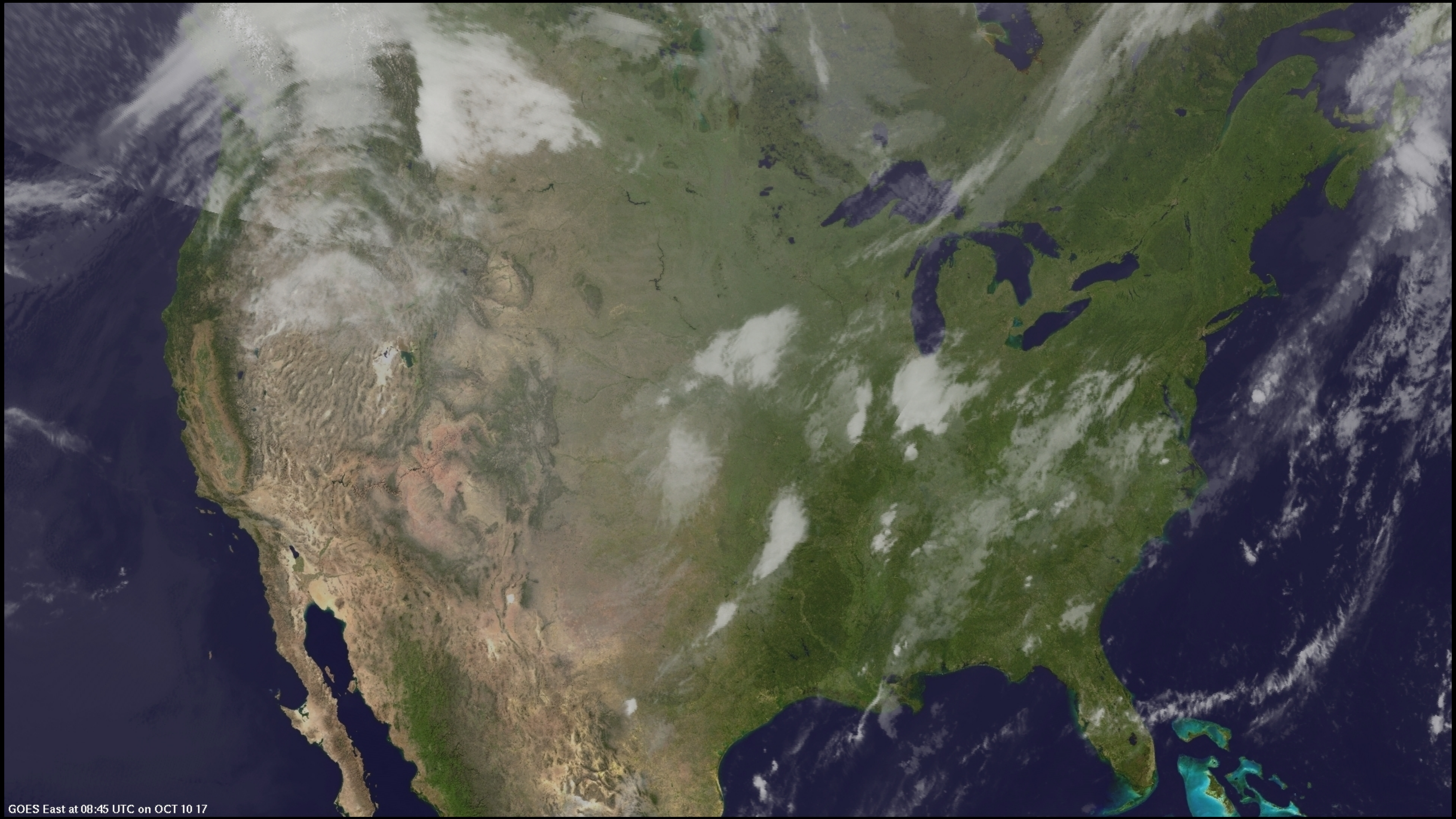
```
#pragma omp parallel for shared(files)
    private(i) schedule(dynamic, 3)

for i ← 0 to range(files) do
    drawOpticalFlowMap()
    saveOverlays()
    #pragma omp parallel for shared(files) private(index, row, col) averageFlows()
    for index ← 0 to numberOfFlows do
        readImage()
        for row ← 0 to imageRows do
            for col ← 0 to imageCols do
                addPoints()
            end for
        end for
    end for
    #pragma omp parallel for private(row, col)
    for row ← 0 to imageRows do
        for col ← 0 to imageCols do
            updatePoints()
        end for
    end for
end for
```

```
loc ← input[Tid]
v1, v2, v3 ← (0, 0)
for i ← -radius to radius do
    for j ← -radius to radius do
        v ← vel[(loc.y + i) * width + loc.x + j]
        if v ≠ (0, 0) then
            accumulateVelocity(v, v1, v2, v3)
        end if
    end for
end for
output[Tid] ← v1 + v2 + v3
```

Contents

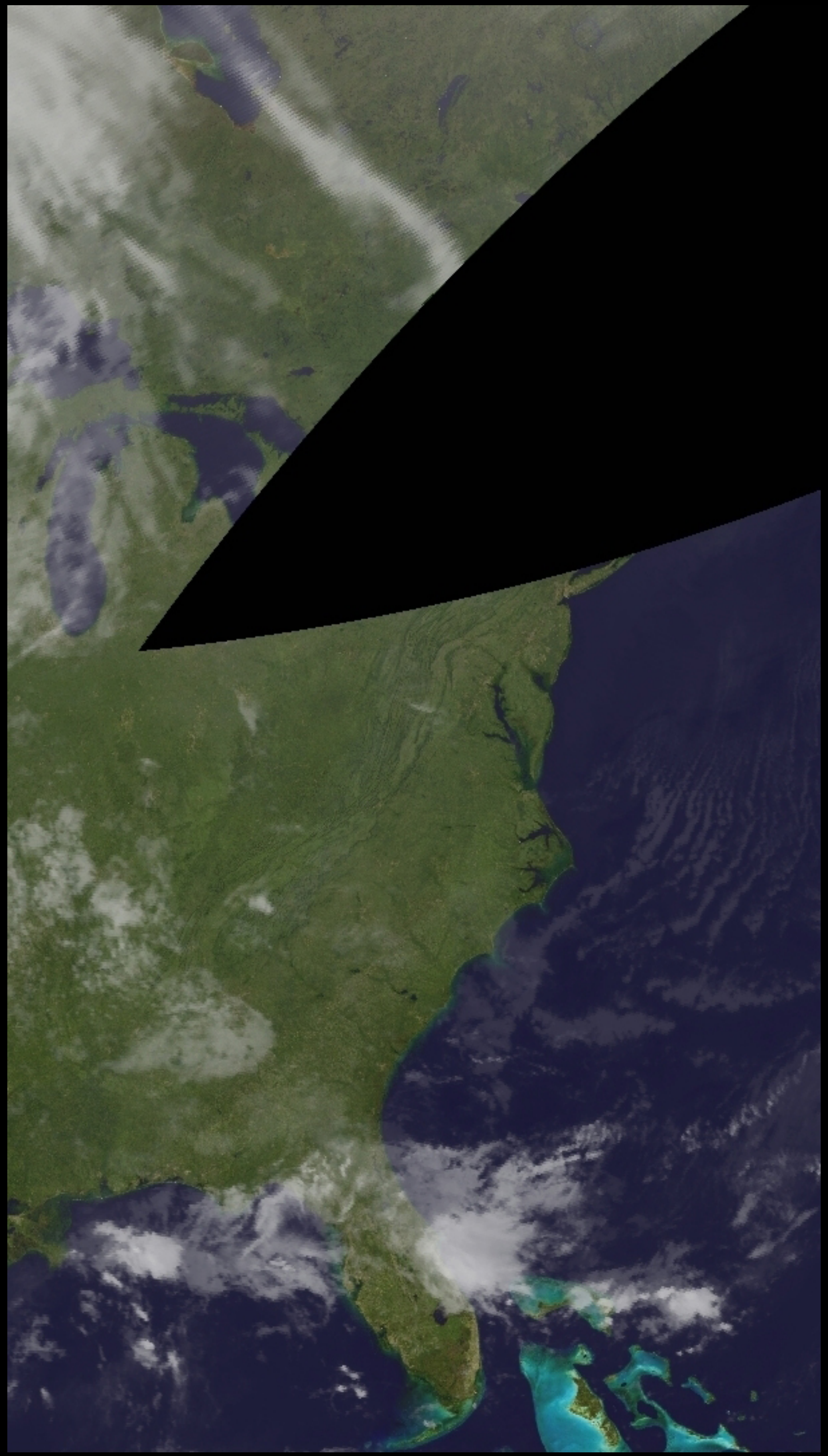
- Motivation
- State of Art
- World Model
- Modified Boids Algorithm
- **Results**
- Conclusion

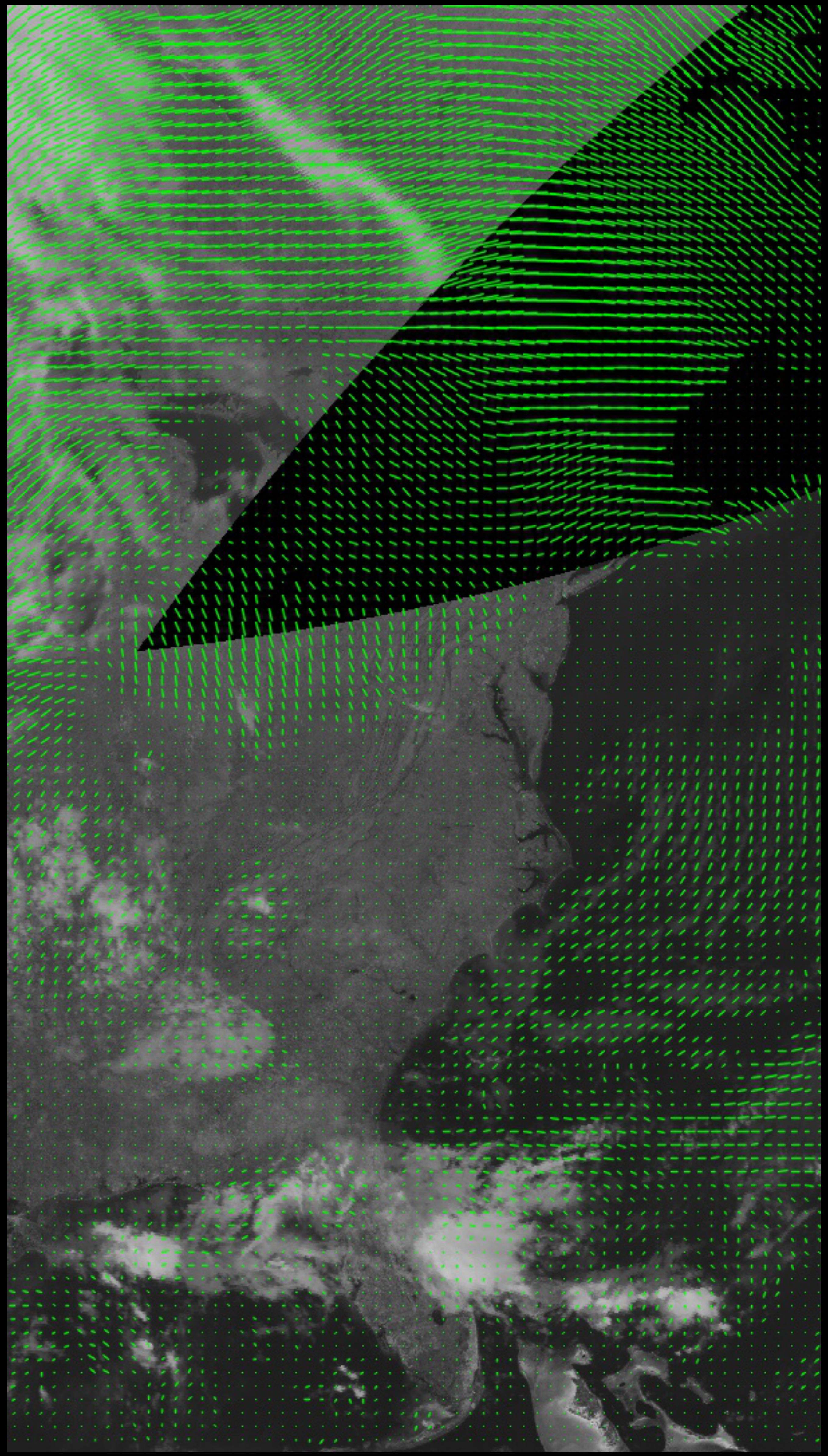
Dataset



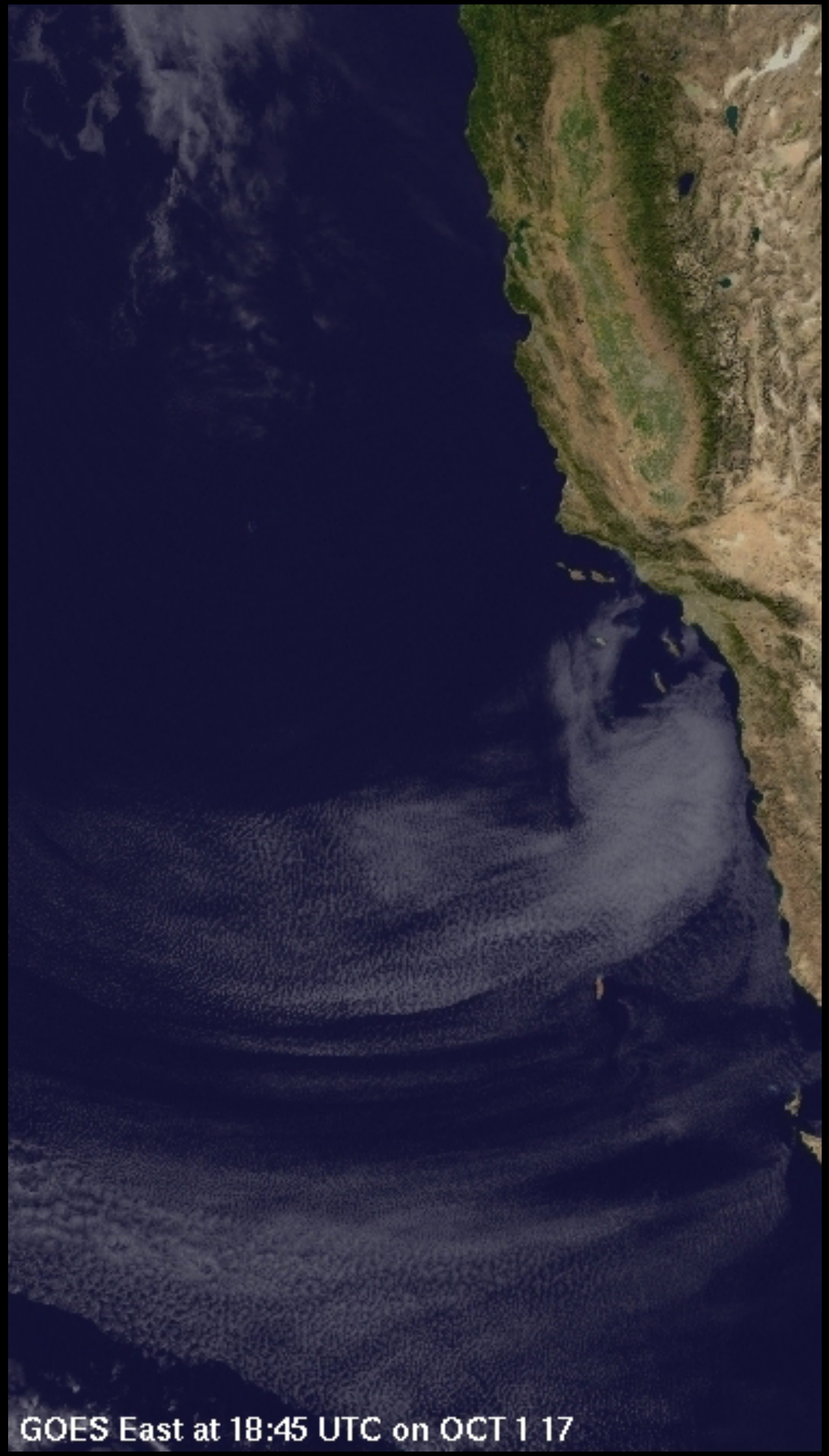
NOAA Archive

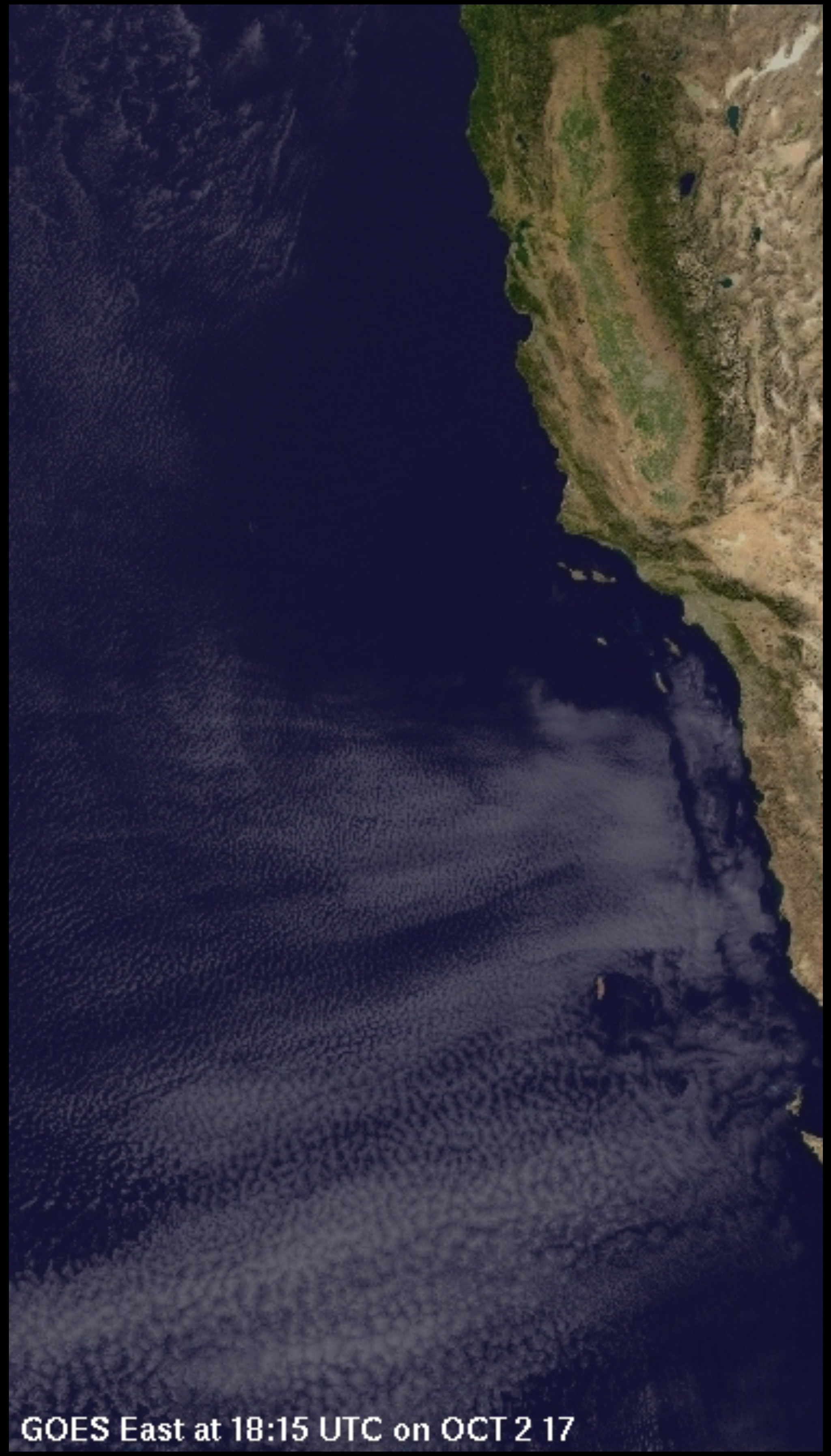
Dataset Anomalies



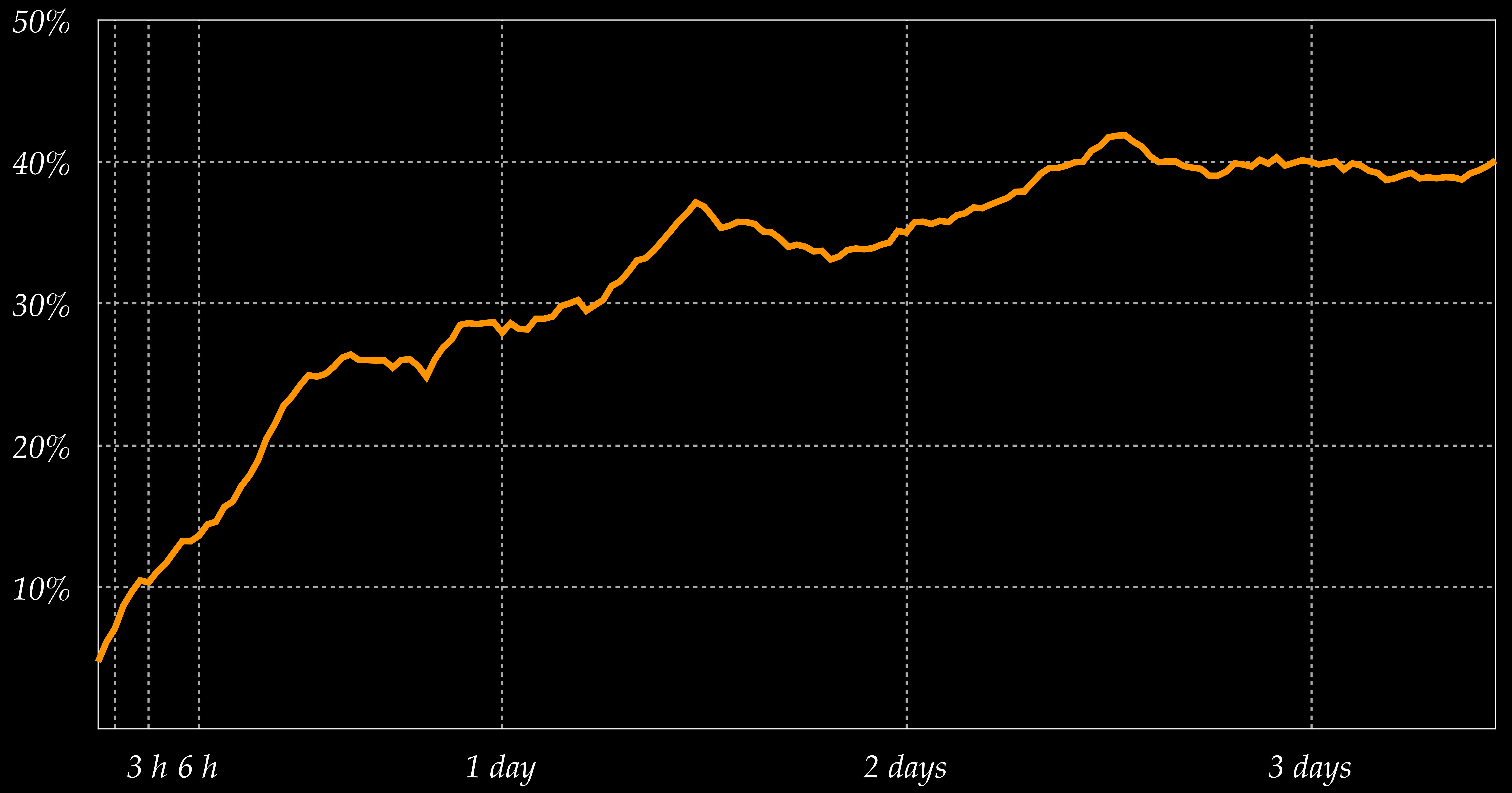


Dataset Anomalies

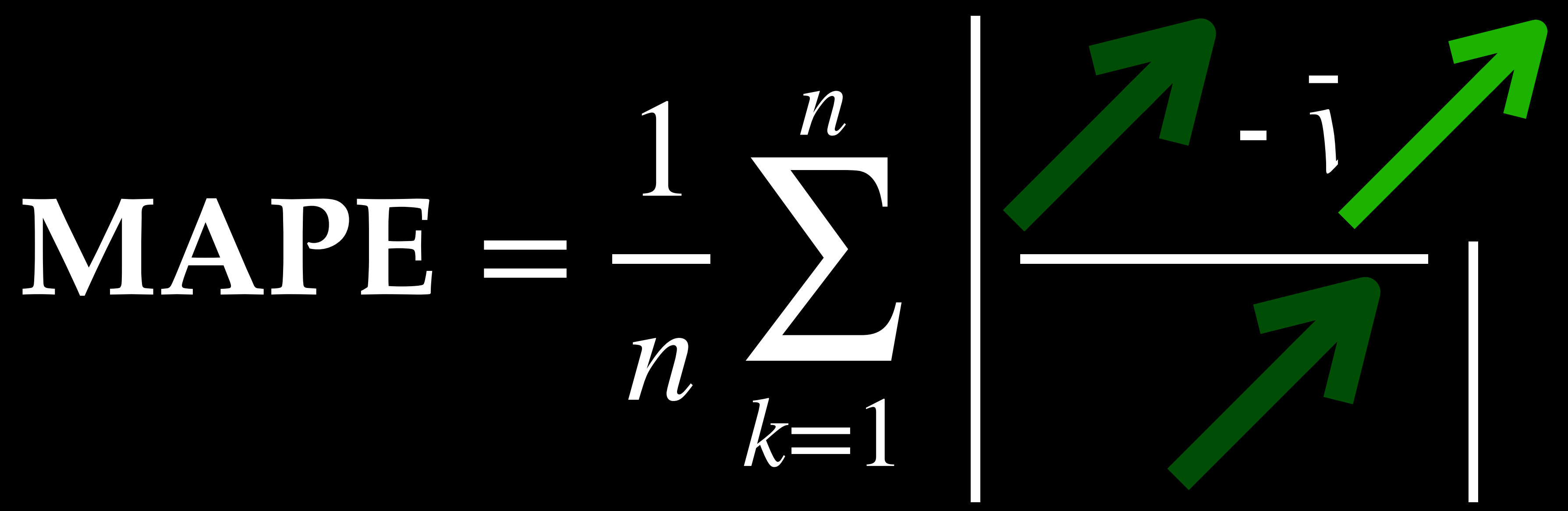


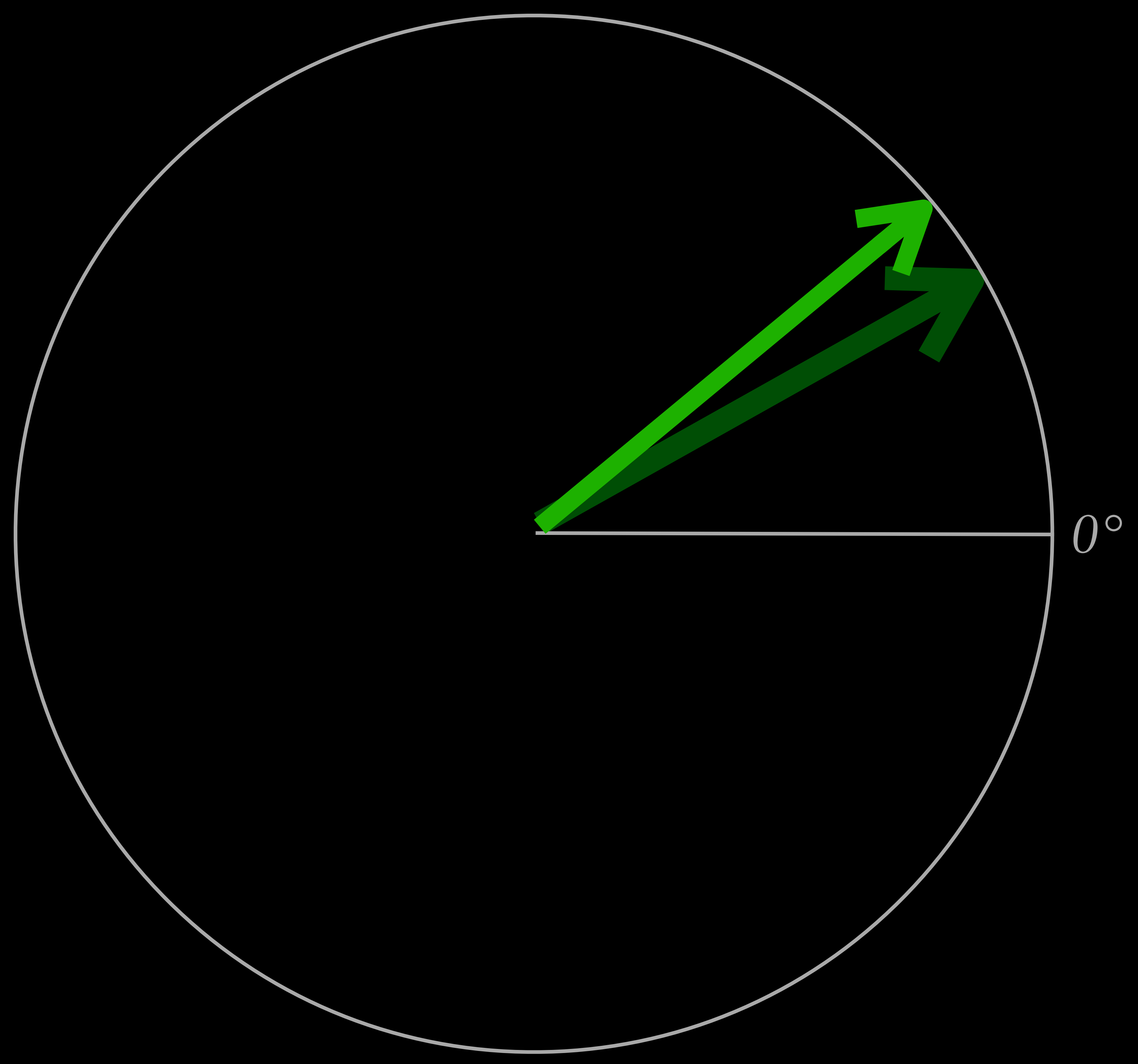


MAPE score



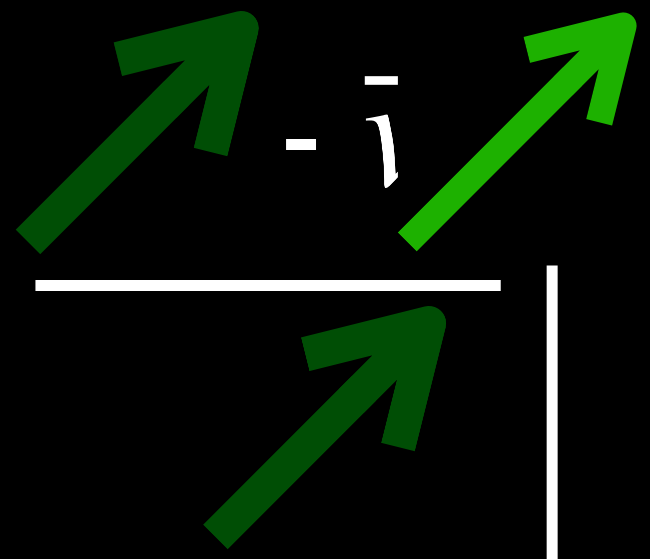
Bias



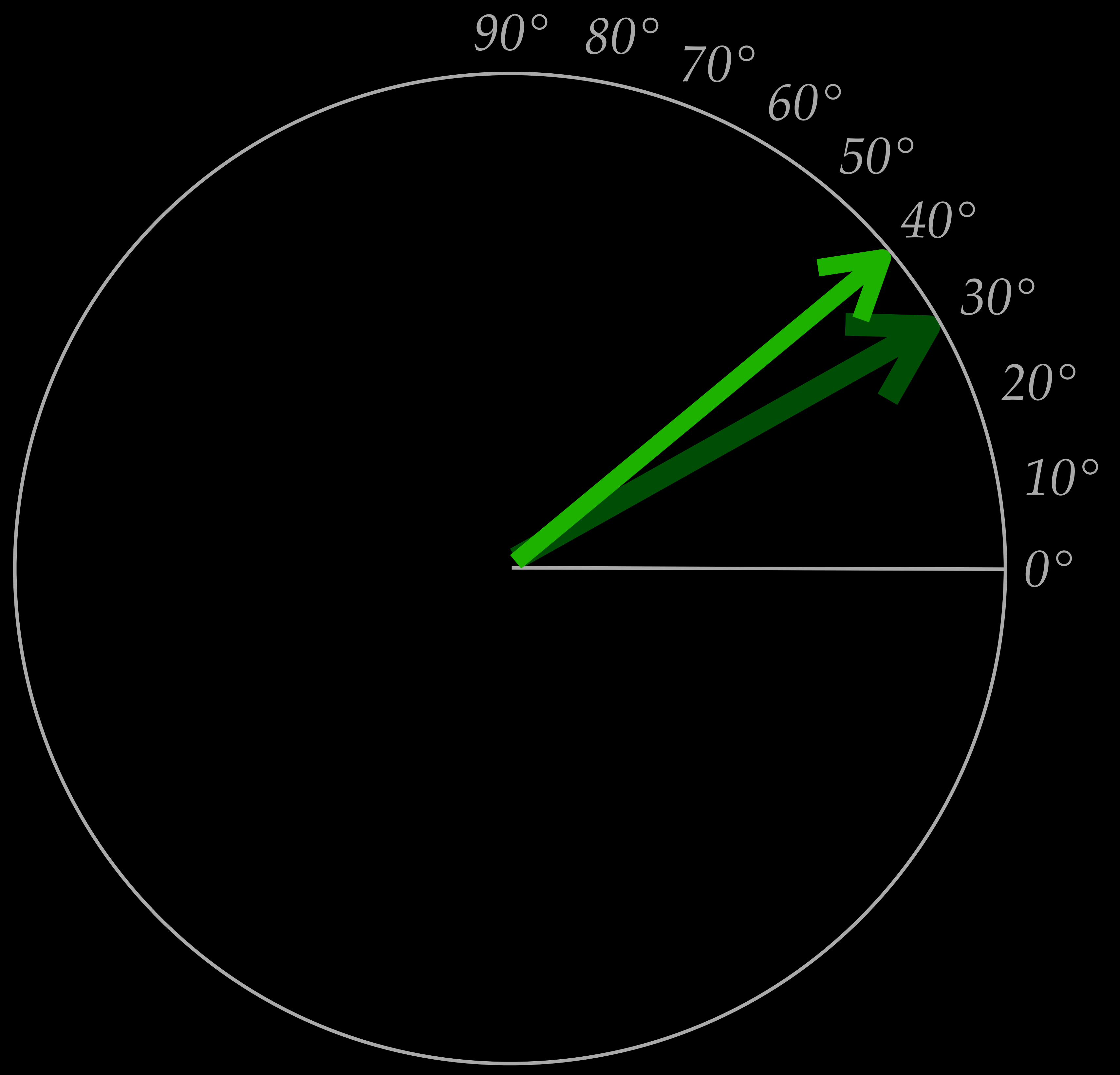


Bias

$$\mathrm{MAPE} = \frac{1}{n}\sum_{k=1}^{n}\left| \quad \right|$$

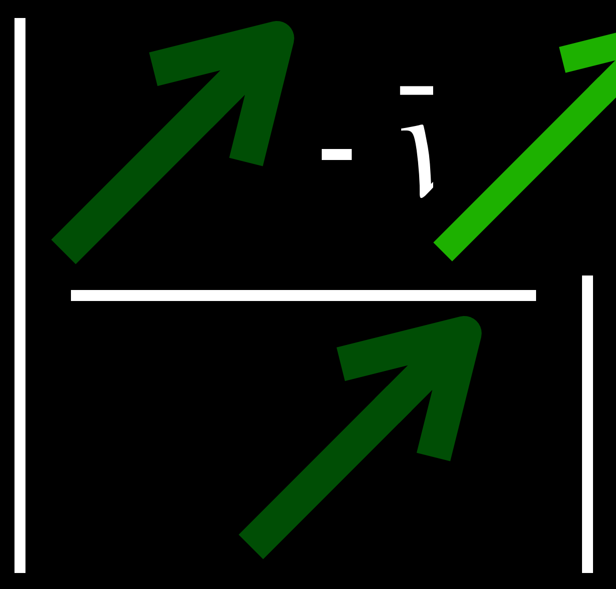


$$\left|\frac{30^\circ - 40^\circ}{30^\circ}\right| = 0.33$$



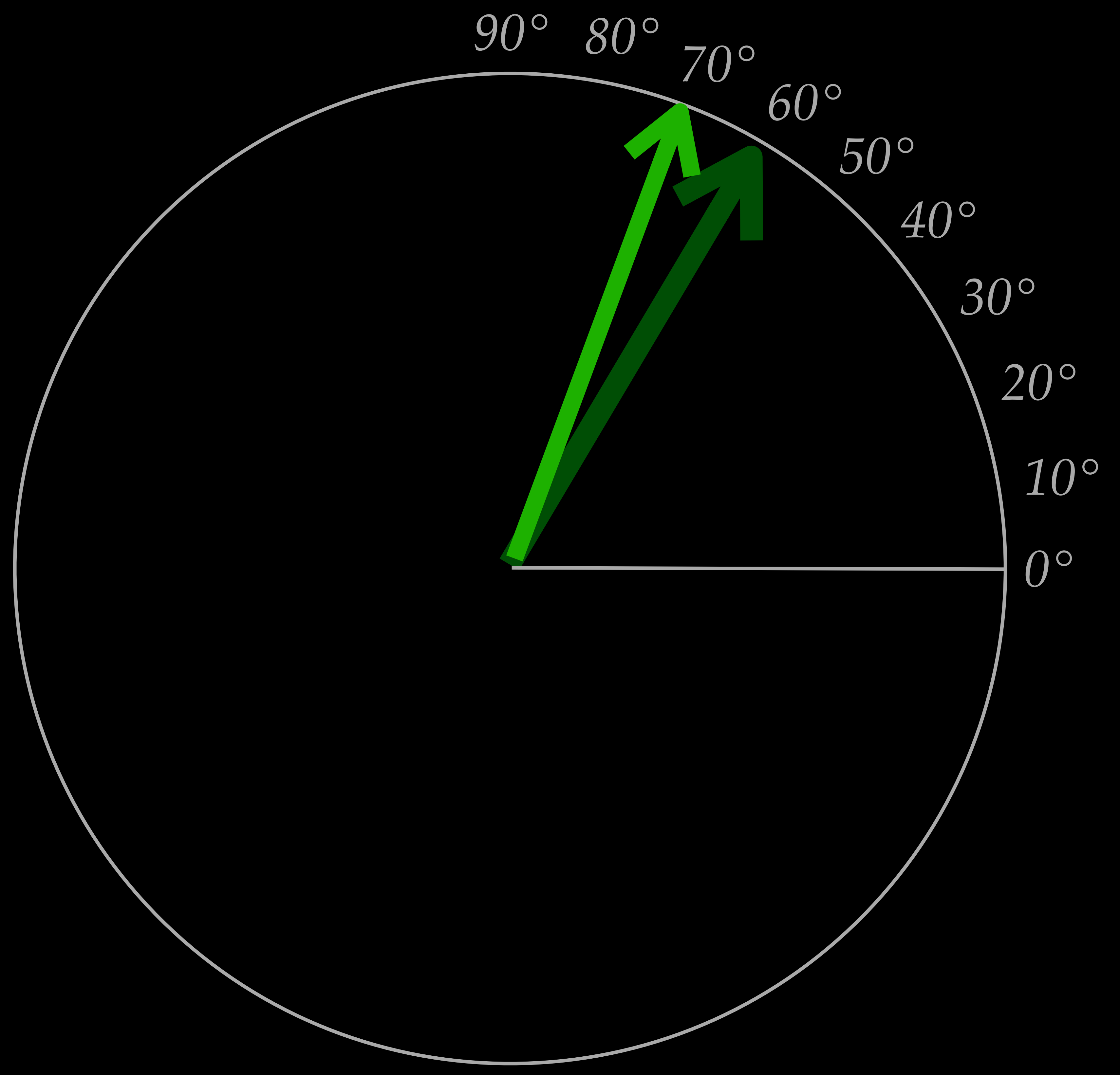
Bias

$$\mathbf{MAPE} = \frac{1}{n}\sum_{k=1}^{n}$$



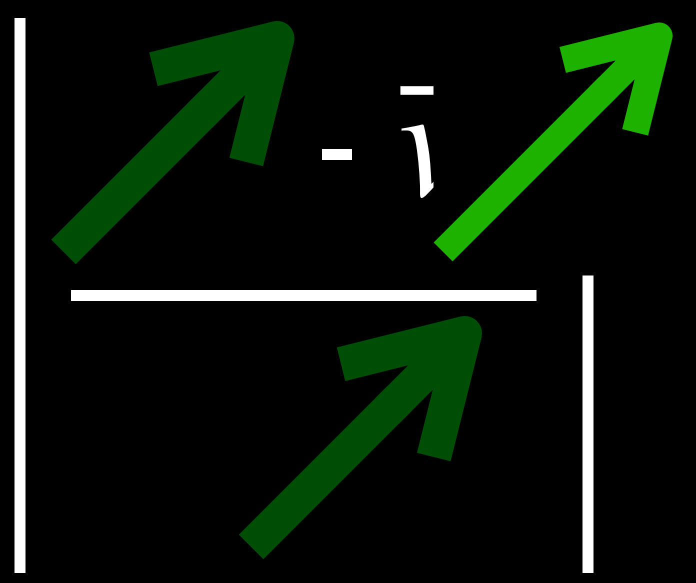
$$\left|\frac{30^\circ - 40^\circ}{30^\circ}\right| = 0.33$$

$$\left|\frac{60^\circ - 70^\circ}{60^\circ}\right| = 0.16$$



Bias

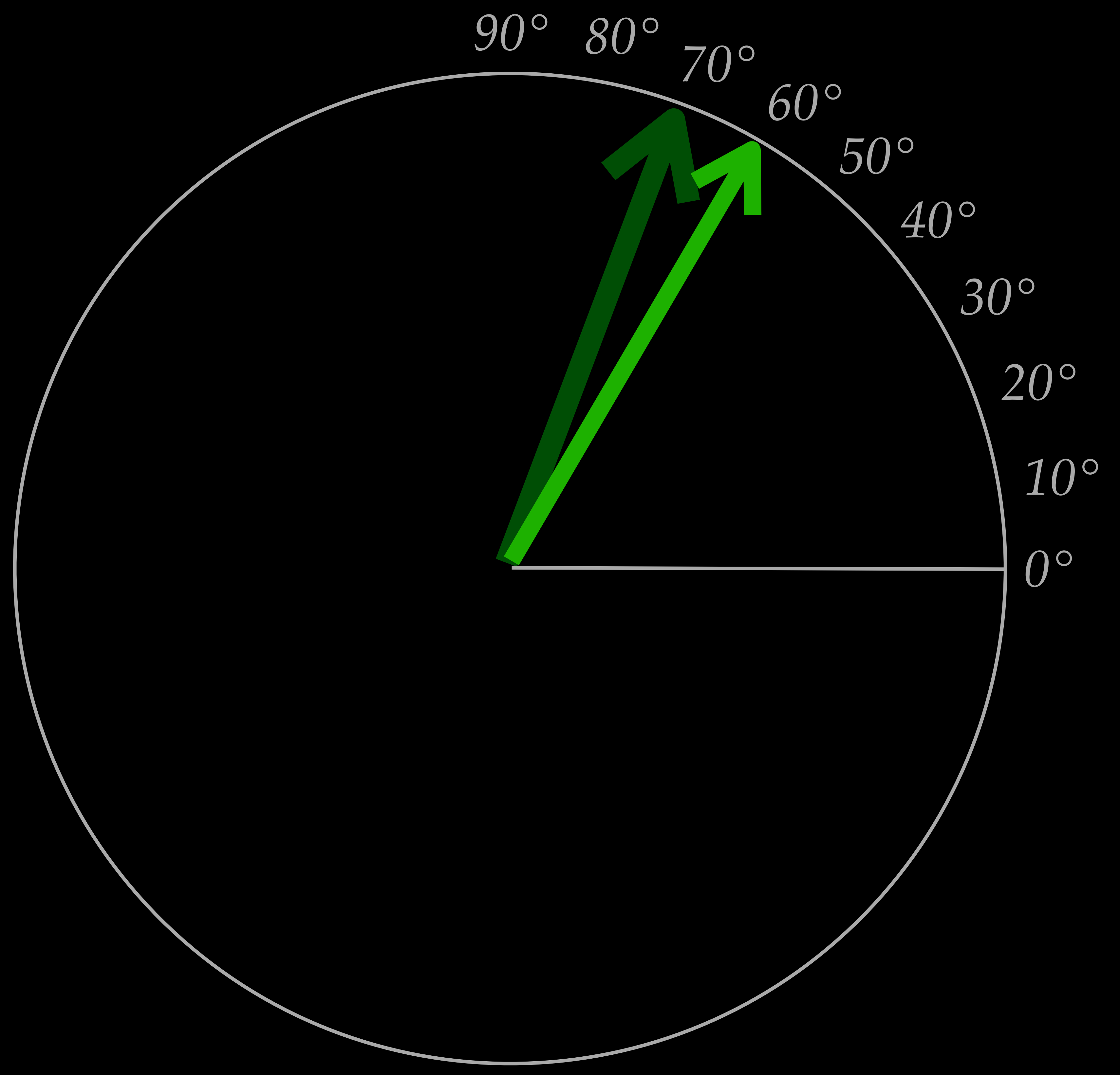
$$\mathbf{MAPE} = \frac{1}{n}\sum_{k=1}^{n}$$



$$\left|\frac{30^\circ - 40^\circ}{30^\circ}\right| = 0.33$$

$$\left|\frac{60^\circ - 70^\circ}{60^\circ}\right| = 0.16$$

$$\left|\frac{70^\circ - 60^\circ}{70^\circ}\right| = 0.14$$



Bias

$$\mathbf{MAPE} = \frac{1}{n}\sum_{k=1}^{n}\left|\frac{\nearrow - \nearrow}{\nearrow}\right| \qquad \mathbf{MAnE}^{*} = \frac{\theta(\nearrow,\nearrow)}{\pi}$$

$$\left|\frac{30^\circ - 40^\circ}{30^\circ}\right| = 0.33 \qquad \frac{\theta(30^\circ,40^\circ)}{\pi} = 0.05$$

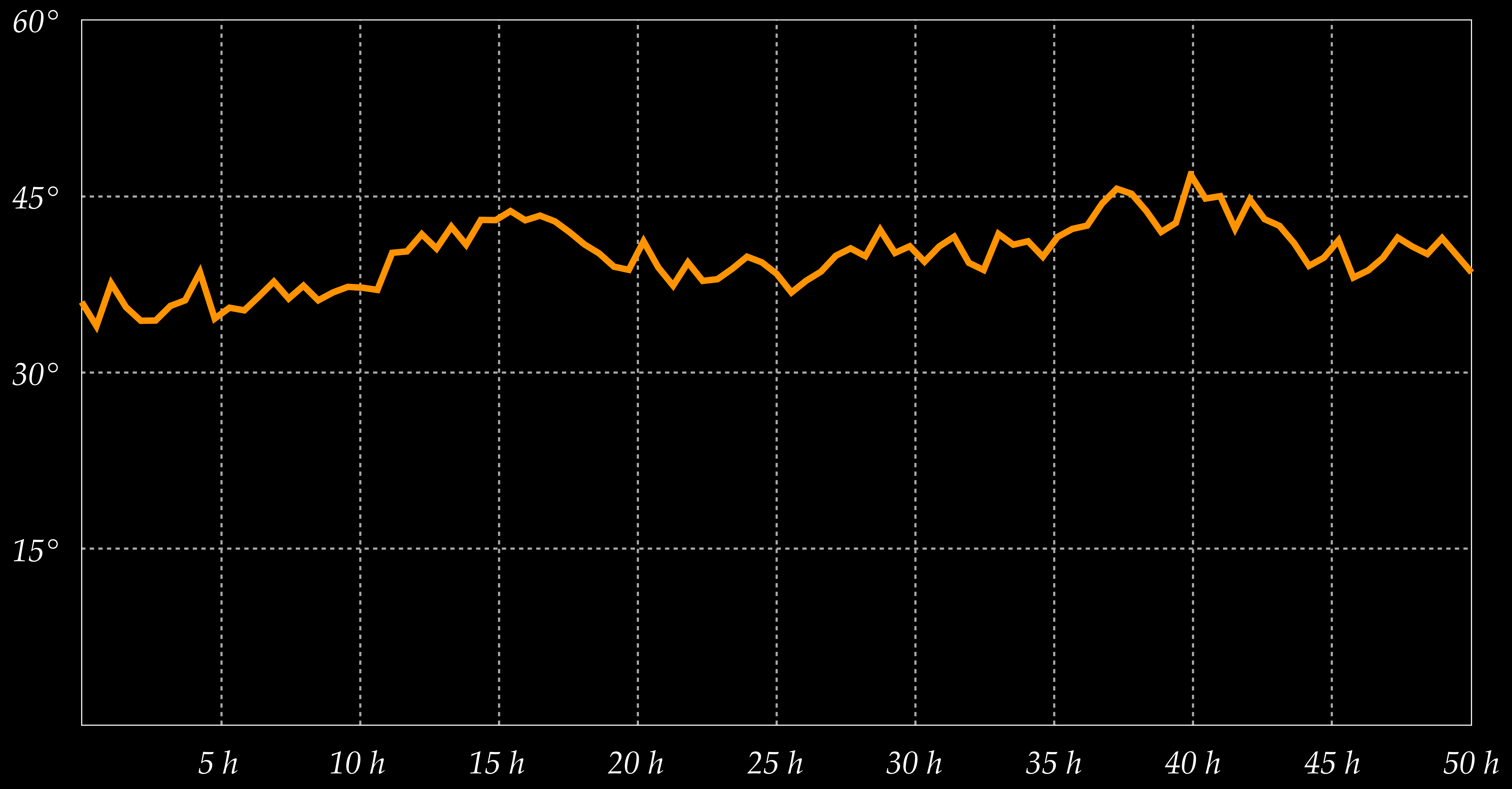
$$\left|\frac{60^\circ - 70^\circ}{60^\circ}\right| = 0.16 \qquad \frac{\theta(60^\circ,70^\circ)}{\pi} = 0.05$$

$$\left|\frac{70^\circ - 60^\circ}{70^\circ}\right| = 0.14 \qquad \frac{\theta(70^\circ,60^\circ)}{\pi} = 0.05$$

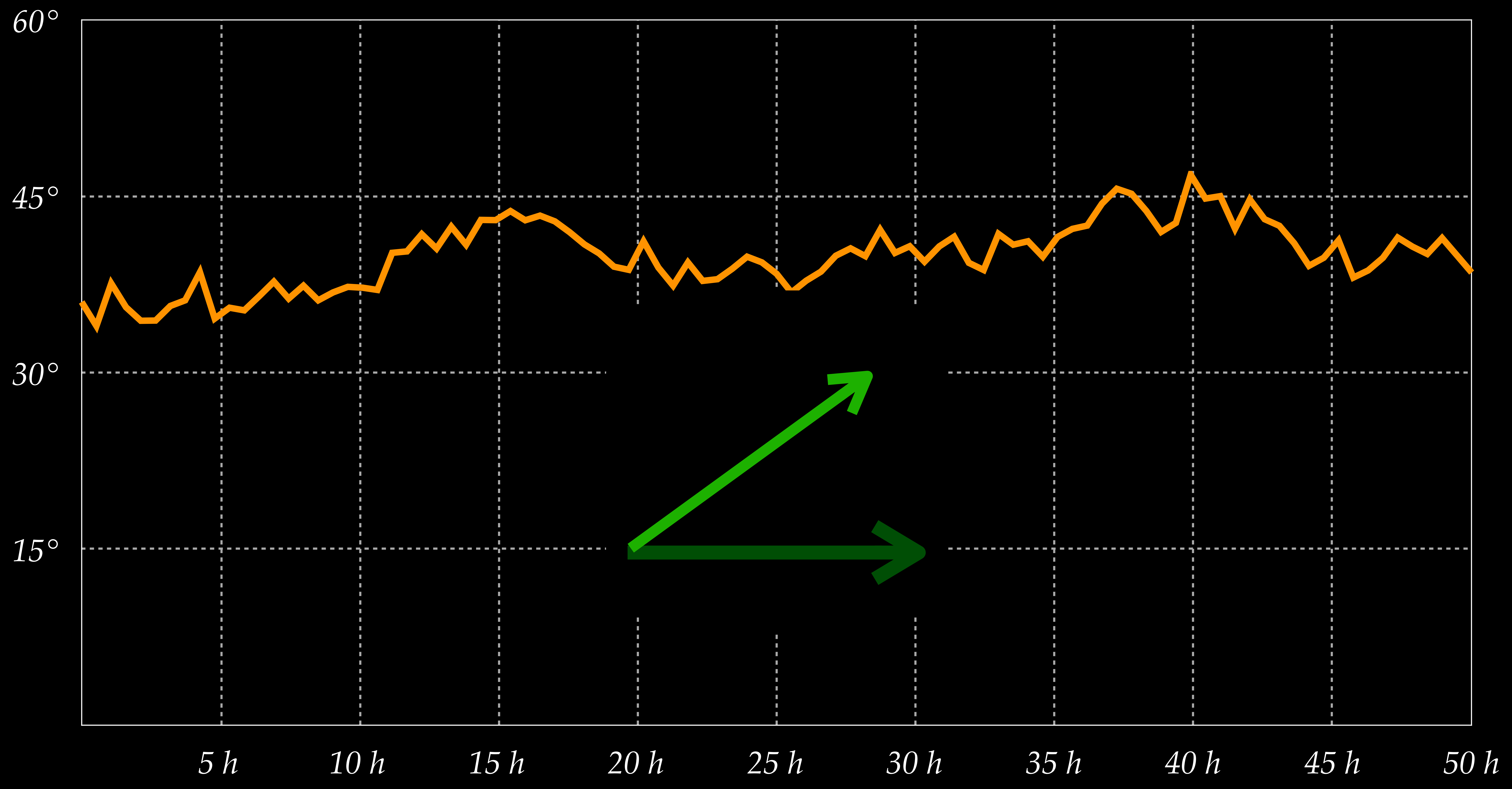
*Mean Angle Error

$\theta \in [0,\pi]$

Mean Angle Error



Mean Angle Error



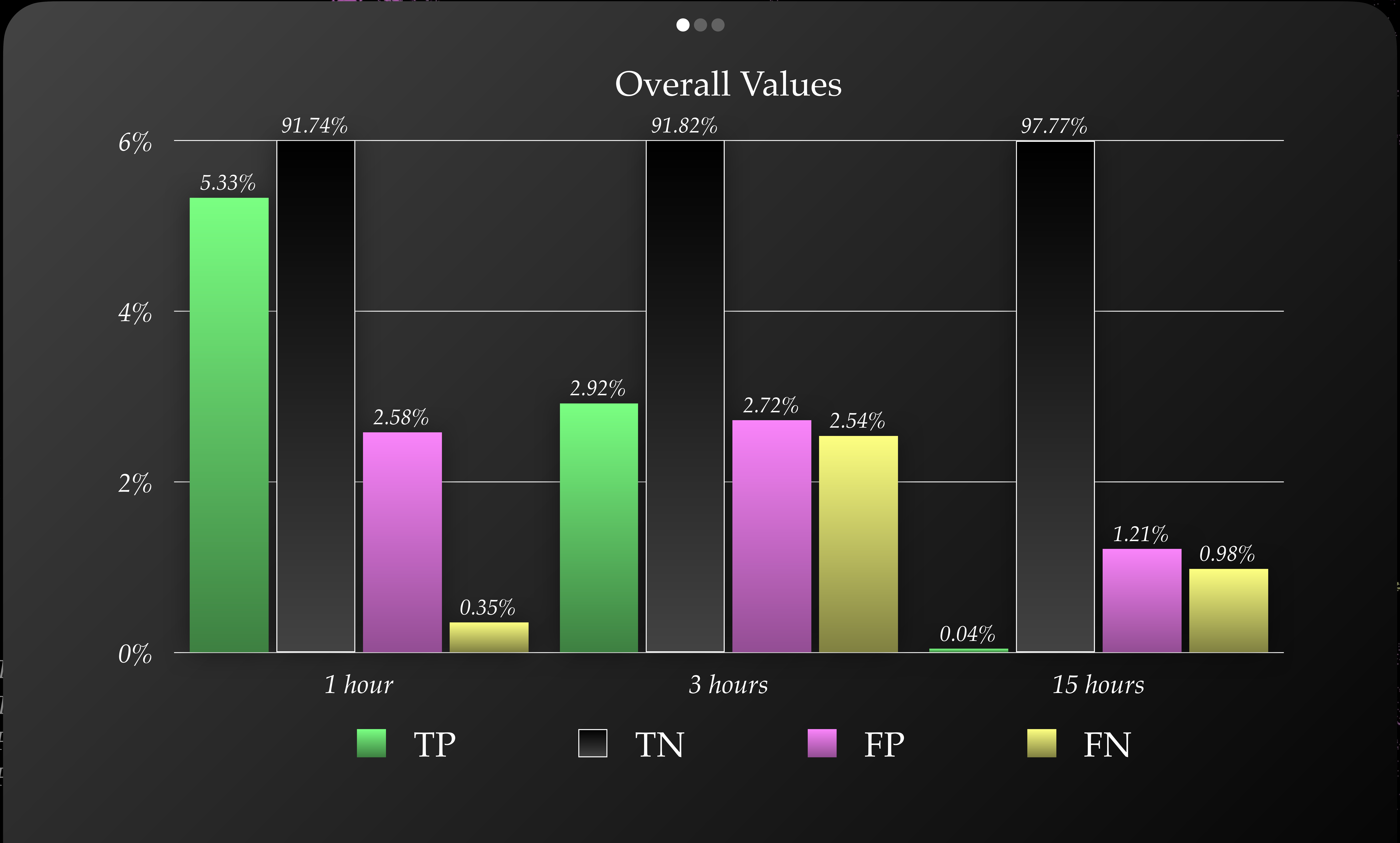
CM • 1 hour forecast

CM • 3 hours forecast

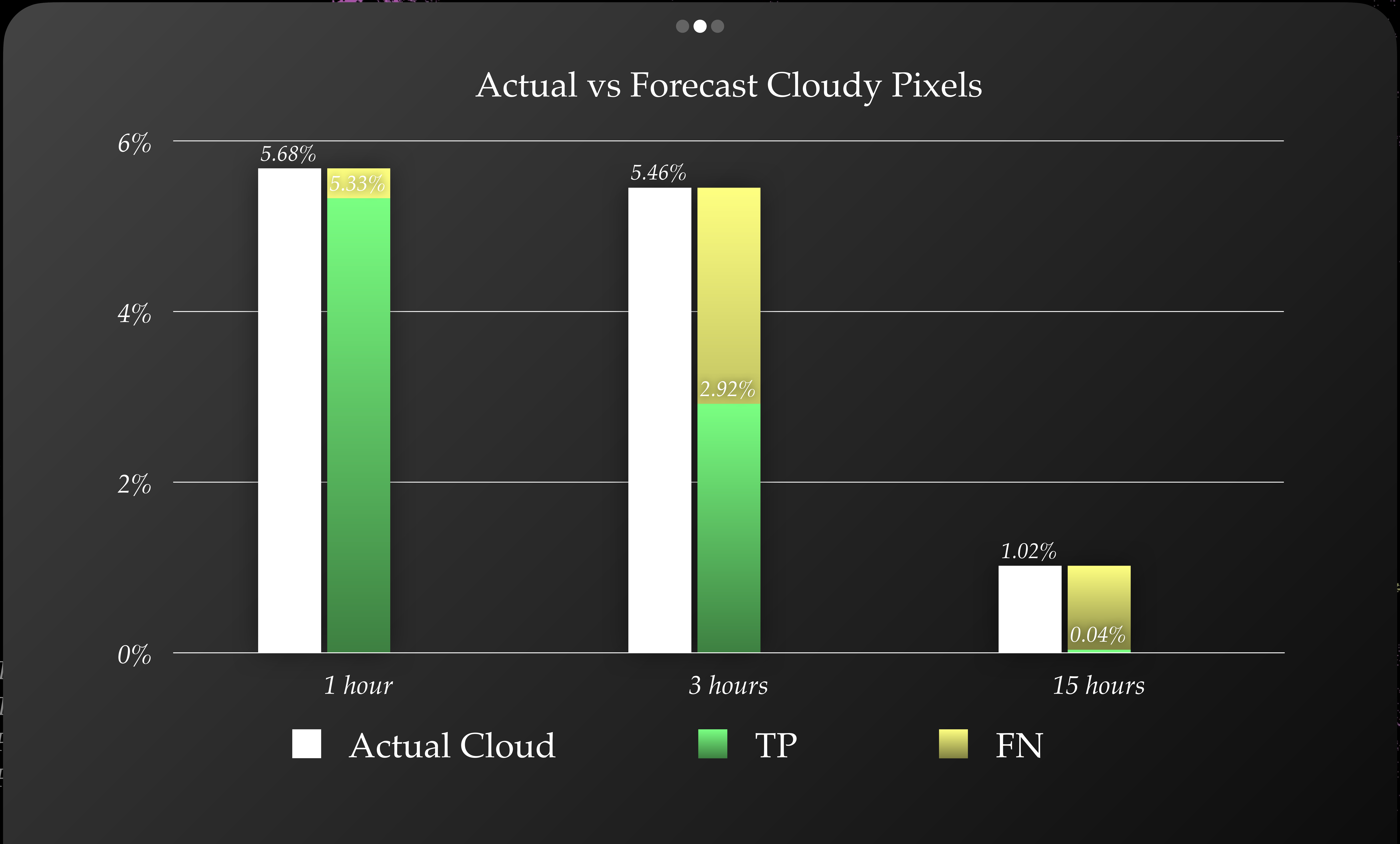
CM • 15 hours forecast

footnote

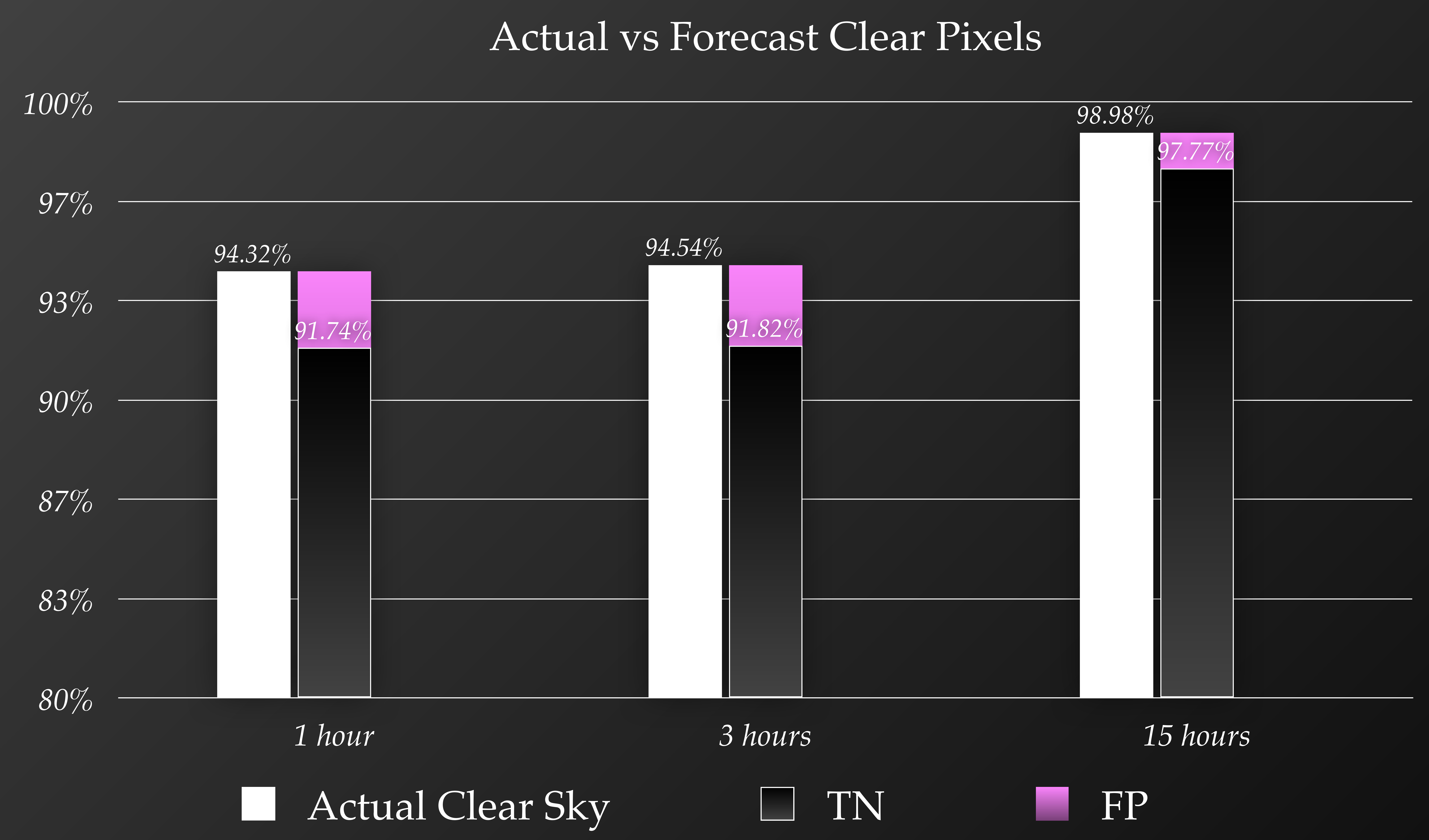
CM • 15 hours forecast



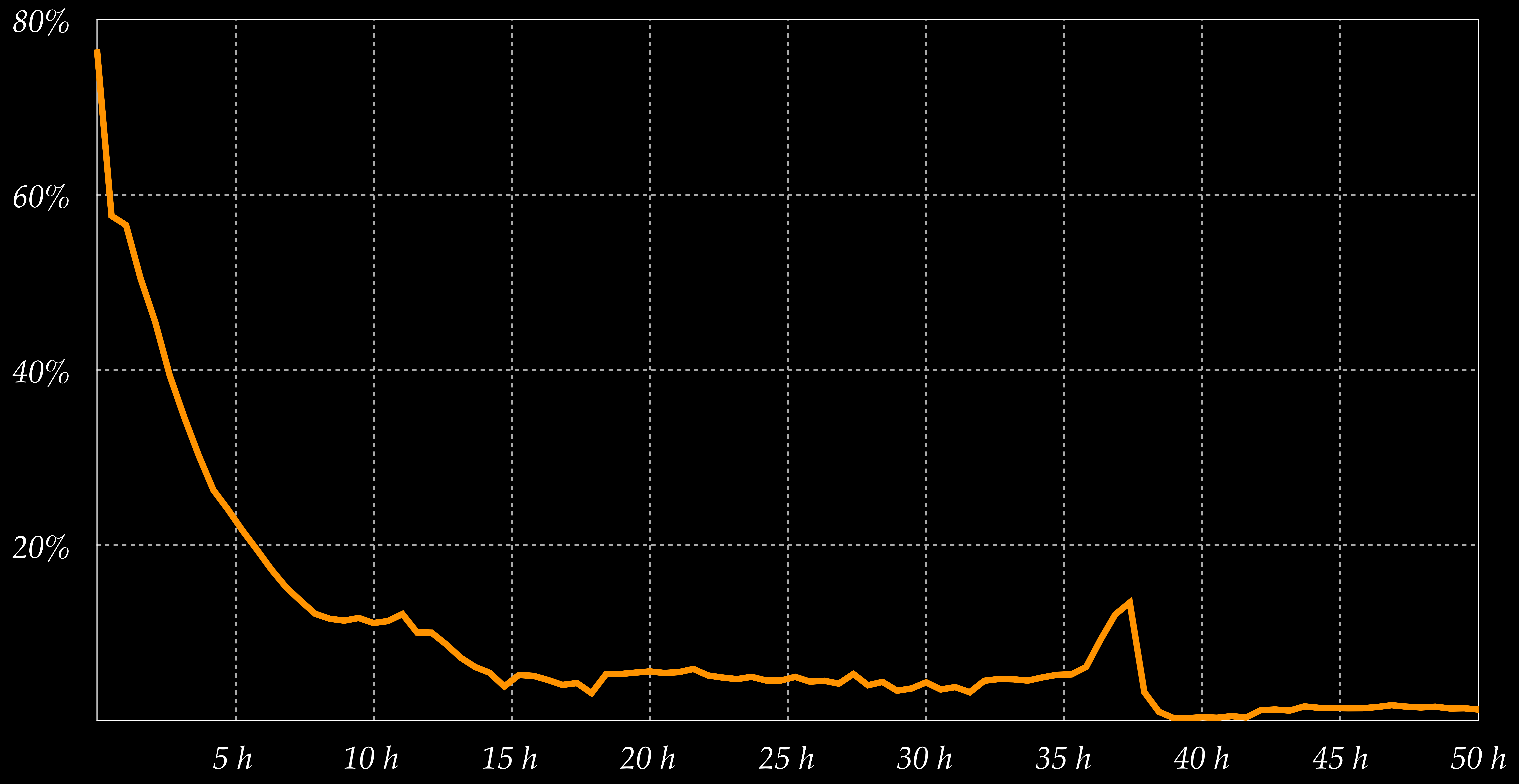
CM • 15 hours



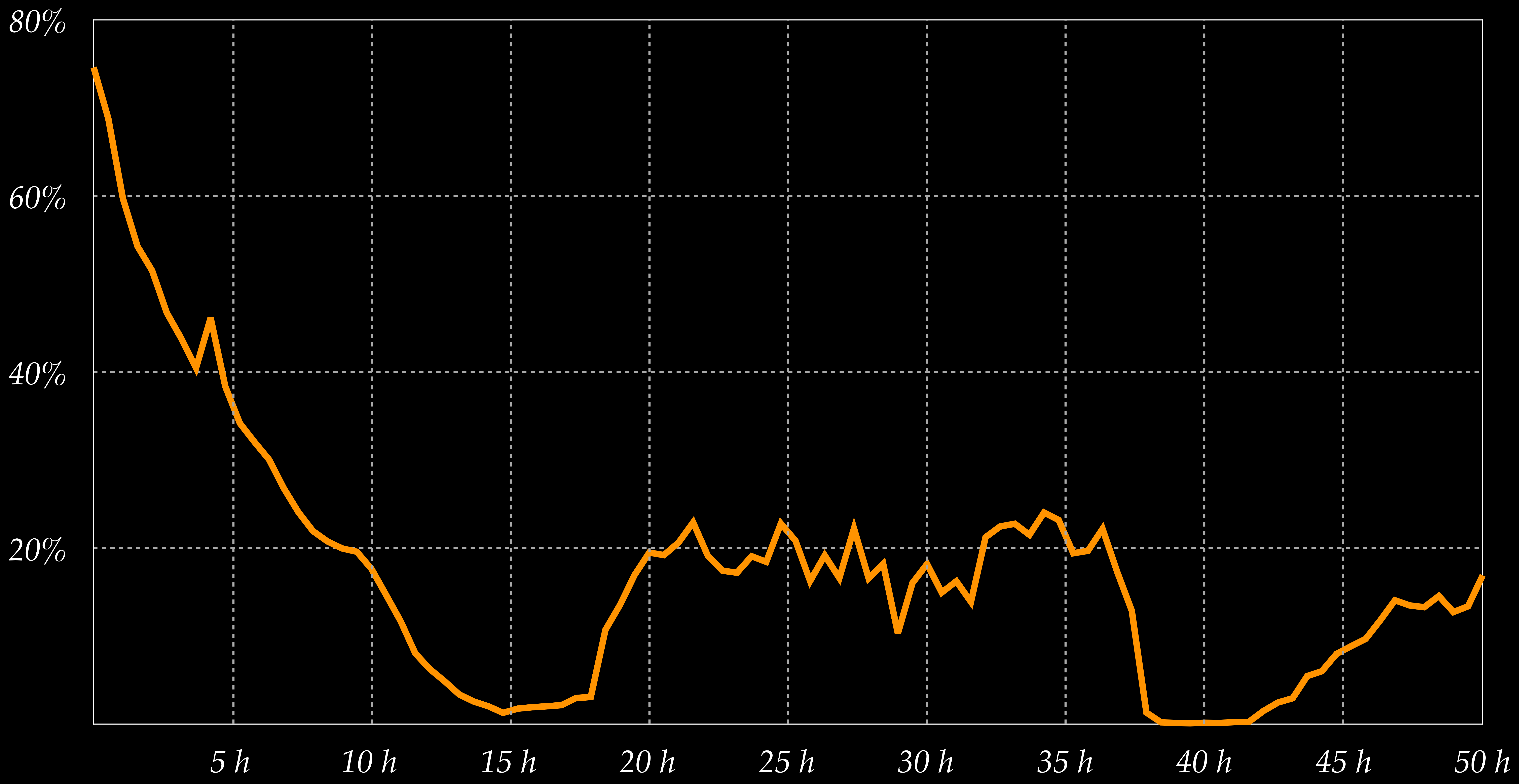
Actual vs Forecast Cloudy Pixels
6%
4%
2%
0%
5.68%
5.33%
5.46%
2.92%
1.02%
0.04%
1 hour
3 hours
15 hours
Actual Cloud
TP
FN


Actual vs Forecast Clear Pixels
100%
97%
93%
90%
87%
83%
80%
94.32%
91.74%
94.54%
91.82%
98.98%
97.77%
1 hour
3 hours
15 hours
Actual Clear Sky
TN
FP

Recall Score = TP/(TP + FN)



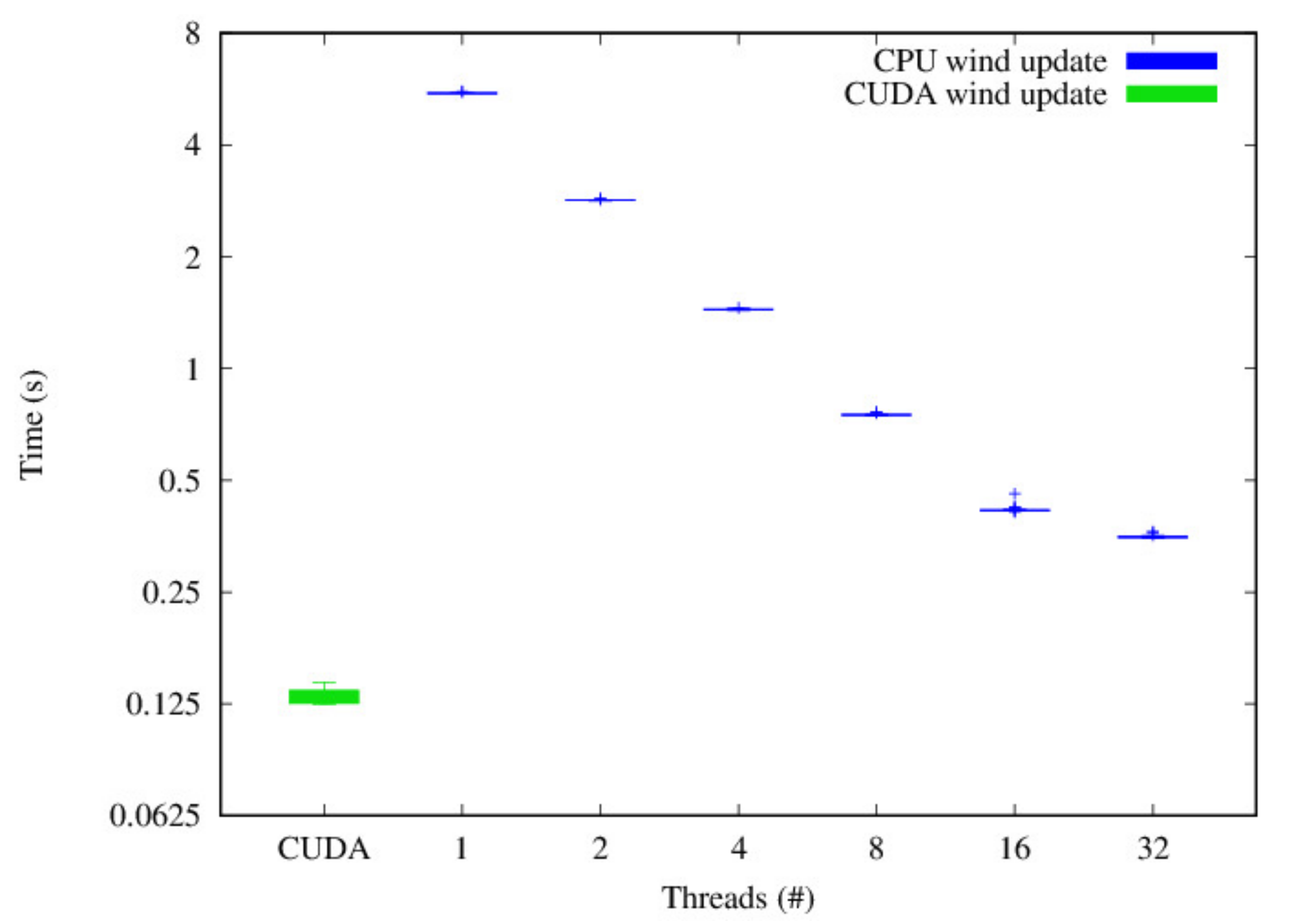
Precision Score = TP/(TP + FP)



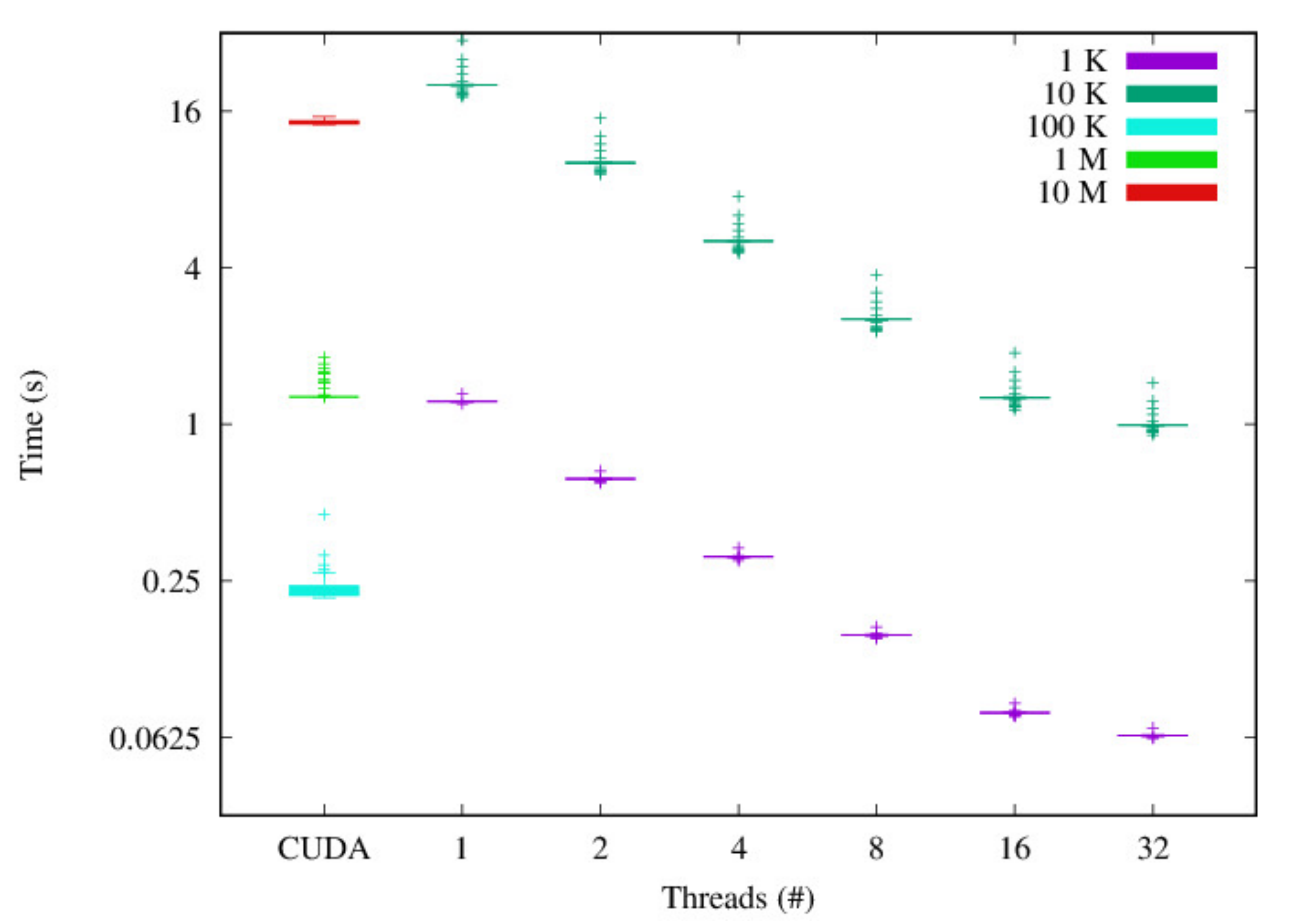
The sequential algorithm forecasts one frame containing 10 000 particles in around 16 seconds. Parallel versions have a **shorter execution time** and can process orders of magnitude more particles.

The number of $\vec{v}$ is **upper bound**.
The number of C *particles* is **not**.

Wind map update execution time



Cloud particles update execution time



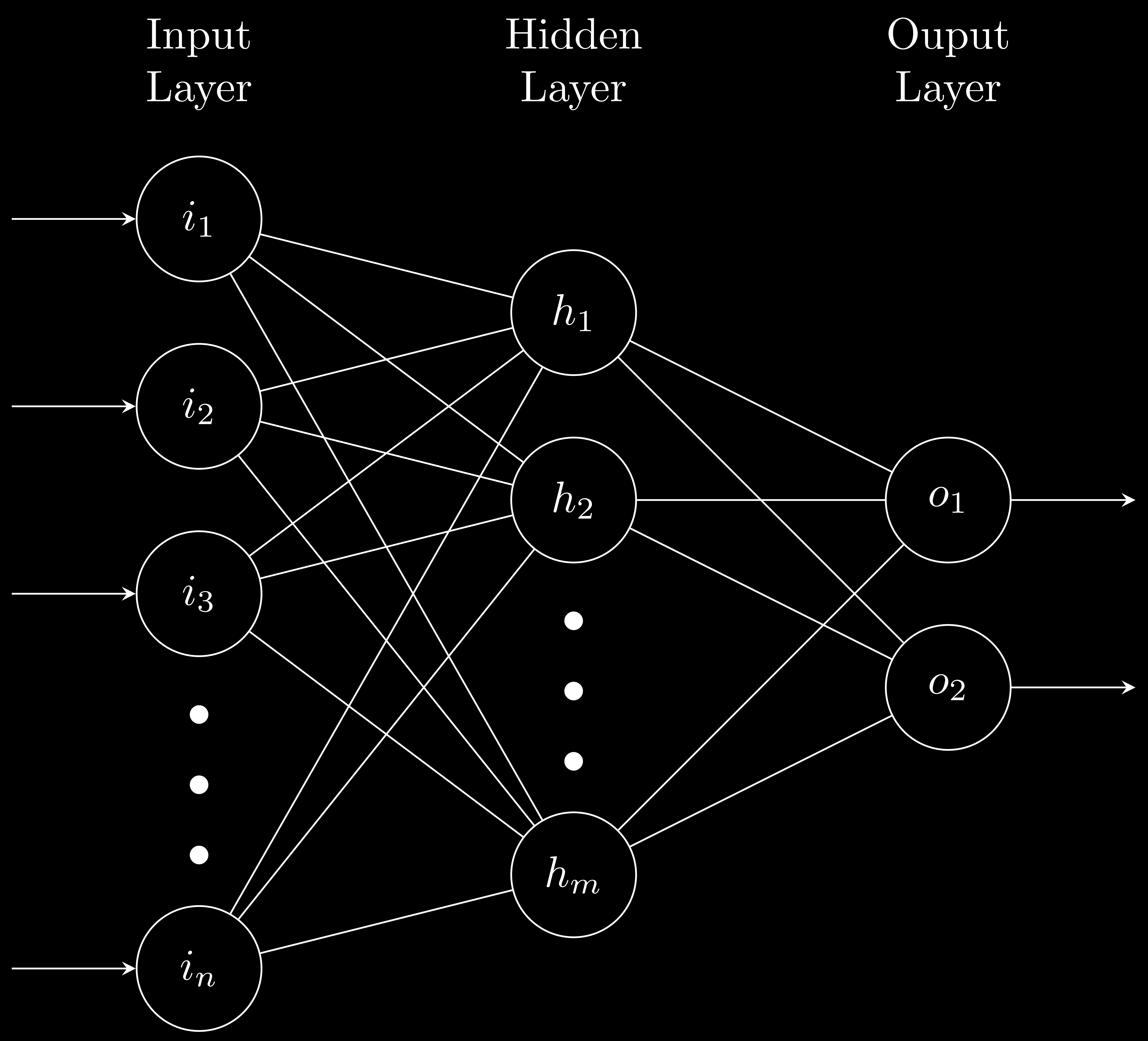
Isolate C using a thresholding method.

Generate $\vec{v}$ using the *Optical Flow Procedure*.

Train the network on $\vec{v}$ time series.

Feed Forward Back Propagation Neural Network

(Rojas, 1996)



Neural Network results

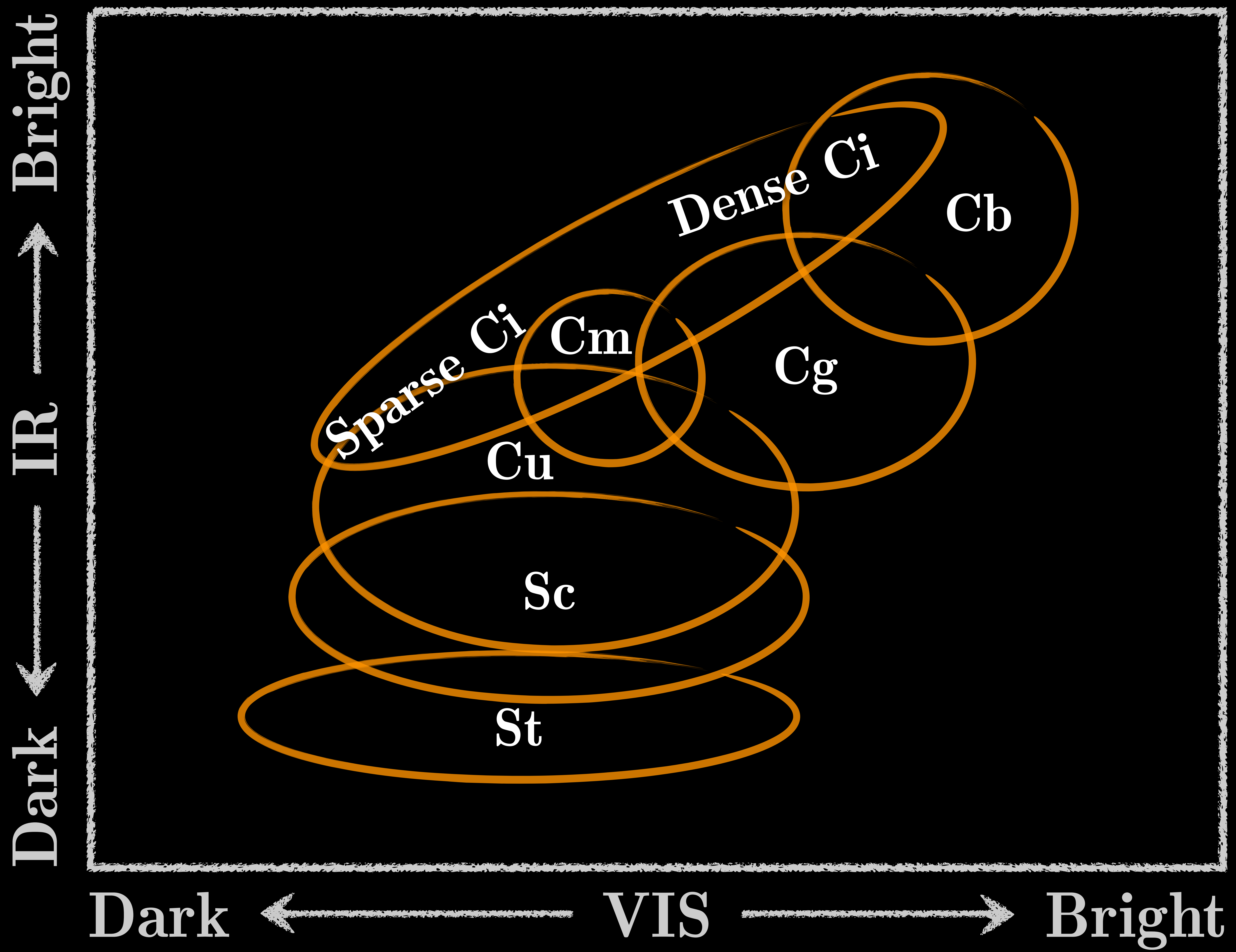
Mean Squared Error is between 4.5×10^{-6} and 9.76×10^{-6} depending on **input size**.

Inefficient compared to the Modified Boids Algorithm because the network was trained on **small sections** of images.

The network is overfitted because it needs **vast amounts** of resources to train and forecast.

Correlation with Irradiance

(Mamoru, 2002)



Ci - High Thin Cirrus

Cm - Medium Clouds

St - Stratus, Mist

Sc - Stratocumulus

Cu - Cumulus

Cb - Cumulonimbus

Cg - Cumulus Congestus

(European Space Agency, 2015)

Band	Resolution	Wavelength	Purpose
1	60 m	443 nm	Aerosol Detection
2	10 m	490 nm	Color Blue
3	10 m	560 nm	Color Green
4	10 m	665 nm	Color Red
5	20 m	705 nm	Vegetation
6	20 m	740 nm	Vegetation
7	20 m	783 nm	Vegetation
8	10 m	842 nm	Near Infrared
8A	20 m	865 nm	Vegetation
9	60 m	945 nm	Water Vapor
10	60 m	1375 nm	Cirrus Cloud
11	20 m	1610 nm	Snow-Ice-Cloud
12	20 m	2190 nm	Snow-Ice-Cloud

(Menzel, 2006)

Band	Resolution	Wavelength	Purpose
1	60 m	443 nm	Aerosol Detection
2	10 m	490 nm	Color Blue
3	10 m	560 nm	Color Green
4	10 m	665 nm	Color Red
5	20 m	705 nm	Vegetation
6	20 m	740 nm	Vegetation
7	20 m	783 nm	Vegetation
8	10 m	842 nm	Near Infrared
8A	20 m	865 nm	Vegetation
9	60 m	945 nm	Water Vapor
10	60 m	1375 nm	Cirrus Cloud
11	20 m	1610 nm	Snow-Ice-Cloud
12	20 m	2190 nm	Snow-Ice-Cloud

(Menzel, 2006)

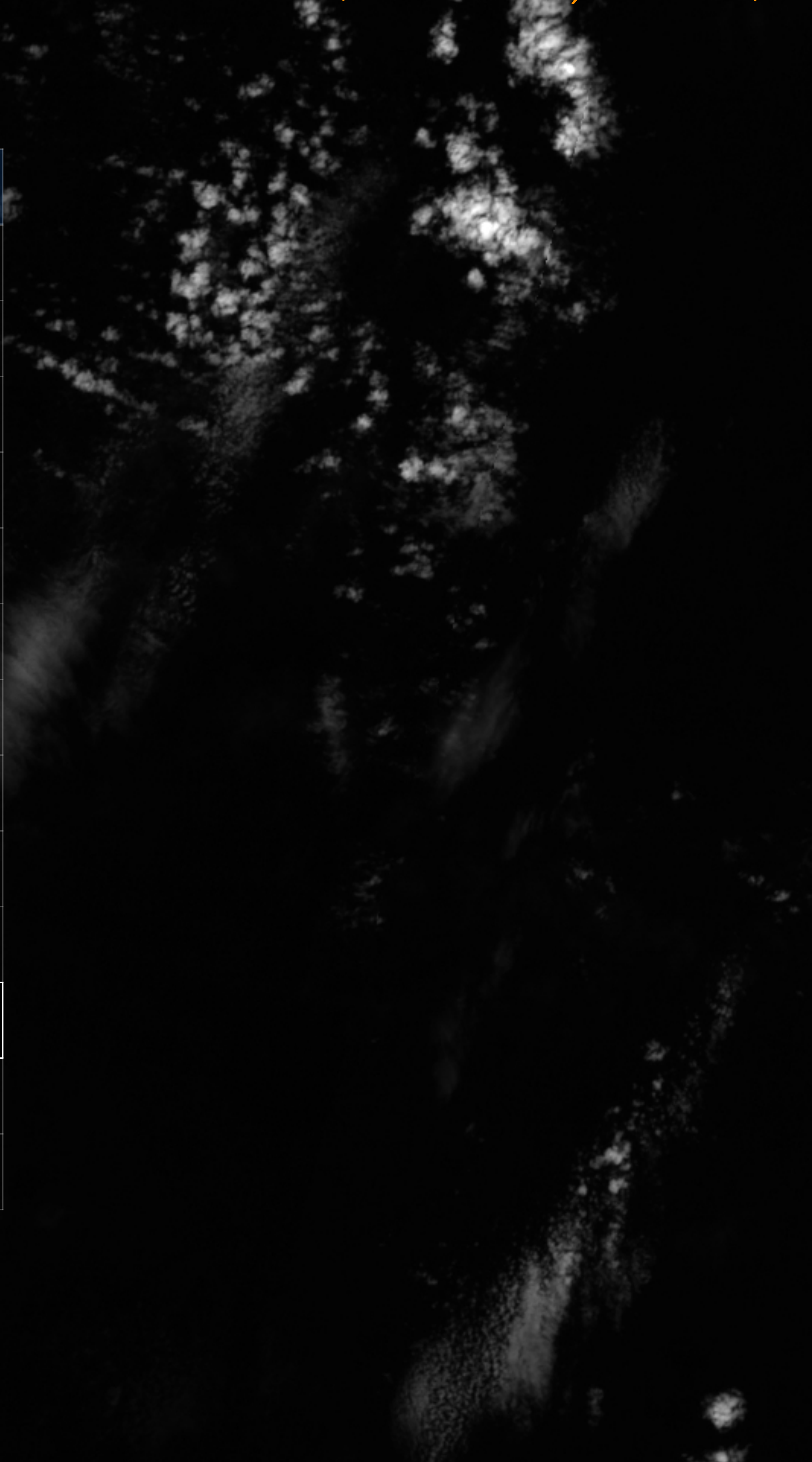
Band	Resolution	Wavelength	Purpose
1	60 m	443 nm	Aerosol Detection
2	10 m	490 nm	Color Blue
3	10 m	560 nm	Color Green
4	10 m	665 nm	Color Red
5	20 m	705 nm	Vegetation
6	20 m	740 nm	Vegetation
7	20 m	783 nm	Vegetation
8	10 m	842 nm	Near Infrared
8A	20 m	865 nm	Vegetation
9	60 m	945 nm	Water Vapor
10	60 m	1375 nm	Cirrus Cloud
11	20 m	1610 nm	Snow-Ice-Cloud
12	20 m	2190 nm	Snow-Ice-Cloud

(Menzel, 2006)

Band	Resolution	Wavelength	Purpose
1	60 m	443 nm	Aerosol Detection
2	10 m	490 nm	Color Blue
3	10 m	560 nm	Color Green
4	10 m	665 nm	Color Red
5	20 m	705 nm	Vegetation
6	20 m	740 nm	Vegetation
7	20 m	783 nm	Vegetation
8	10 m	842 nm	Near Infrared
8A	20 m	865 nm	Vegetation
9	60 m	945 nm	Water Vapor
10	60 m	1375 nm	Cirrus Cloud
11	20 m	1610 nm	Snow-Ice-Cloud
12	20 m	2190 nm	Snow-Ice-Cloud

(Menzel, 2006)

Band	Resolution	Wavelength	Purpose
1	60 m	443 nm	Aerosol Detection
2	10 m	490 nm	Color Blue
3	10 m	560 nm	Color Green
4	10 m	665 nm	Color Red
5	20 m	705 nm	Vegetation
6	20 m	740 nm	Vegetation
7	20 m	783 nm	Vegetation
8	10 m	842 nm	Near Infrared
8A	20 m	865 nm	Vegetation
9	60 m	945 nm	Water Vapor
10	**60 m**	**1375 nm**	**Cirrus Cloud**
11	20 m	1610 nm	Snow-Ice-Cloud
12	20 m	2190 nm	Snow-Ice-Cloud




Solar Platform
Timișoara

(Paulescu et al., 2010)

Date	Time	Global Irradiance	Diffuse Irradiance
...	...	...	...
2021-07-08	09:40:29	884.3	111.2
2021-08-10	09:30:41	865.1	182.7
2021-07-13	09:40:31	865.5	187.3
2021-07-15	09:30:39	870.7	242.4
2021-07-18	09:40:39	870.5	128.5
2021-07-20	09:30:41	377.7	342.8
2021-07-23	09:40:31	891.9	184.5
2021-07-25	09:30:39	849.5	161.5
2021-07-28	09:40:29	836	167.3
2021-07-30	09:30:41	852	131.3

W/m^2

Date	Time	Global Irradiance	Diffuse Irradiance
...	...	...	...
2021-07-08	09:40:29	884.3	111.2
2021-08-10	09:30:41	865.1	182.7
2021-07-13	09:40:31	865.5	187.3
2021-07-15	09:30:39	870.7	242.4
2021-07-18	09:40:39	870.5	128.5
2021-07-20	09:30:41	377.7	342.8
2021-07-23	09:40:31	891.9	184.5
2021-07-25	09:30:39	849.5	161.5
2021-07-28	09:40:29	836	167.3
2021-07-30	09:30:41	852	131.3

Date	Time	Global Irradiance	Diffuse Irradiance	Cirrus Band
...	...	...	...	...
2021-07-08	09:40:29	884.3	111.2	14
2021-08-10	09:30:41	865.1	182.7	18
2021-07-13	09:40:31	865.5	187.3	18
2021-07-15	09:30:39	870.7	242.4	16
2021-07-18	09:40:39	870.5	128.5	16
2021-07-20	09:30:41	377.7	342.8	80
2021-07-23	09:40:31	891.9	184.5	44
2021-07-25	09:30:39	849.5	161.5	18
2021-07-28	09:40:29	836	167.3	14
2021-07-30	09:30:41	852	131.3	14

	Global Irradiance	Diffuse Irradiance	Cirrus Cloud
Global Irradiance	1	0.040009	-0.875854
Diffuse Irradiance	0.040009	1	0.812411
Cirrus Cloud	-0.875854	0.812411	1

	Global Irradiance	Diffuse Irradiance	Cirrus Cloud
Global Irradiance	1	0.040009	-0.875854
Diffuse Irradiance	0.040009	1	0.812411
Cirrus Cloud	-0.875854	0.812411	1

	Global Irradiance	Diffuse Irradiance	Cirrus Cloud
Global Irradiance	1	0.040009	-0.875854
Diffuse Irradiance	0.040009	1	0.812411
Cirrus Cloud	-0.875854	0.812411	1

Contents

- Motivation
- State of Art
- World Model
- Modified Boids Algorithm
- Results
- **Conclusion**

Conclusion

Conclusion

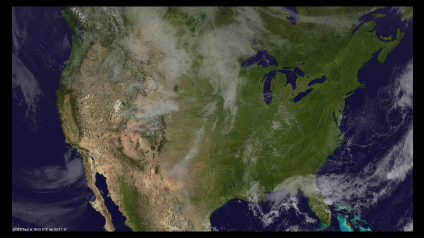
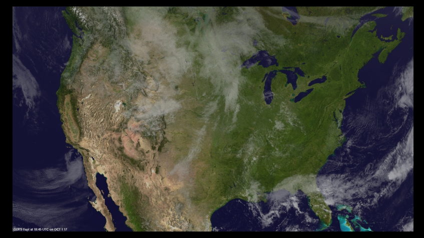
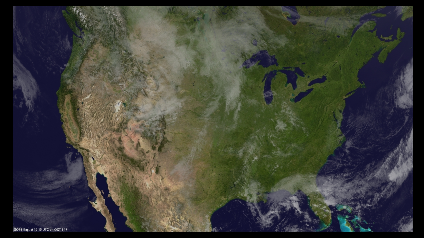
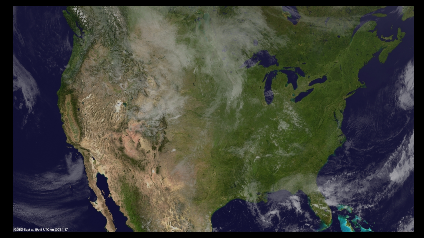
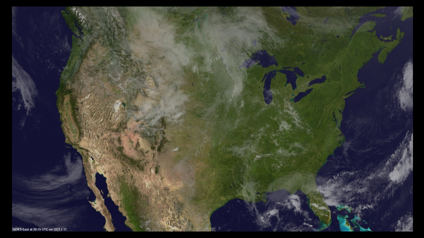
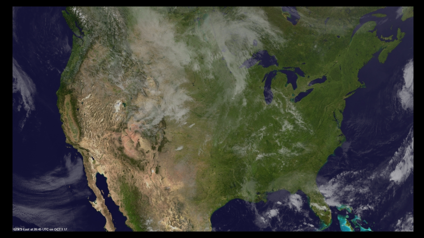
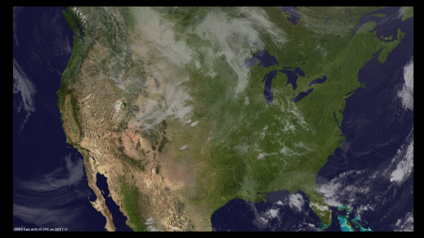
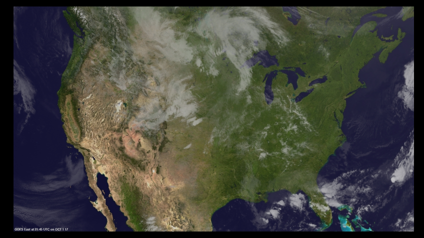
nowcasting | **short-term** | mid-term | sub-seasonal | seasonal | long-term

minutes | **hours** | days | weeks | months | years

Conclusion

nowcasting | **short-term** | mid-term | sub-seasonal | seasonal | long-term

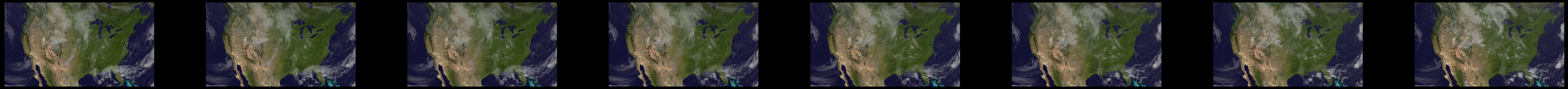
minutes | **hours** | days | weeks | months | years

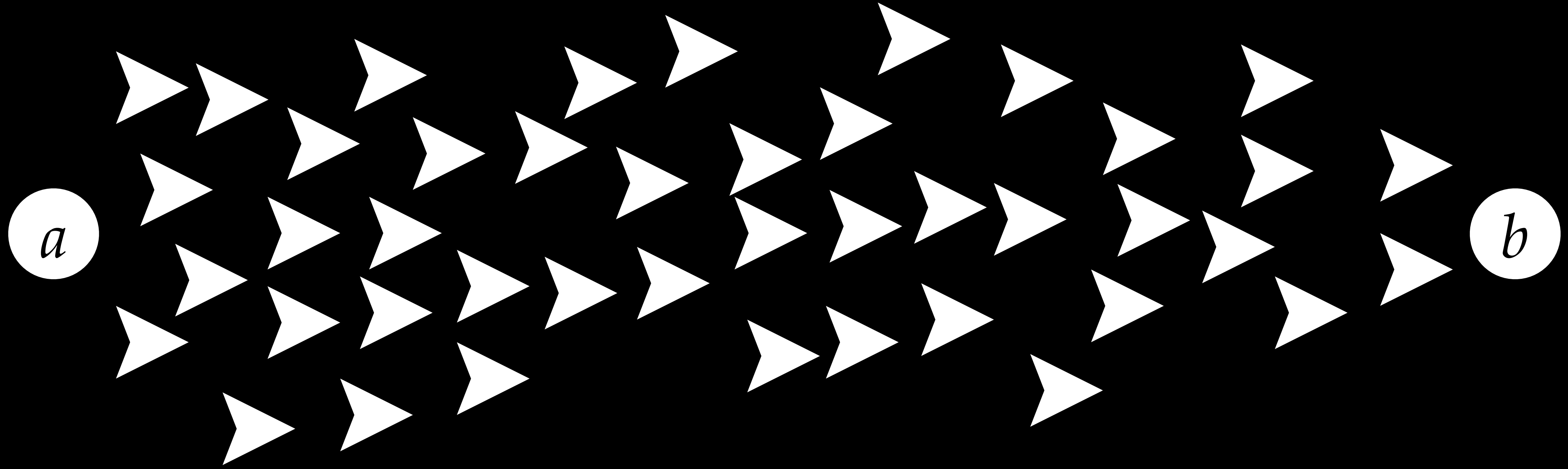









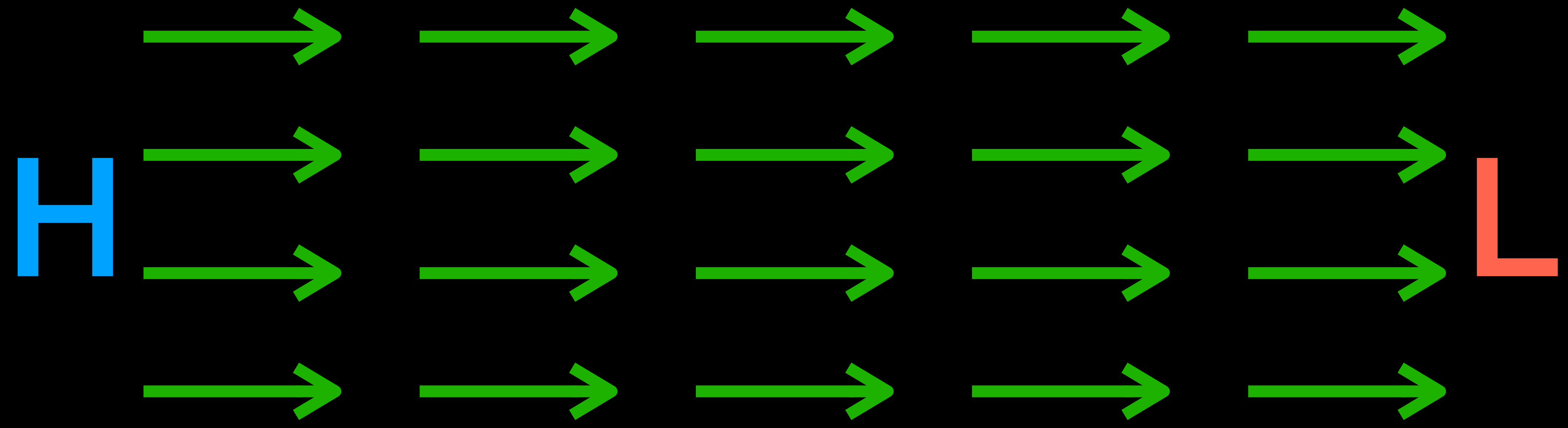
Conclusion

nowcasting | **short-term** | mid-term | sub-seasonal | seasonal | long-term

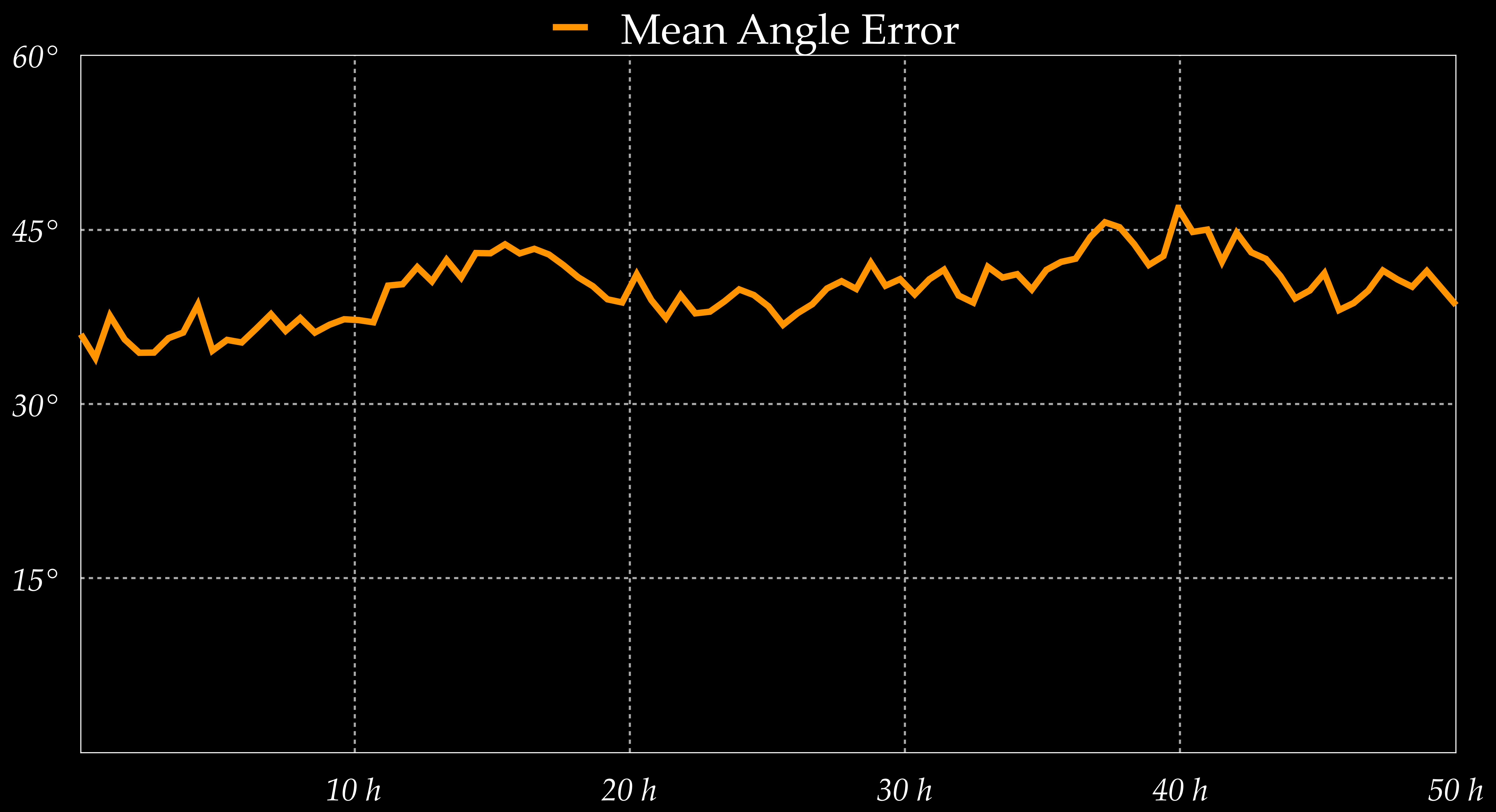
minutes | **hours** | days | weeks | months | years



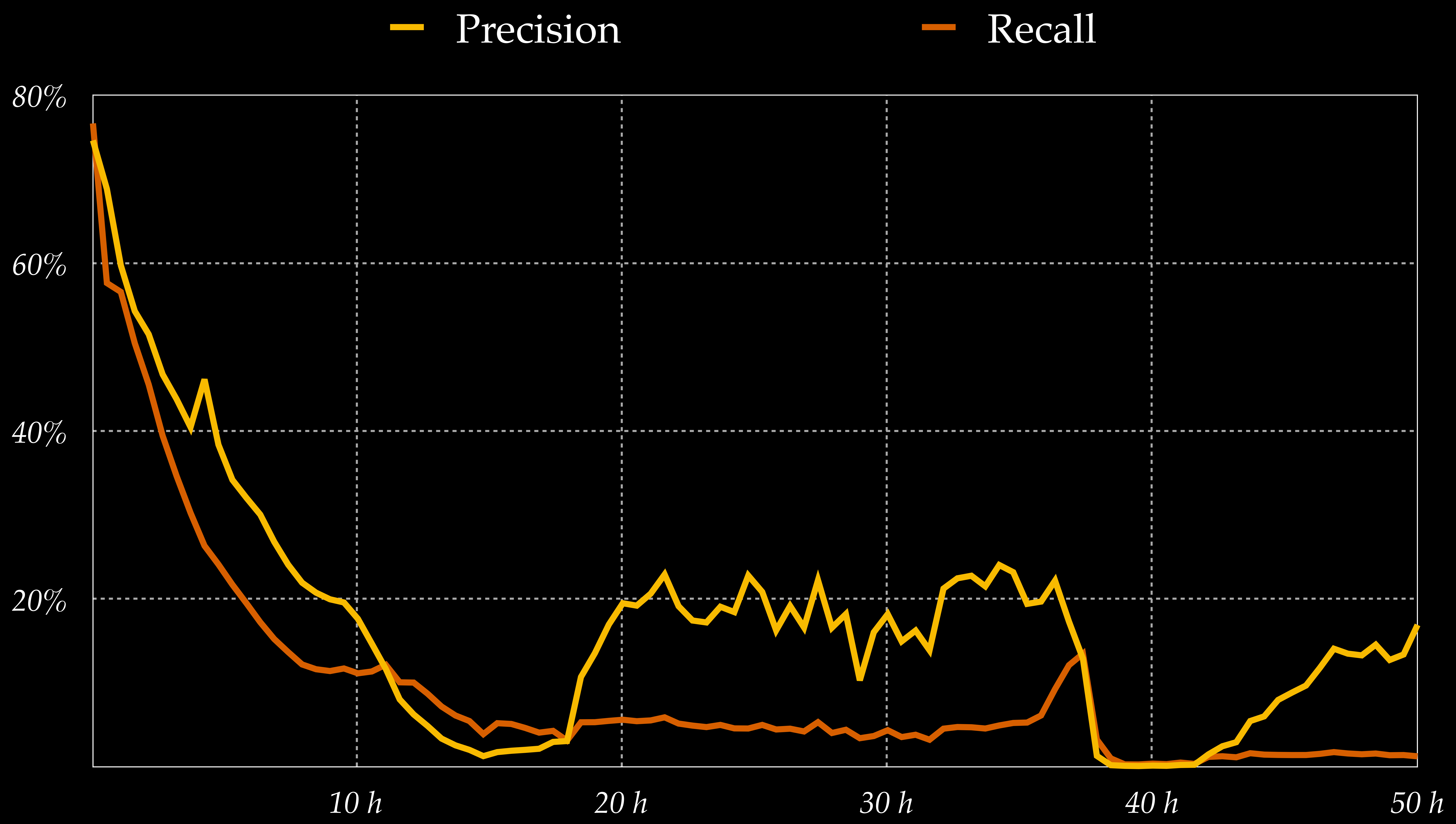




Conclusion

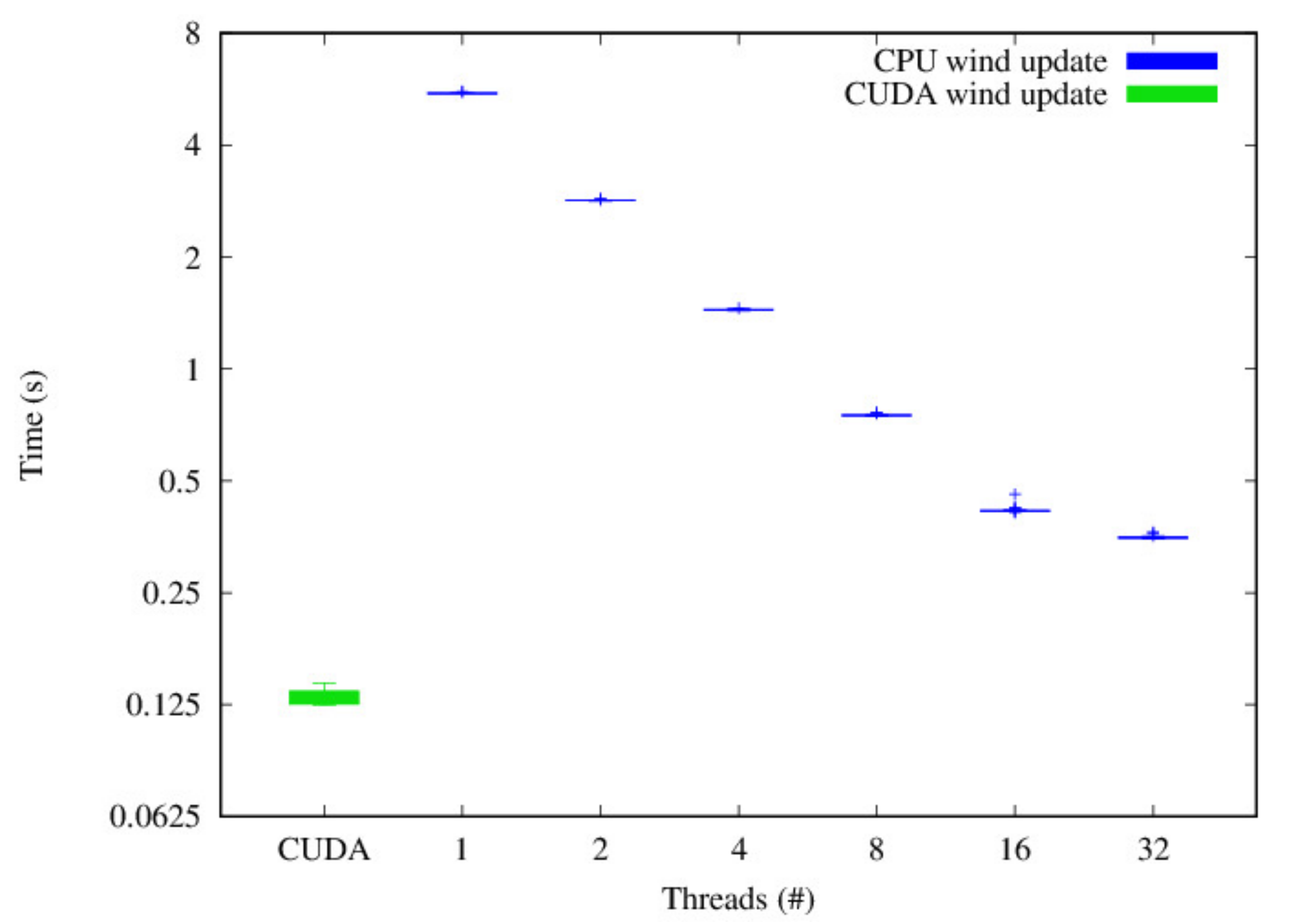


Conclusion

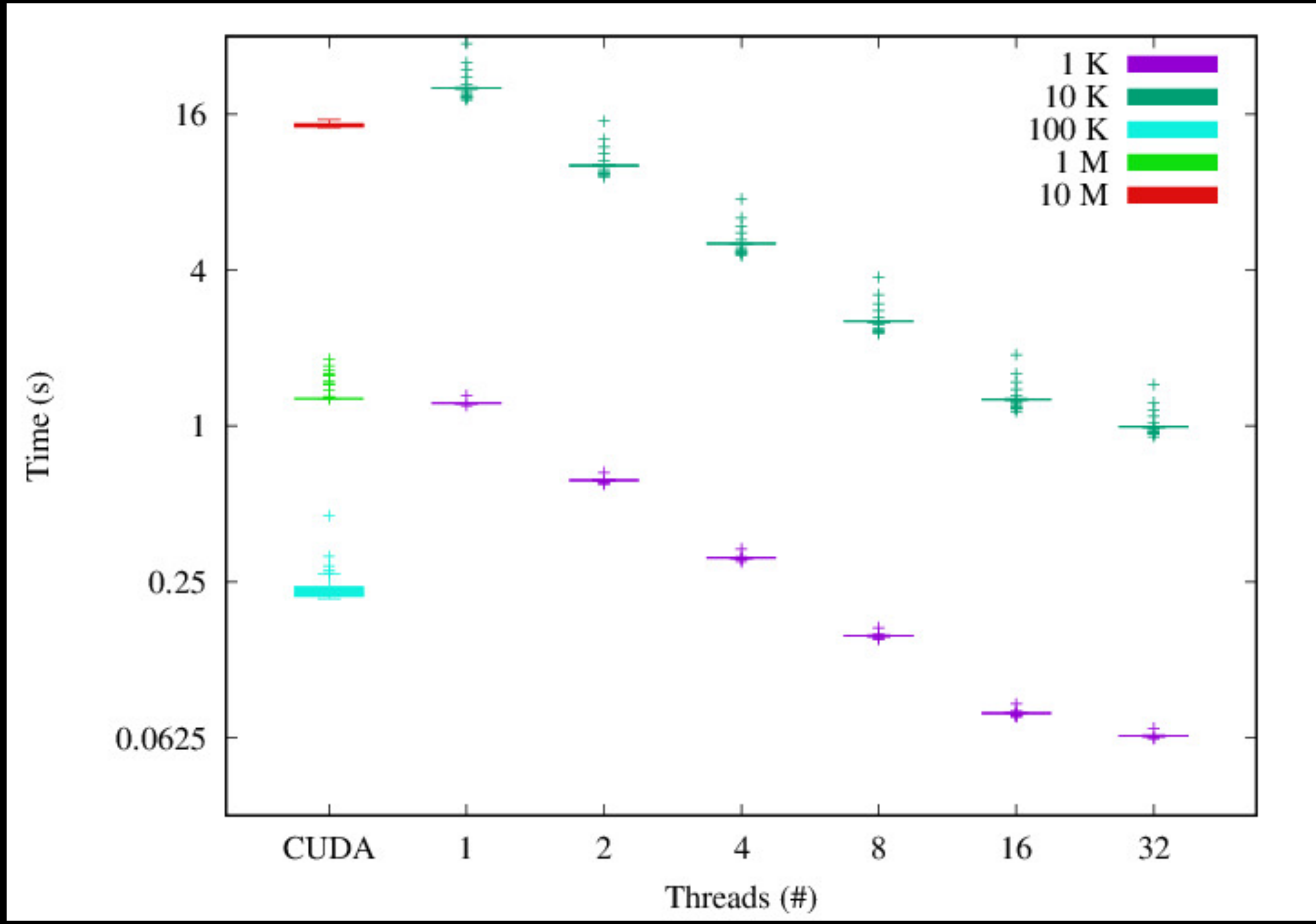


Conclusion

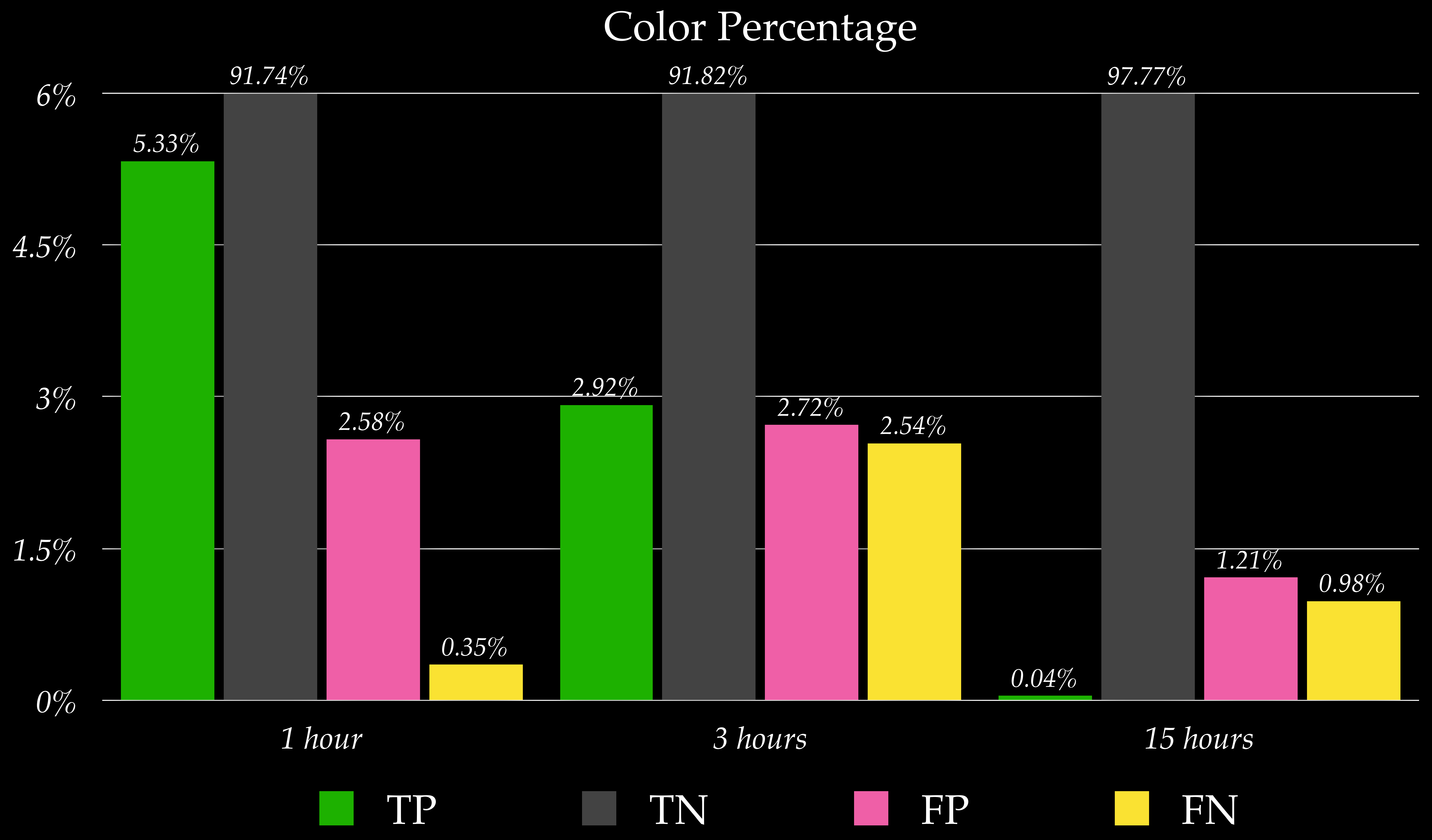
Wind map update execution time



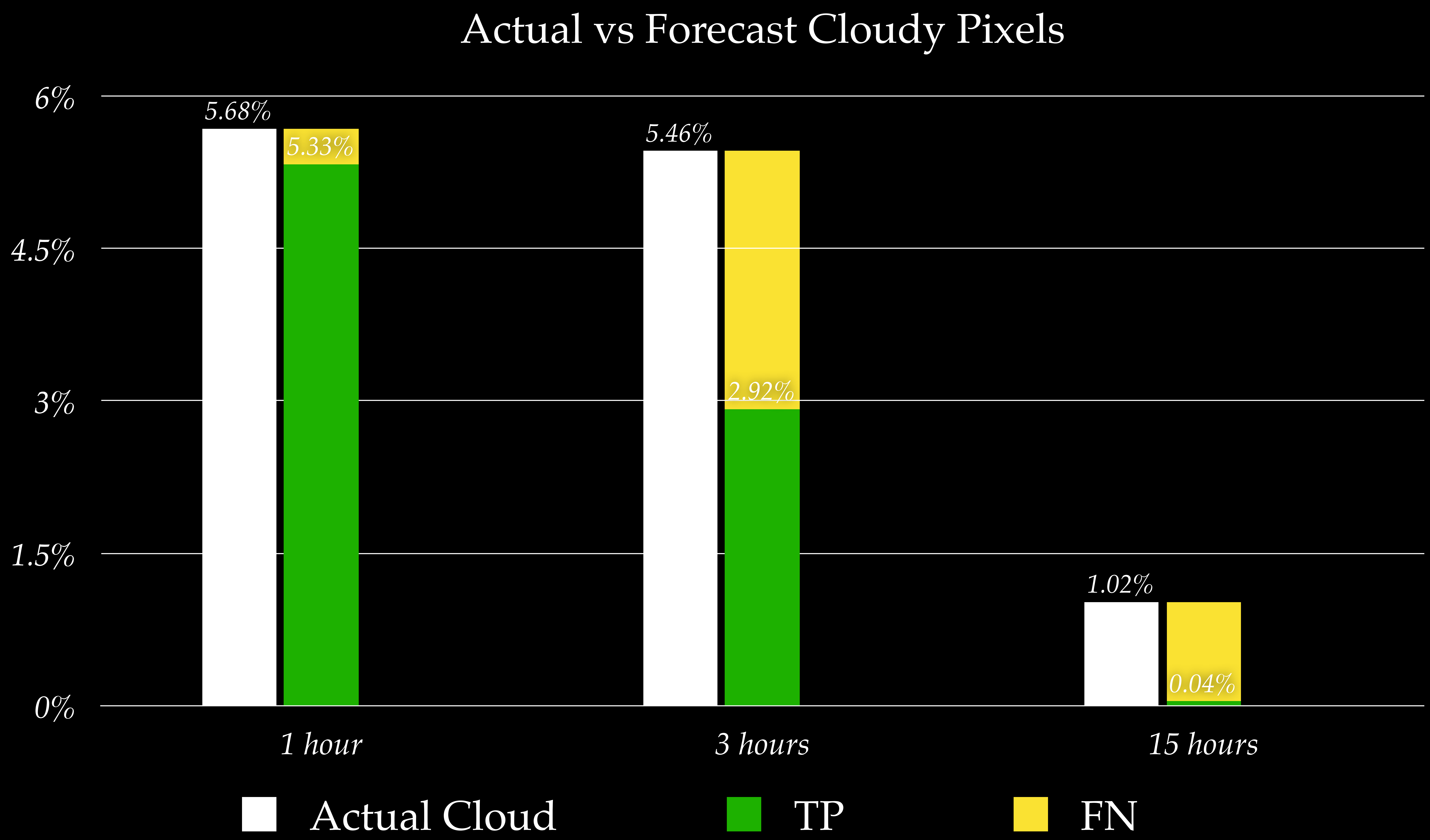
Cloud particles update execution time



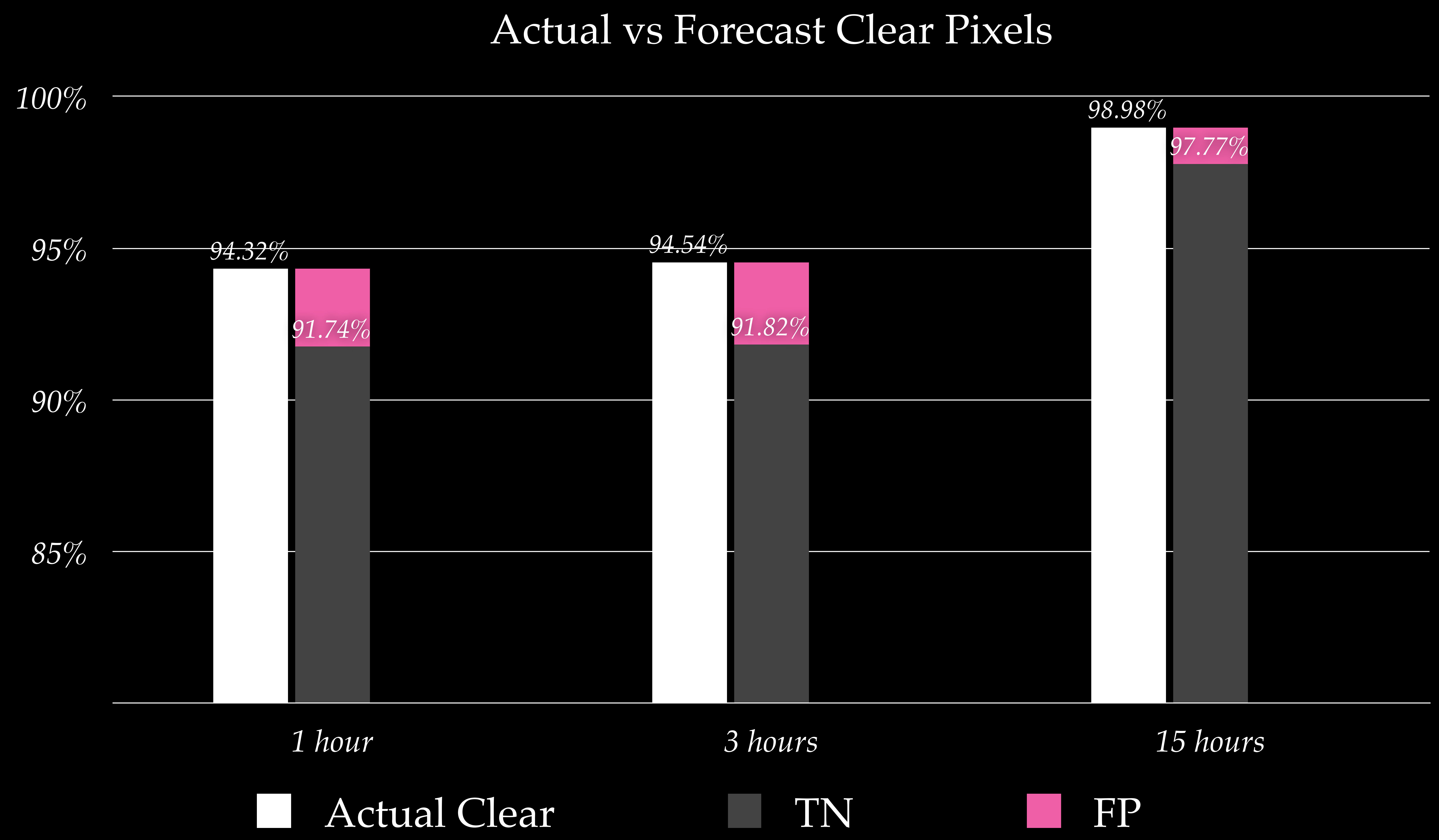
Conclusion


Color Percentage
6%
4.5%
3%
1.5%
0%
91.74%
91.82%
97.77%
5.33%
2.92%
2.58%
2.72%
2.54%
0.35%
0.04%
1.21%
0.98%
1 hour
3 hours
15 hours
TP
TN
FP
FN

Conclusion


Actual vs Forecast Cloudy Pixels
6%
4.5%
3%
1.5%
0%
5.68%
5.33%
5.46%
2.92%
1.02%
0.04%
1 hour
3 hours
15 hours
Actual Cloud
TP
FN

Conclusion



Actual vs Forecast Clear Pixels
100%
95%
90%
85%
94.32%
91.74%
94.54%
91.82%
98.98%
97.77%
1 hour
3 hours
15 hours
Actual Clear
TN
FP

Future Work

Higher resolution images;

Online forecasting service;

Applicability on sky images;

Effectiveness of auxiliary input.

Published Papers

- SYNASC 2019
- IEEE BigData 2021
- ISPDC 2021
- IEEE ISGT 2021

Thank You!

DOCTORAL THESIS

Cloud Movement Forecasting based on Wind Modeled from Satellite Imagery and a Modified Flocking Algorithm

Author:
Marius E. PENTELIUC

Advisor:
Dr. Marc E. FRÎNCU