Context for vehicles and sensing

- Environments:
  - Case 1:
    - work anywhere
      - likely: various gadgets improve safety and experience
      - implausible: full autonomy
  - Case 1.5:
    - work some places
      - some are specialized to freeways, etc.
  - Case 2:
    - work only in tightly controlled environment (eg smart city)
      - there are models of full autonomy (eg transporters at airports)

Case 1

- Various gadgets improve safety and experience
- case-by-case reasoning about representation and sensing
- Issues:
  - what's worth doing?
  - what can be done easily?
  - how much sensing?

Case 1 examples

- Reversing cameras
- Reversing sonar
- Forward sonar for collision avoidance
- Active collision management
- Pedestrian detection
- Various safety cameras
  - driver attention
  - record events for dispute resolution
  - driver sobriety
- Smarter links to maps

Case 1.5:

- Mostly, more specialized gadgets, mostly for highways
  - lane following for highways
  - predicting highway turnoffs
  - speed control that's aware of cars in front
  - neat tricks to reduce traffic jams
- Issues
  - what's worth doing?
  - what can be done easily?
  - how much sensing?

Case 2: Strongly controlled environments

- Full autonomy quite plausible
  - depending on regulatory and environmental control
    - there are models of full autonomy (eg transporters at airports)
  - This case is valuable, and may be important
    - public transport -> apartment in high density living areas
- Issues:
  - how much control do you need?
  - what density of traffic can be sustained?
  - how do you ensure safe behavior if weird stuff happens?

The questions that will plague us

- What representation do we need?
- How much data do we need to make it?
  - and where do we get it?
- How do we know if it works

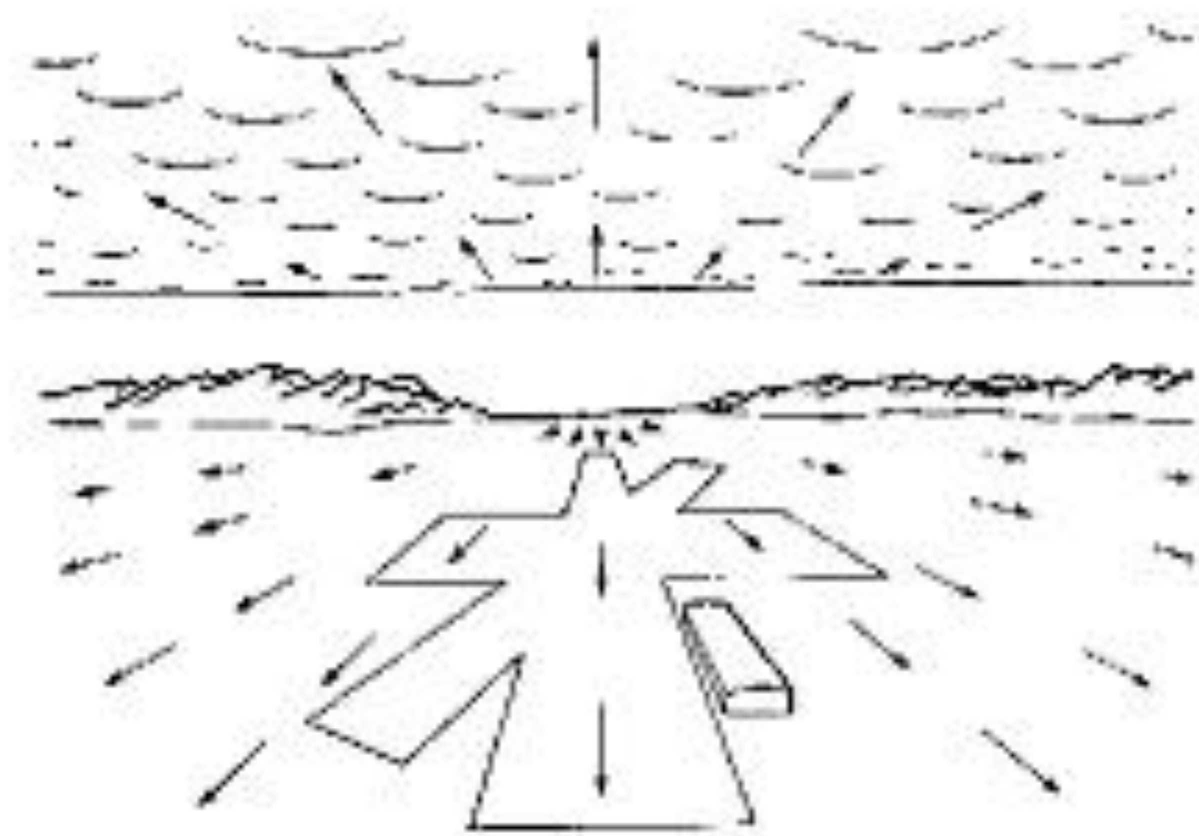
Representation

- A1: No representation required
  - link control inputs to sensing with multiple network layers
  - train on simulation with reinforcement learning
  - dubious position, but…
    - notice that, IN PRINCIPLE, this deals with full autonomy
  - Q:
    - how do you know it will do the right thing in a given situation?
  - A (dubious)
    - watch what it does on training data

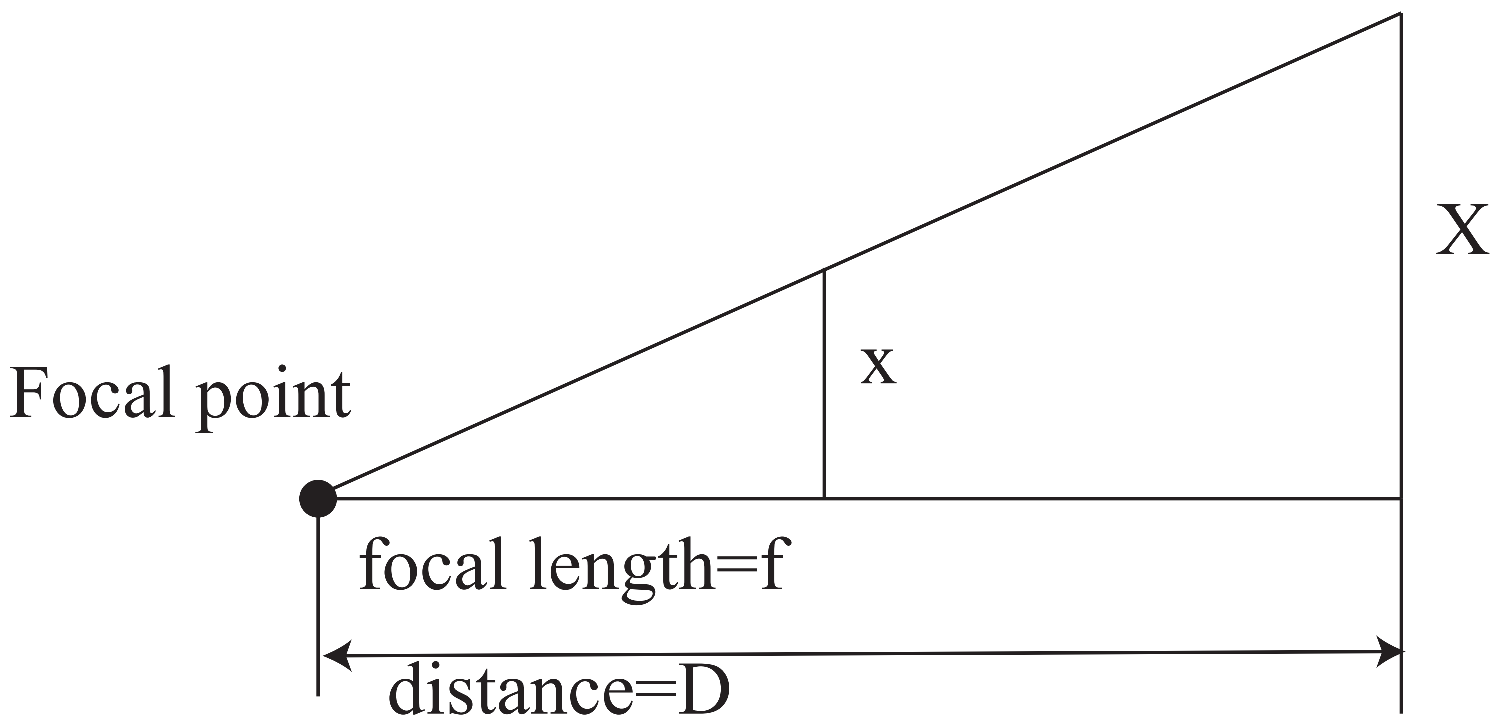
Representation

- A2: 3D reconstruction
  - build complete 3D model of world around you
    - LIDAR, SFM, etc.
    - label it with appropriate labels (next slides)
  - use a planner, etc to make paths in that environment
  - follow paths
  - Q:
    - how do you know it will do the right thing in a given situation?
  - A (dubious):
    - prove that environment is right and software is correct
  - Q:
    - do you really need a 3D representation?
  - A:
    - who knows?

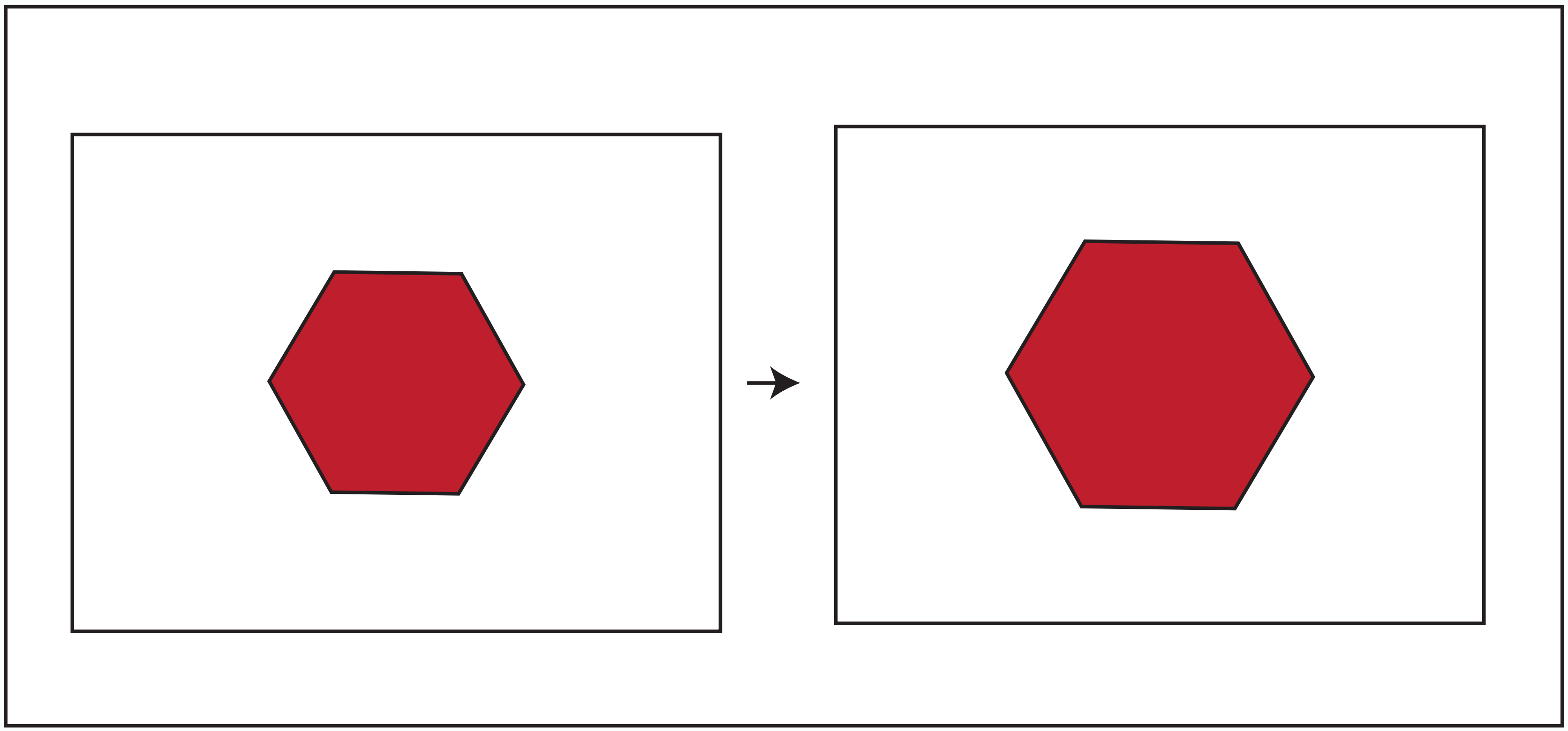
Optic flow as a theory of perception



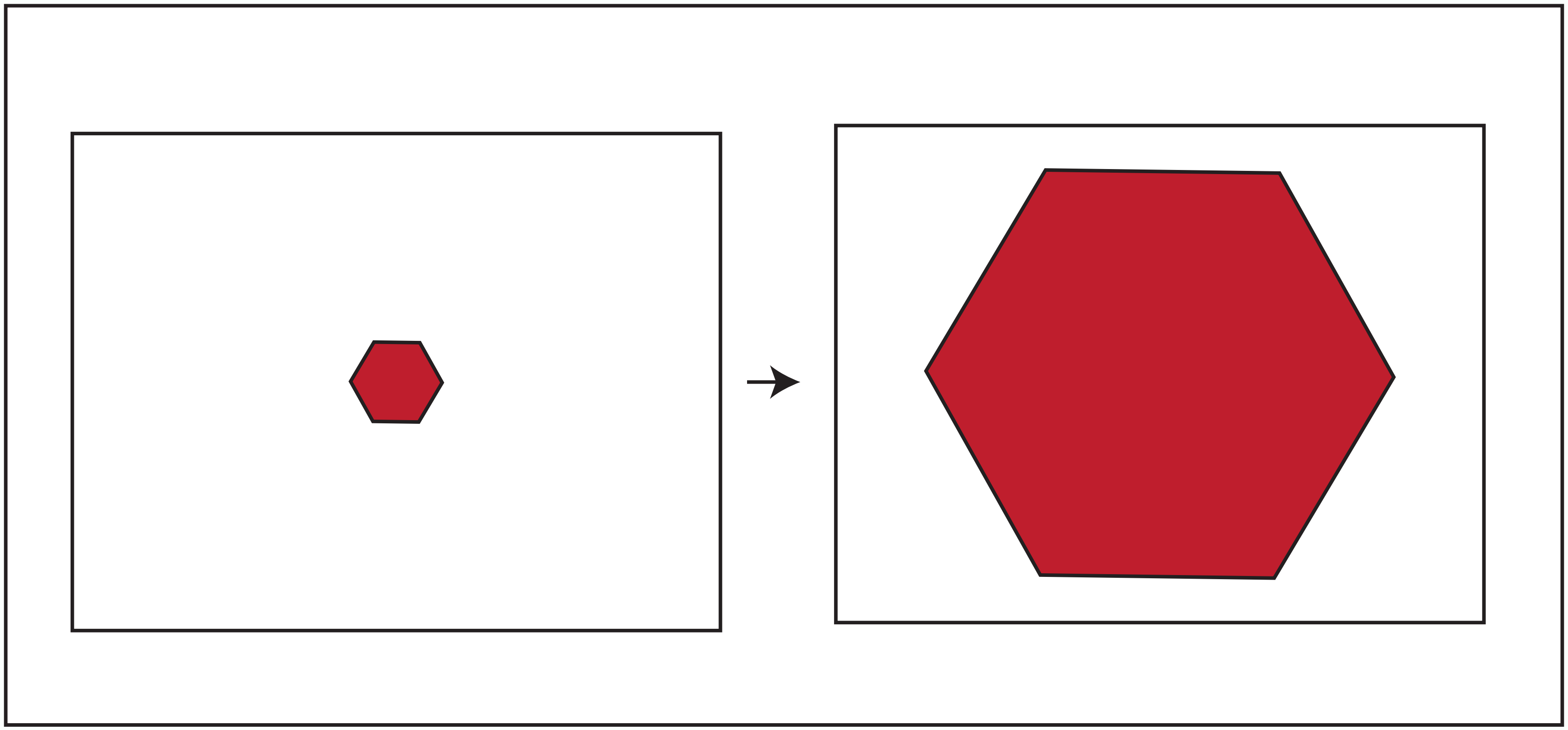
Fun fact about vision



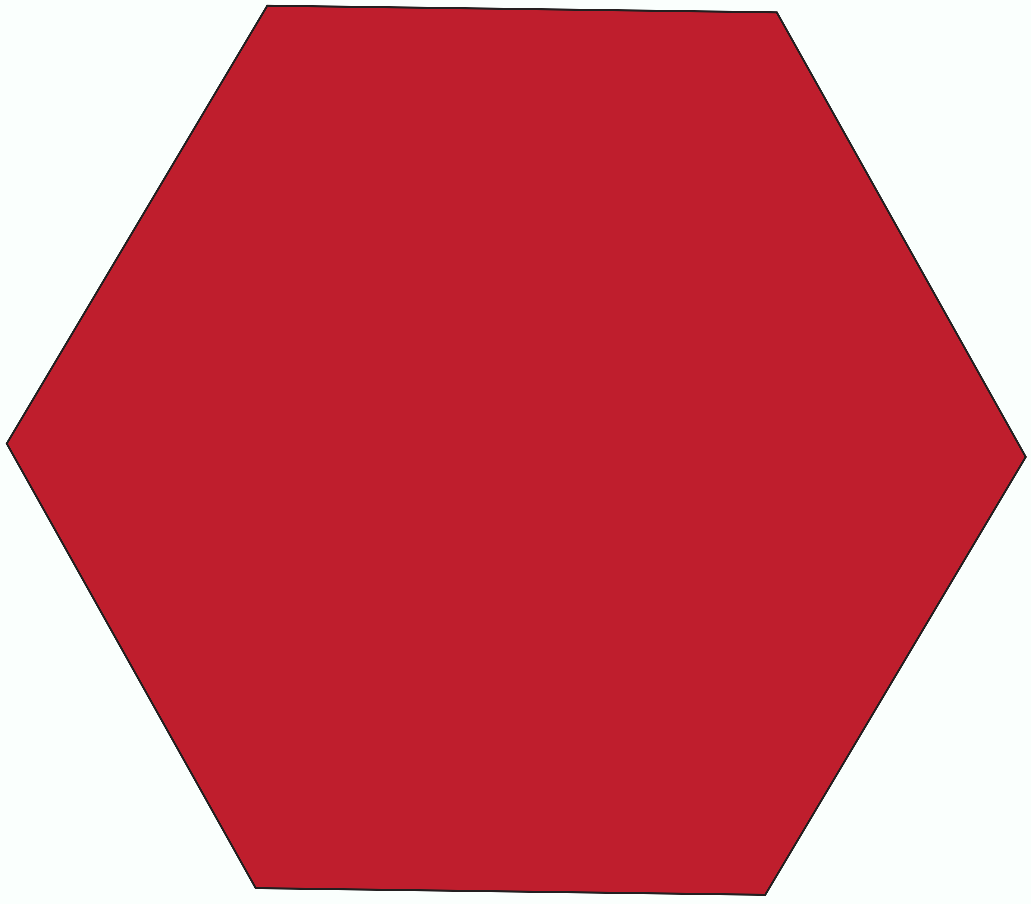
Fun fact: time to contact = x/(dx/dt)

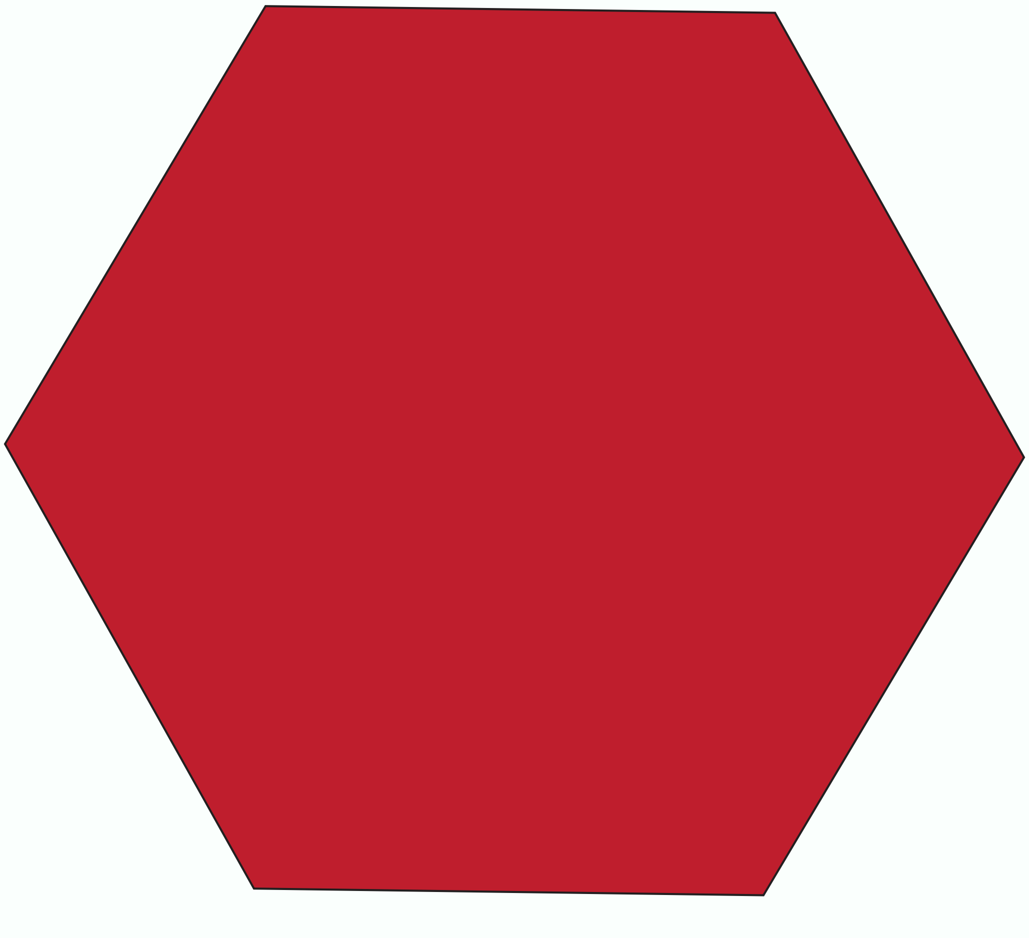


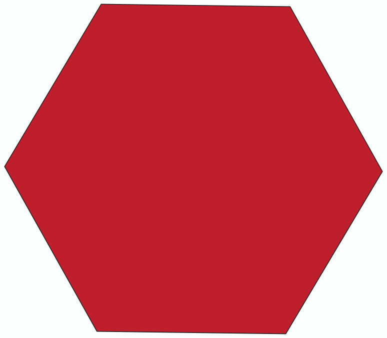
TTC - Long

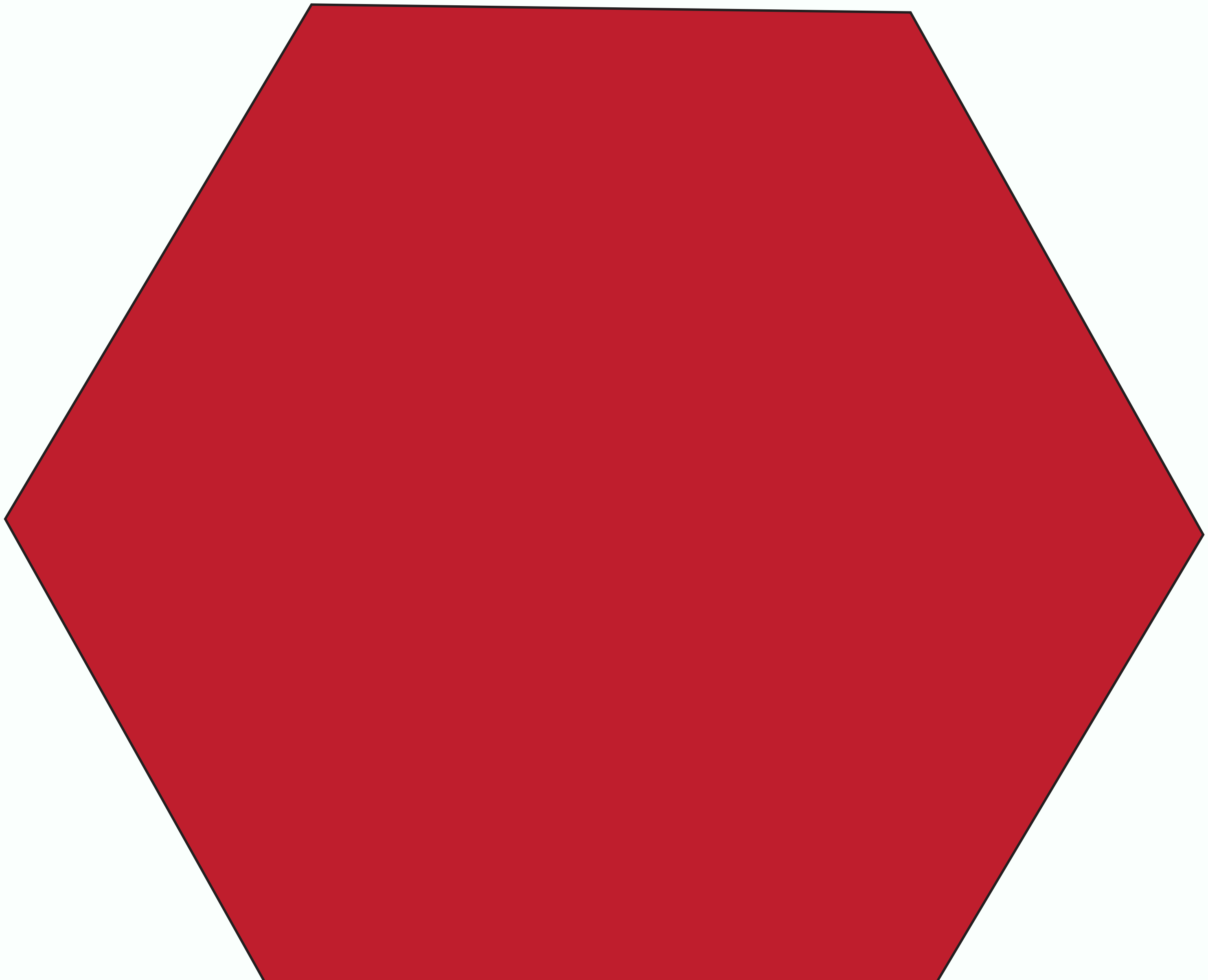


TTC - AAARGH!









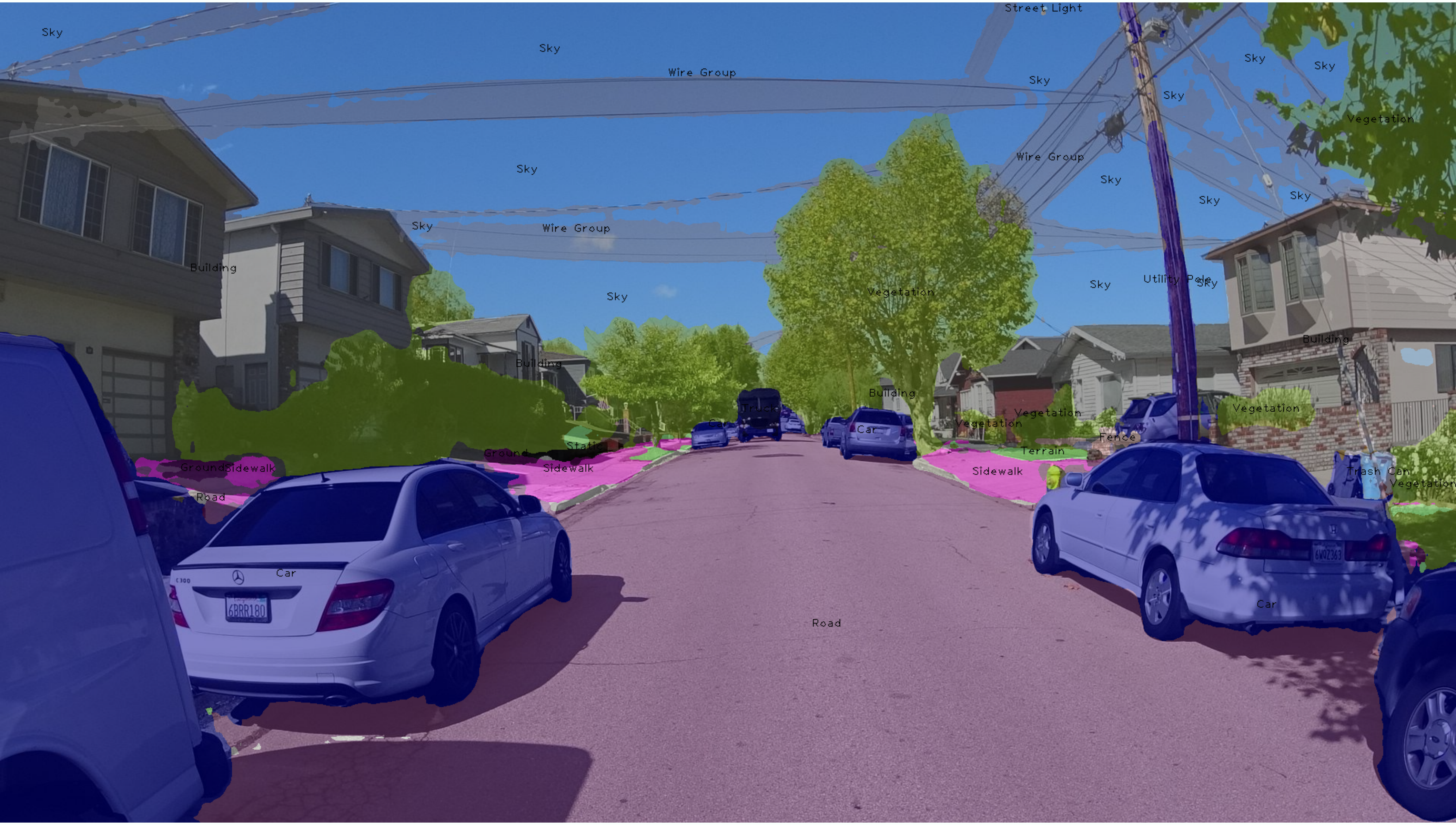
Likely truth about 3D vs 2D

- Straightforward to convert from 2D to 3D repns and back
- This means anything you can do w/3D, you can do w/2D
- But: convenience is important
  - some planners want 3D
  - sensing 3D as 3D might be a good idea (LIDAR)
  - detection is generally faster in 2D, might be easier

Representation

- A3: Label images (or 3D reconstruction)
  - with what?
    - label all possible objects with all names
    - label some classes, ignore others
  - what taxonomy?
    - likely a derived taxonomy from actions
  - Q:
    - how do you know it will do the right thing in a given situation?
  - A (dubious):
    - prove that environment is right and software is correct
  - Q:
    - what should be labelled and what should be ignored?
  - A:
    - who knows? likely the things that most affect performance?

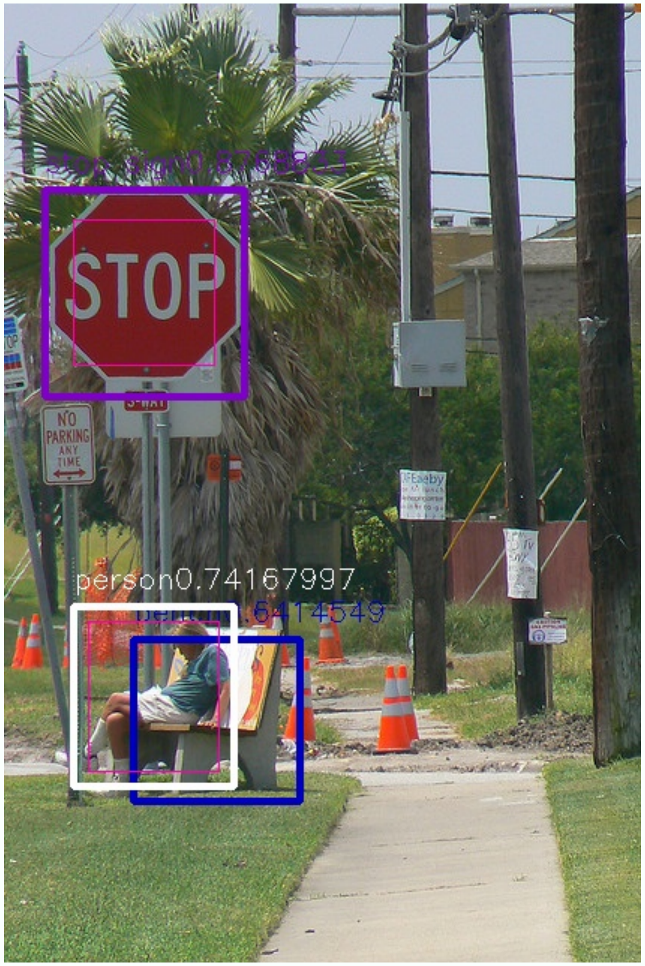
Labelling


Sky
Sky
Wire Group
Street Light
Sky
Sky
Sky
Sky
Vegetation
Wire Group
Sky
Sky
Sky
Sky
Wire Group
Building
Sky
Vegetation
Sky
Utility Pole
Building
Building
Building
Vegetation
Vegetation
Vegetation
Car
Fence
Terrain
Ground
Static
Sidewalk
Sidewalk
Ground
Sidewalk
Road
Trash Can
Vegetation
Car
Road
Car

Labelling

MS-CoCo

Labelling

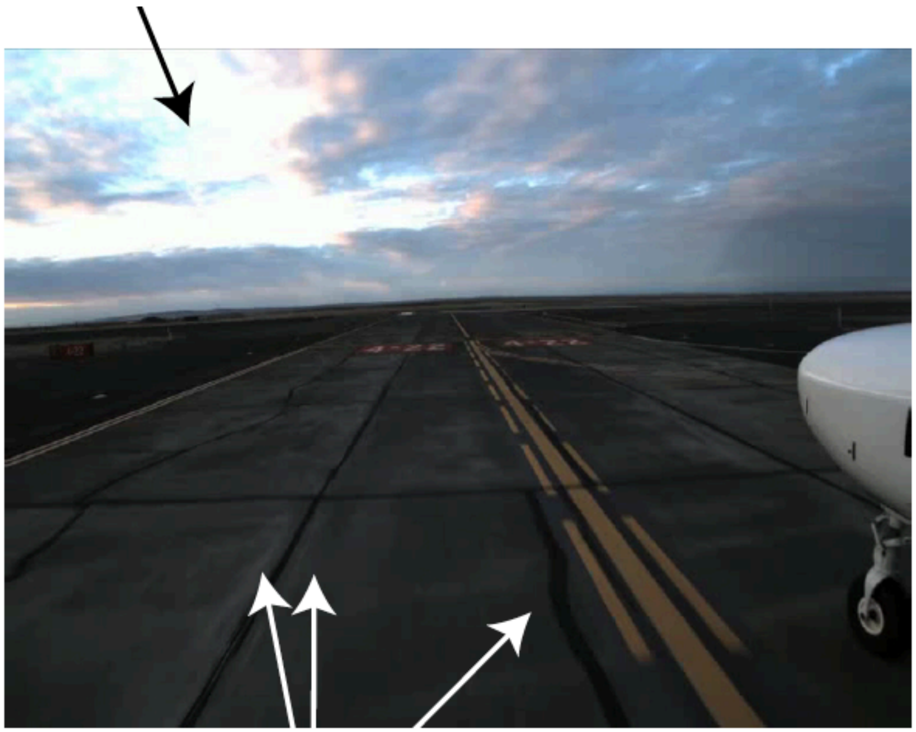


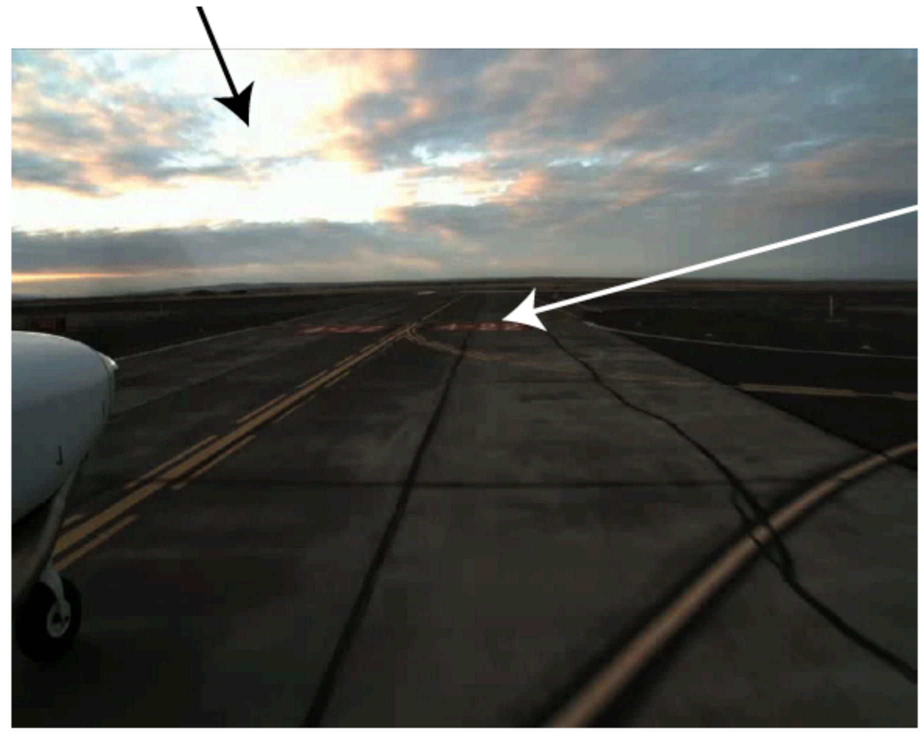
The questions that will plague us

- What representation do we need?
- How much data do we need to make it?
  - and where do we get it?
- How do we know if it works?

XXXX Autonomy data

Special features: rich appearance variation

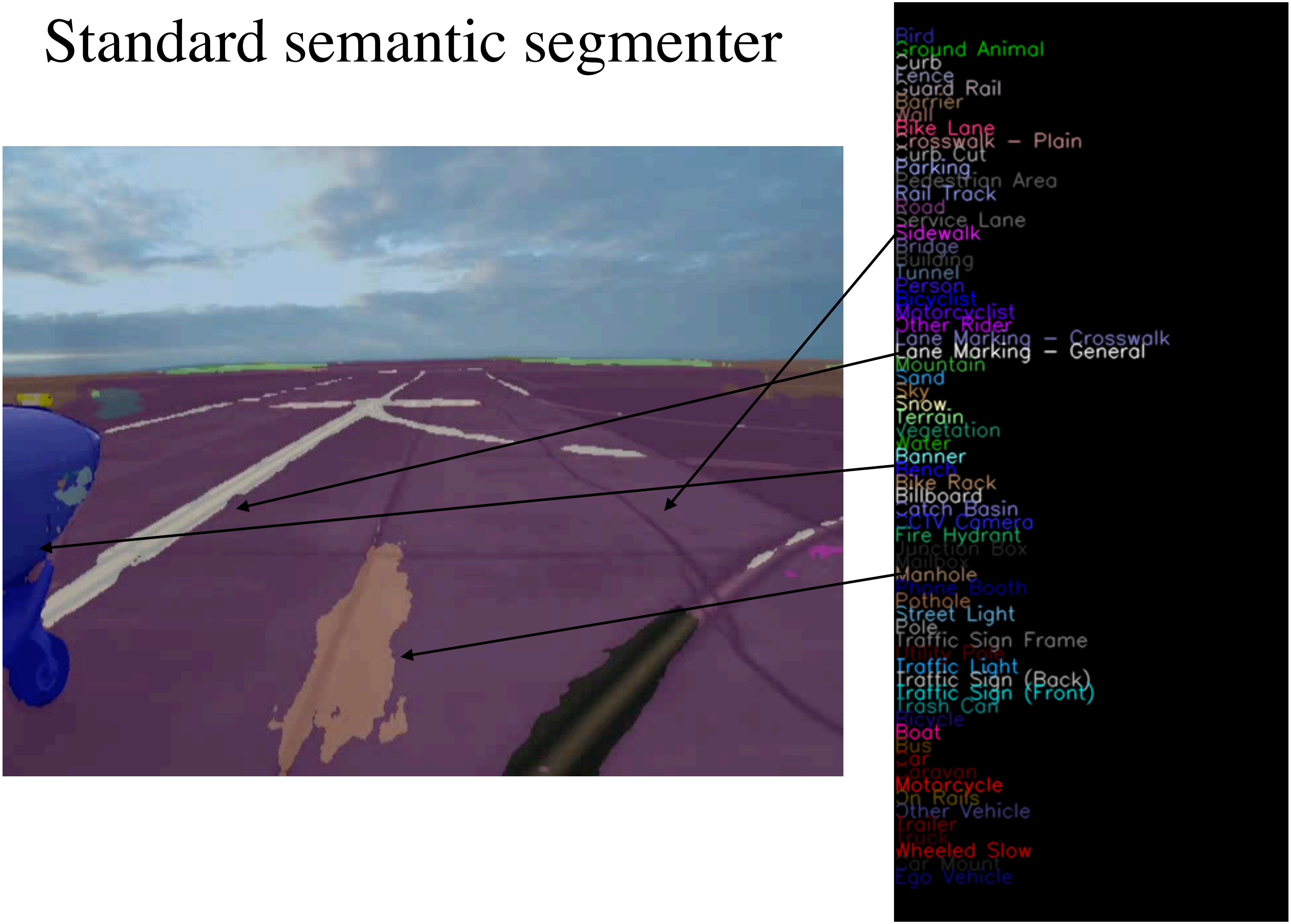




Special features: rich appearance variation

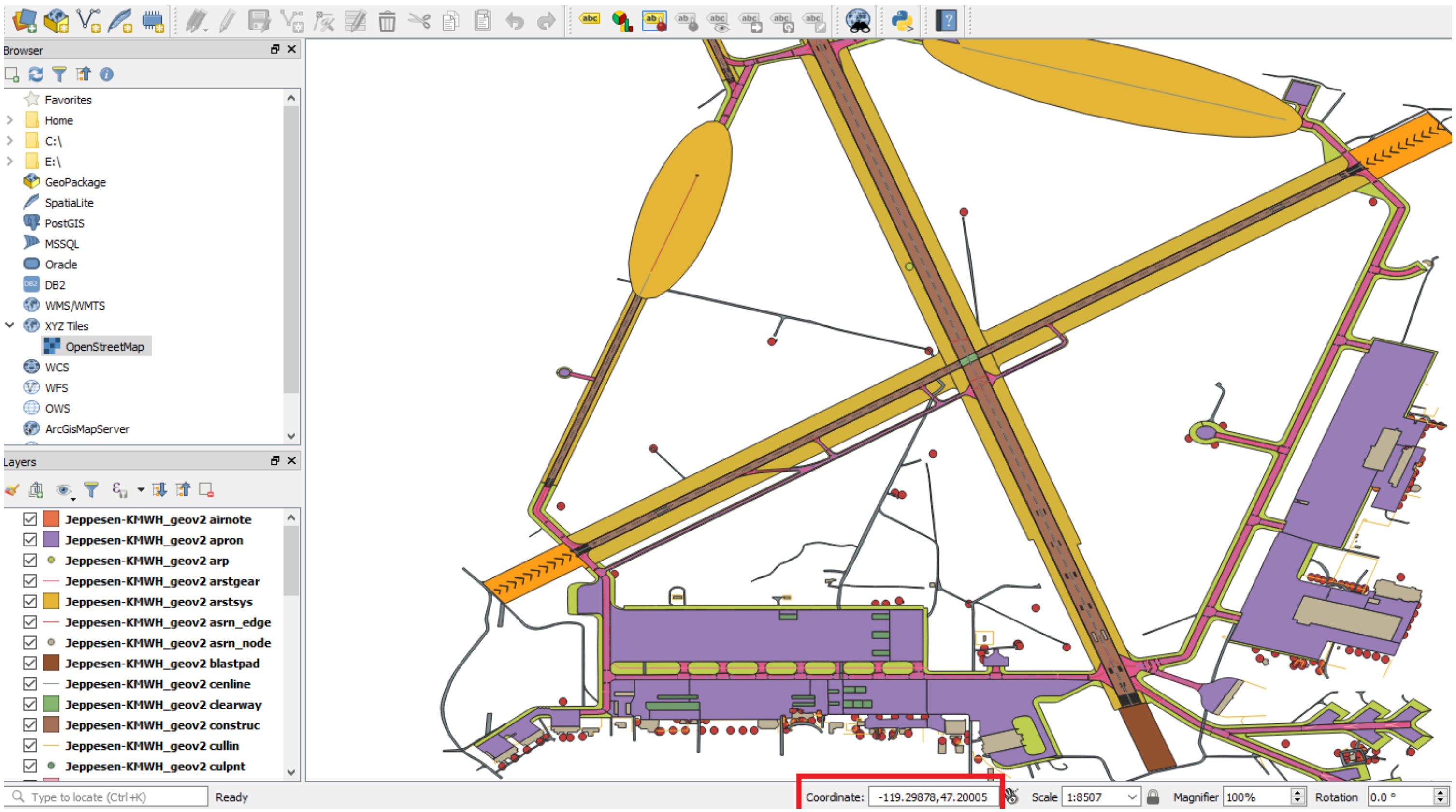
Special features: rich appearance variation

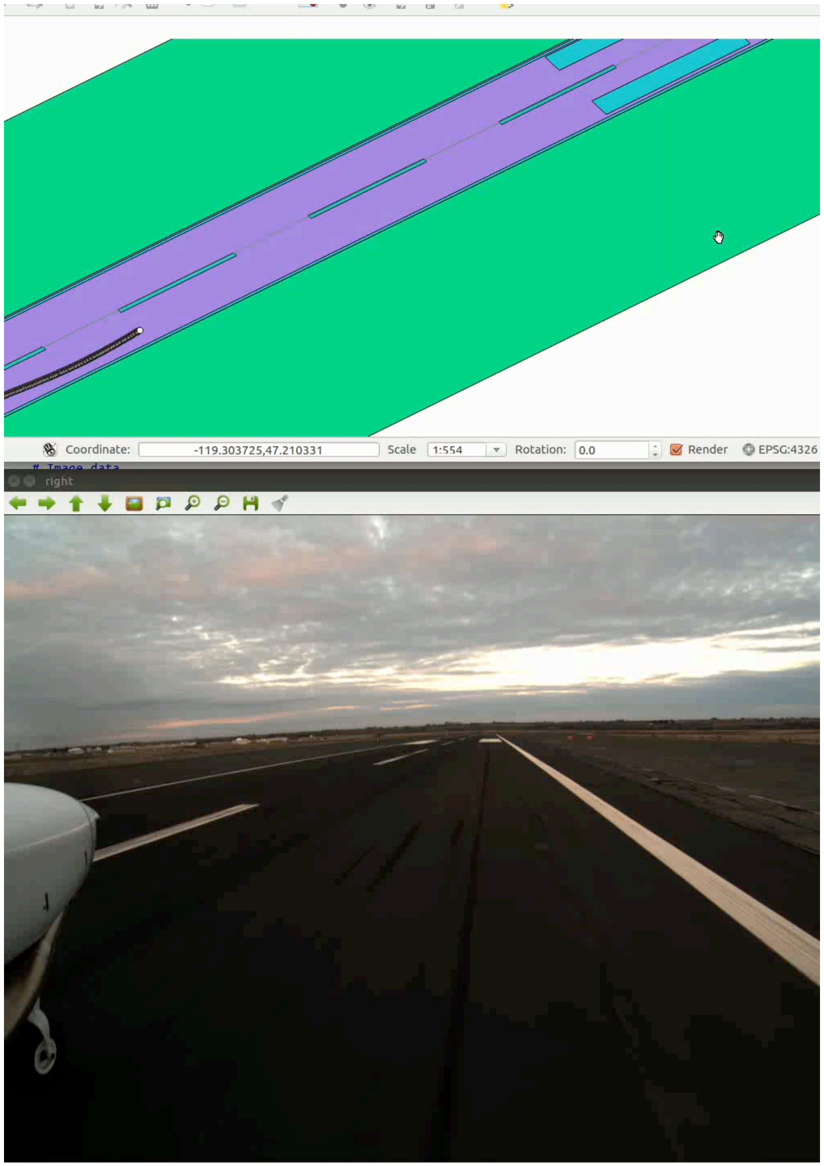
Standard semantic segmenter



XXX data consequences

XXXX data consequences


Browser
Favorites
Home
C:\
E:\
GeoPackage
SpatiaLite
PostGIS
MSSQL
Oracle
DB2
WMS/WMTS
XYZ Tiles
OpenStreetMap
WCS
WFS
OWS
ArcGisMapServer
Layers
Jeppesen-KMWH_geov2 airnote
Jeppesen-KMWH_geov2 apron
Jeppesen-KMWH_geov2 arp
Jeppesen-KMWH_geov2 arstgear
Jeppesen-KMWH_geov2 arstsys
Jeppesen-KMWH_geov2 asrn_edge
Jeppesen-KMWH_geov2 asrn_node
Jeppesen-KMWH_geov2 blastpad
Jeppesen-KMWH_geov2 cenline
Jeppesen-KMWH_geov2 clearway
Jeppesen-KMWH_geov2 construc
Jeppesen-KMWH_geov2 cullin
Jeppesen-KMWH_geov2 culpnt
Type to locate (Ctrl+K)
Ready
Coordinate: -119.29878,47.20005
Scale 1:8507
Magnifier 100%
Rotation 0.0 °



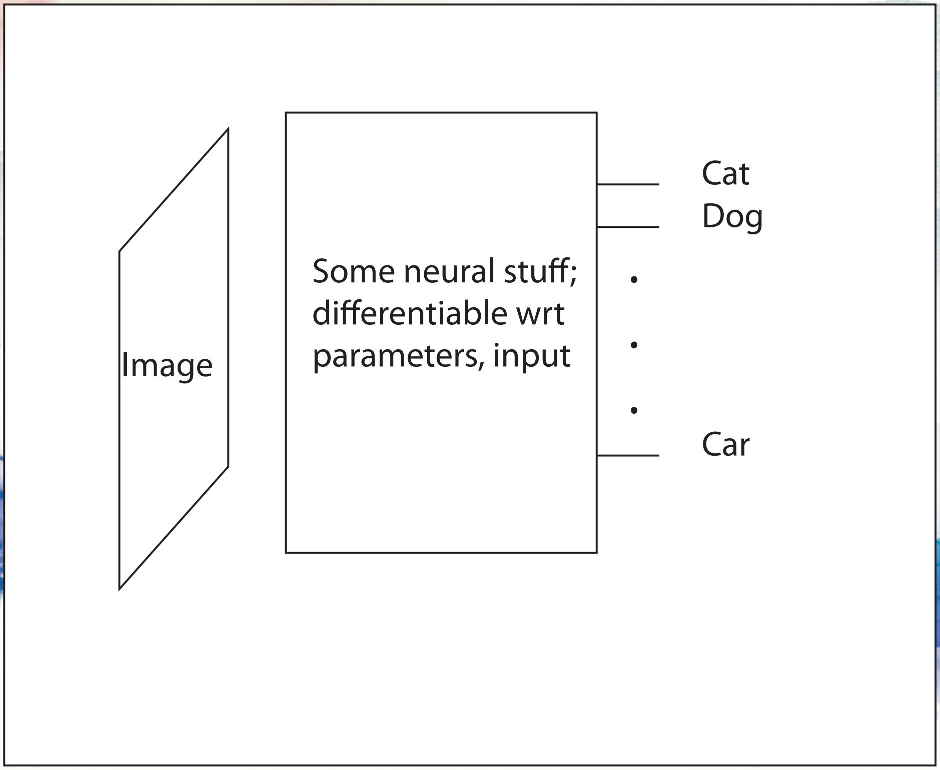
GPS is quite good, but not perfect

Another curious data problem

The questions that will plague us

- What representation do we need?
- How much data do we need to make it?
  - and where do we get it?
- How do we know if it works?

Image classification



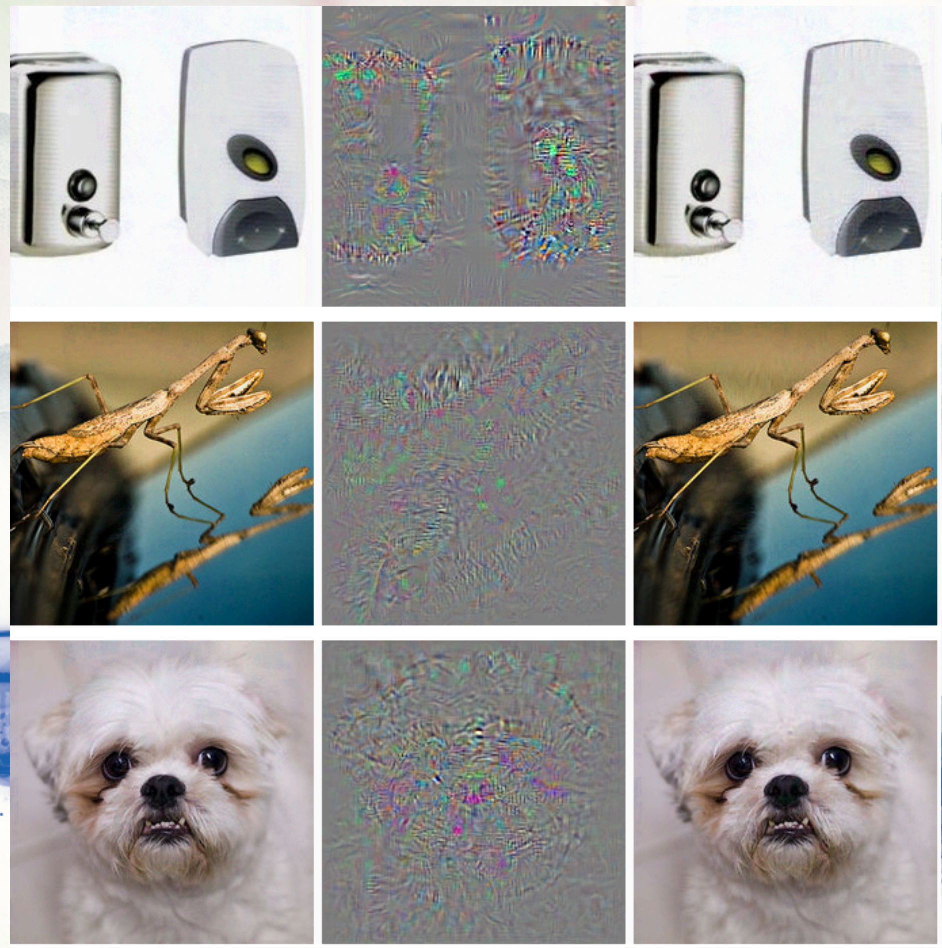
APOLLOSCAPE

Adversarial example

- Search for
    - small update to image
    - such that
        - output for true class is low
        - output for some other class is high

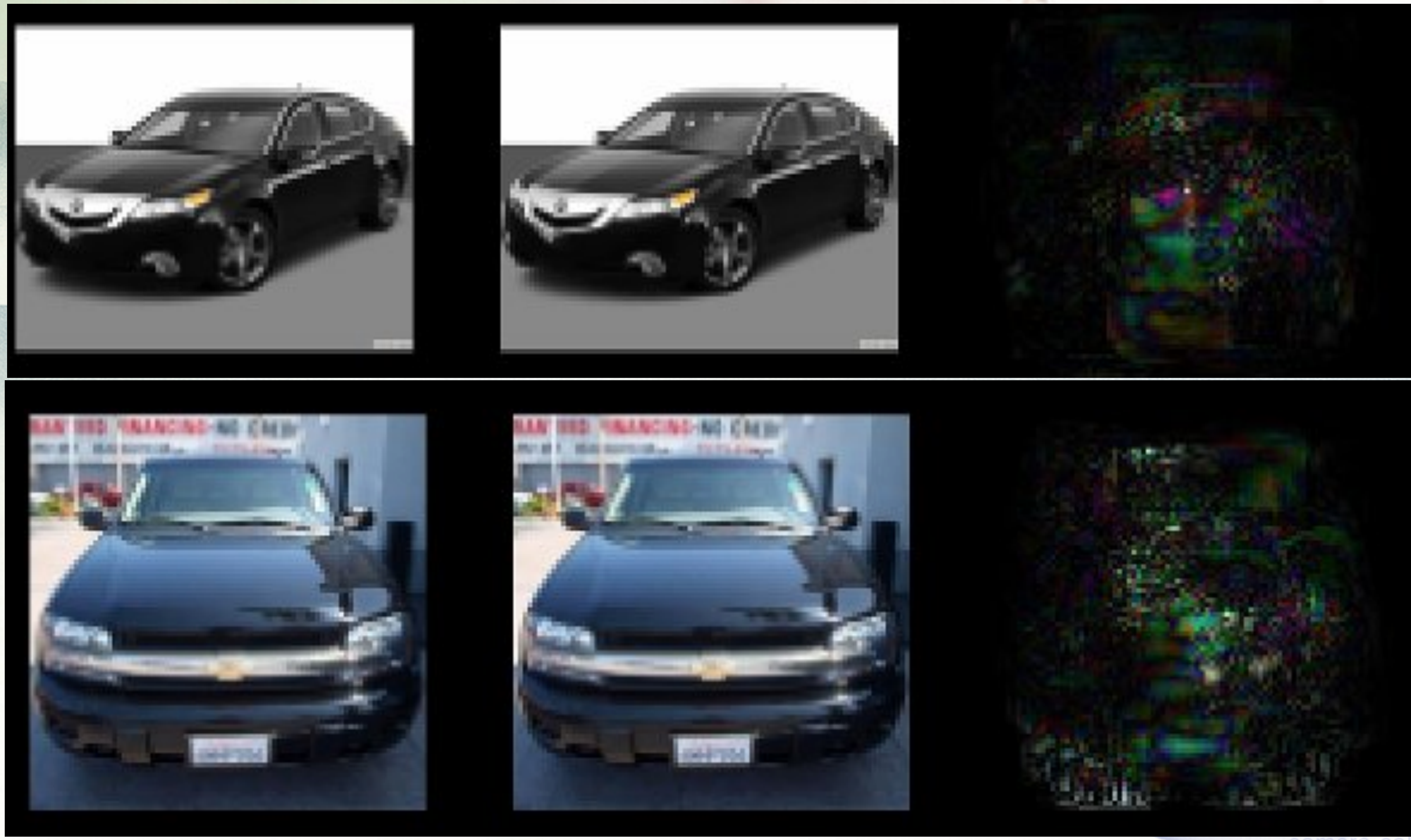
- Surprising fact:
    - such updates can be VERY small

Correctly classified



“Ostrich”

Szegedy et al, 13



Szegedy et al, 13

Fast gradient sign search

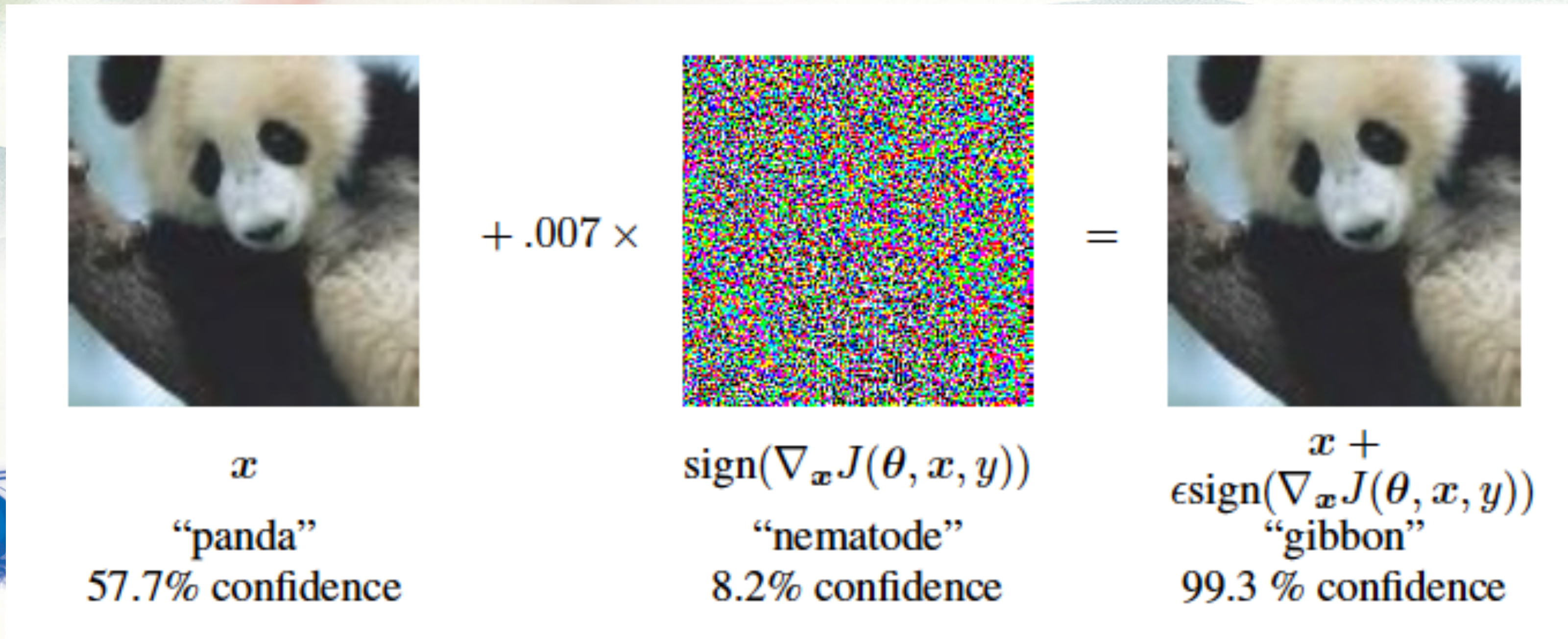
- Search sign(gradient)
  - x is image
  - J is some cost
    - eg J=(true class)-(best false class)
- Iterate

$$\eta = \epsilon \text{sign}\left(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y)\right).$$

$$\boldsymbol{X}_{adv} = \boldsymbol{X} - \epsilon \cdot \text{sign}(\nabla_{\boldsymbol{X}} J(\boldsymbol{X}, y_{fool})).$$

Goodfellow et al 15

Fast gradient sign



Deepfool

APOLLOSCAPE

- Find r by
  - linearize k
    - update
    - (possibly) repeat

update

$$\Delta(x;\hat{k}) := \min_{r} \|r\|_2 \text{ subject to } \hat{k}(x+r) \neq \hat{k}(x)$$

label

image

Moosavi-Desfooli et al 16

Deepfool

Moosavi-Desfooli et al 16

Compare fast gradient sign

Turtle

Difference image

Moosavi-Desfooli et al 16

Flow based methods

- New image obtained by

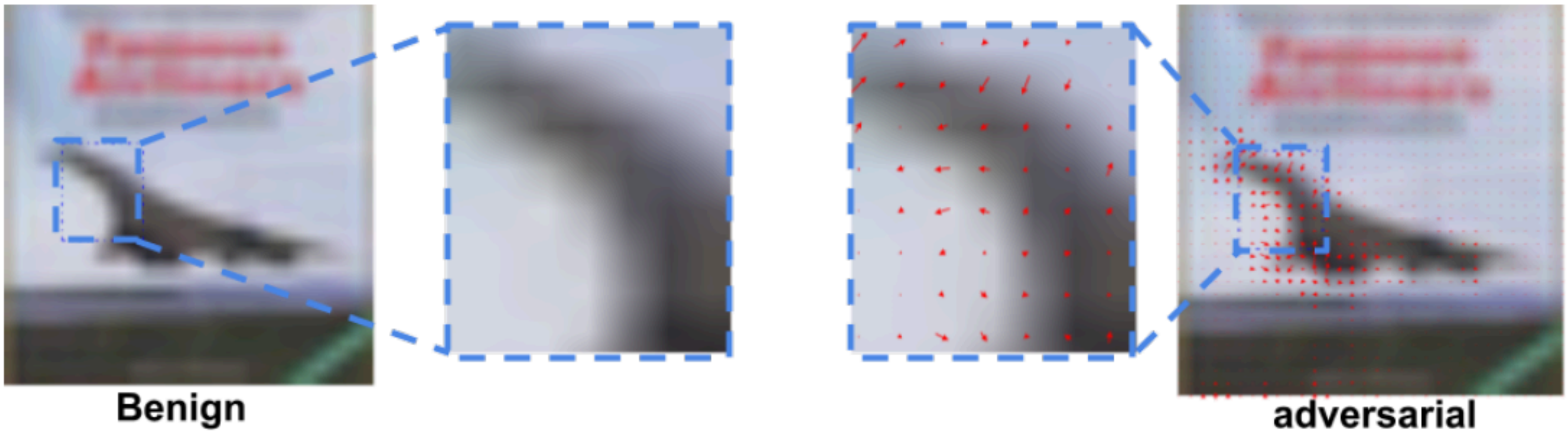


Figure 6: Flow visualization on CIFAR-10. The example is misclassified as bird.

Xiao et al
2018

Flow based methods

- New image obtained by moving pixels

$$\mathcal{X}_{adv}(u, v) = \mathcal{X}(u + f_u, v + f_v)$$

Size of flow

$$f^* = \underset{f}{\operatorname{argmin}} \quad \mathcal{L}_{adv}(x, f) + \tau \mathcal{L}_{flow}(f)$$

Adversarial loss

Xiao et al 2018

Flow methods

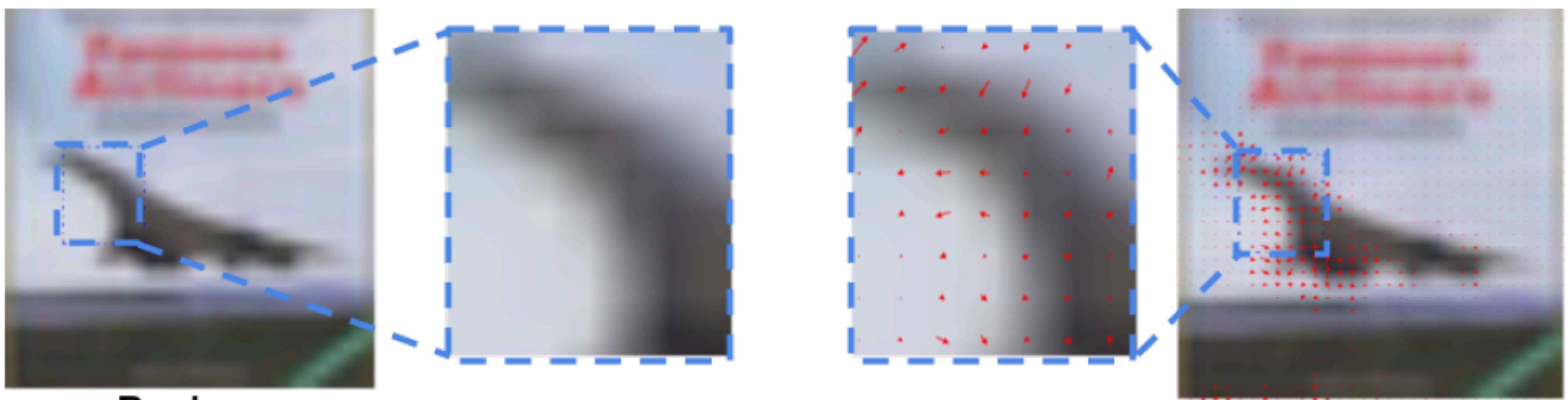
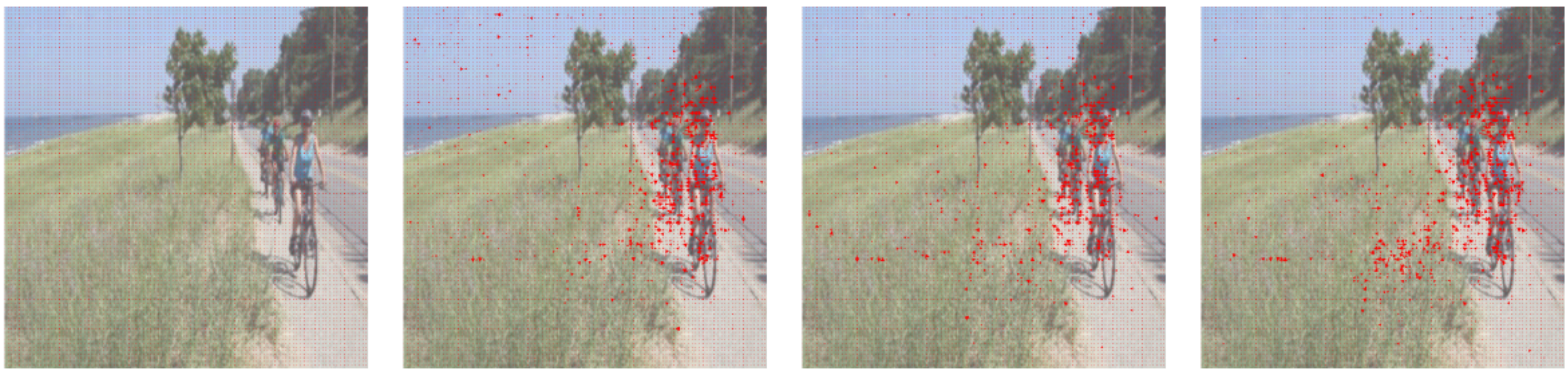


Figure 6: Flow visualization on CIFAR-10. The example is misclassified as bird.

Xiao et al 2018

Flow methods



(a) mountain bike (b) goldfish (c) Maltese dog (d) tabby cat

Xiao et al 2018

Lu et al 18

Yolo attack

- Yolo uses a large image area to
  - predict boxes
  - predict classes
- This means that a detection is
  - affected by pixels OUTSIDE box


stop sign0.8768833
STOP
3-WAY
NO PARKING ANY TIME
person0.74167997


Computer vision technology
advanced driver assistance
systems
Visual driver
monitoring
driver-vehicle
interfaces
Autonomous navigation and exploration


STOP
3-WAY
NO PARKING ANY TIME


motorbike0.65933144
stop sign0.83716106
STOP
HOOTERS
Casino Hotel
TAXI STAND
STAR CAB


Computer vision technology
Vision-based
advanced driver assistance
systems
Visual driver
navigation and exploration
monitoring


stop sign0.8481956
STOP
HOOTERS
Casino Hotel
TAXI STAND
STAR CAB