COM2009-3009
Robotics

Lecture 9

Localisation

Dr Tom Howard

Multidisciplinary Engineering Education (MEE)

Last time

Local Guidance Strategies: **Where am I going?**

- Search
- Beaconing
- Visual Homing

Guidance:
The directing of the motion or position of something, especially an aircraft, spacecraft, or missile.

Today

Localisation: **Where am I?**

Navigation:
The process or activity of accurately ascertaining one's position and planning and following a route.

This lecture

1. External positioning systems

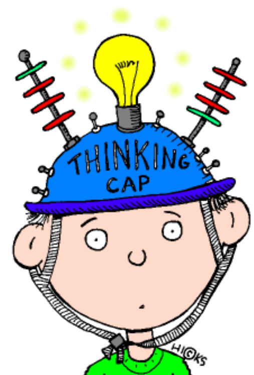
2. On-board positioning

   – **Classic** Odometry

   – **Visual** Odometry

   (a.k.a: Dead-reckoning, Path Integration)

Name an **external** positioning system that a robot might use to navigate:



Name an **external** positioning system that a robot might use to navigate:

$\boxed{\text{GPS}}$

Sonar

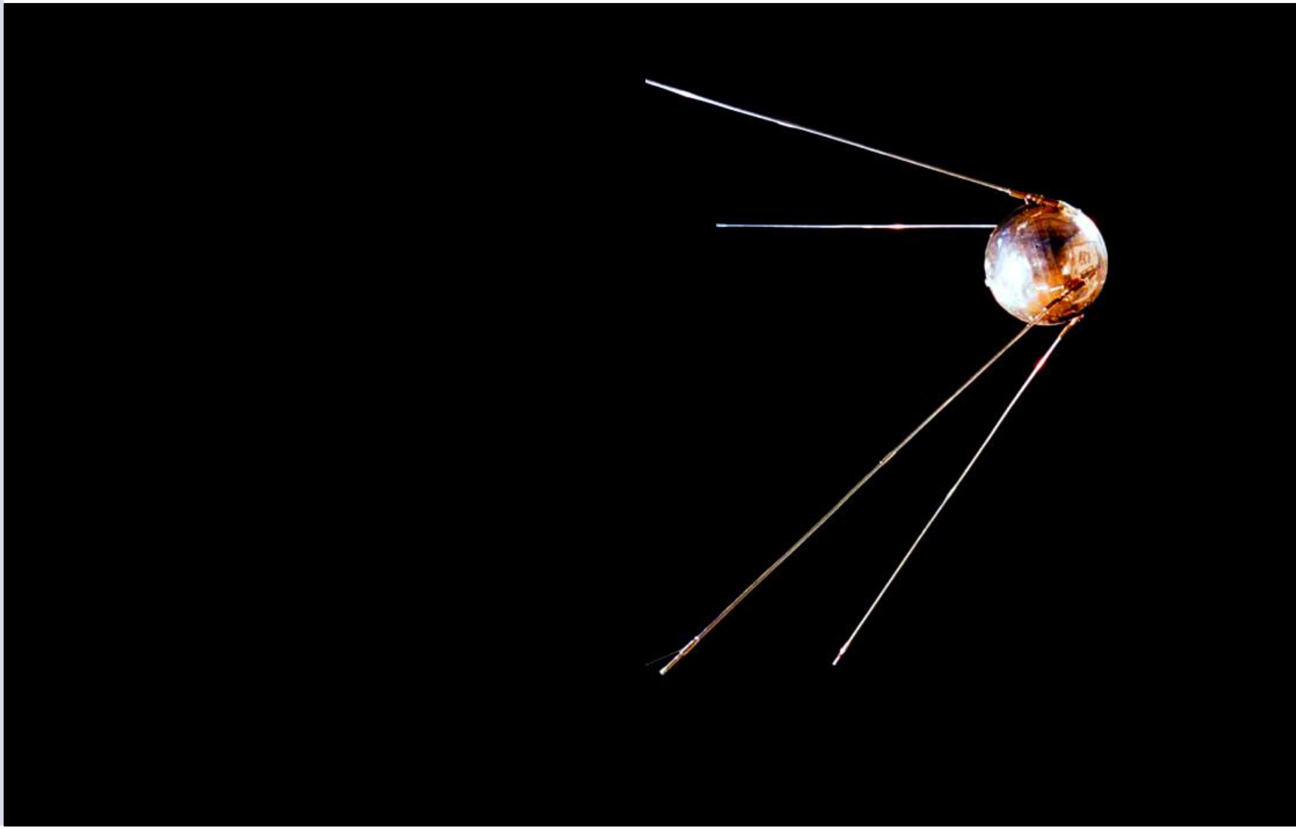
Mobile Phone Masts

Bluetooth

WiFi

地理→参考 定位 系统

Georeferenced Positioning Systems (GPS)



The University Of Sheffield.

SHEFFIELD ROBOTICS

Outdoor Methods

米 级

- GPS provides **metre level** outdoor positioning using information from multiple satellites.

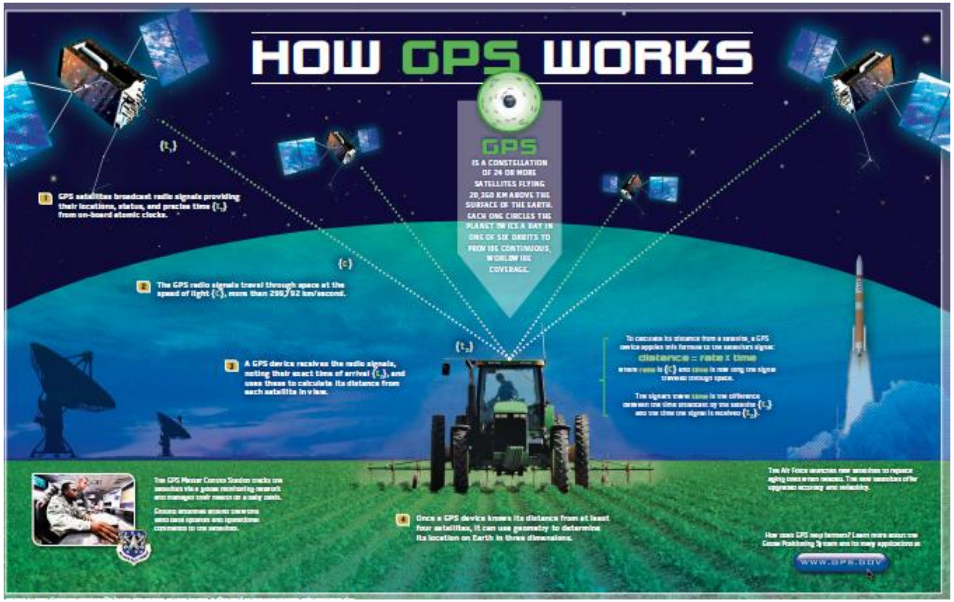
大气

- **Atmospheric conditions limit precision** unless corrected using a reference (Differential GPS).

差分

耗电

- Can be **power hungry**.
- Methods also used by mobile phone masts (三角测量 **triangulation**).
- Doesn't work **indoors**!



Indoor Methods

- WiFi and Bluetooth beacons are providing positioning systems for indoors, but accuracy remains limited.

- Various methods:
  - "In-range"
  - Signal drop-off (attenuation) 信号衰减
  - Triangulation 三角测量

- Some systems use difference in **sound** 声波 and **electromagnetic** 电磁波 waves from the same source.

Limitations

1. Accuracy not sufficient for many tasks:
   – GPS ~2m at the very best
   – Gets worse in urban environments

2. Need for infrastructure limits applications:
   – Exploration of new places (e.g. Mars!)
   – Disaster areas

3. Security

外部 感知

4. Example of **exteroceptive sensing** (using external cues)

But: Robots also need to be able to track their position in the absence of external cues

没有

车载 定位 里程计

On-board Positioning: *Odometry*

The use of data from motion sensors to estimate change in position (or *pose*) over time. 运动传感器

Used by many mobile robots to estimate their position relative to a starting location. 相对起始位置

The word odometry is composed from the Greek words **odos** (meaning *"route"*) and **metron** (meaning *"measure"*). 路线测量
[www.wikipedia.org]

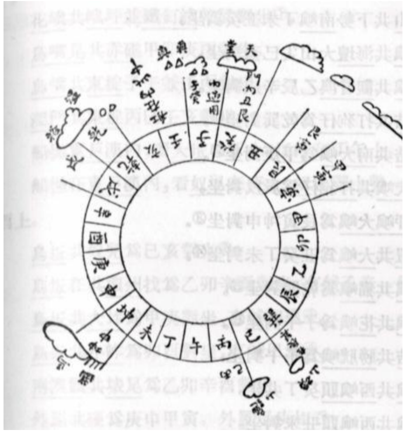
Mars Rover Odometry System:
Distance travelled: Vision
Orientation: Vision

航位 推算

Dead Reckoning – *Ded Reckoning*

从船上扔入个东西，根据静止判断船向

Deducing ones position by deduction of speed through observation of 'dead object' (e.g. barrel or log) thrown overboard plus a compass bearing.



Song Dynasty maritime compass (12th century).

Distance travelled:
Relative to a 'stationary' object over time
Orientation:
Magnetic compass

路径 整合

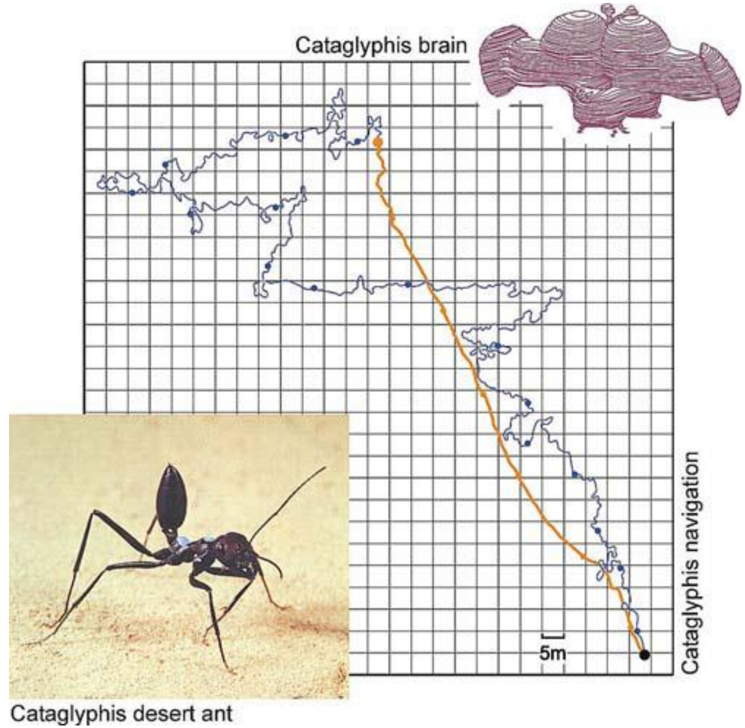
Path Integration

Path integration is the name given to the method thought to be used by animals for dead reckoning.

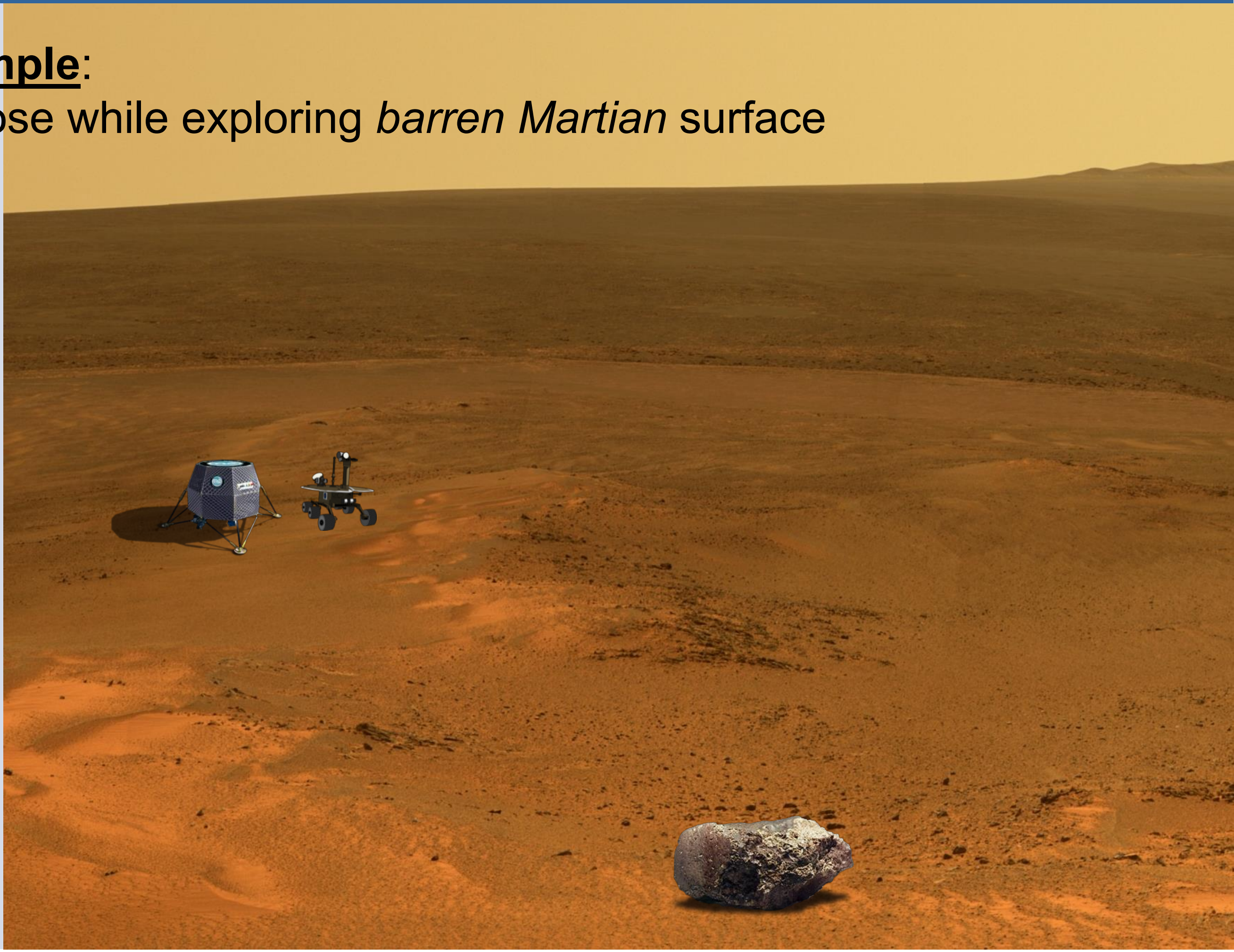
[www.wikipedia.org]

动物的航位推算

Distance travelled:
Step counting / optic flow
Orientation:
Polarised light compass



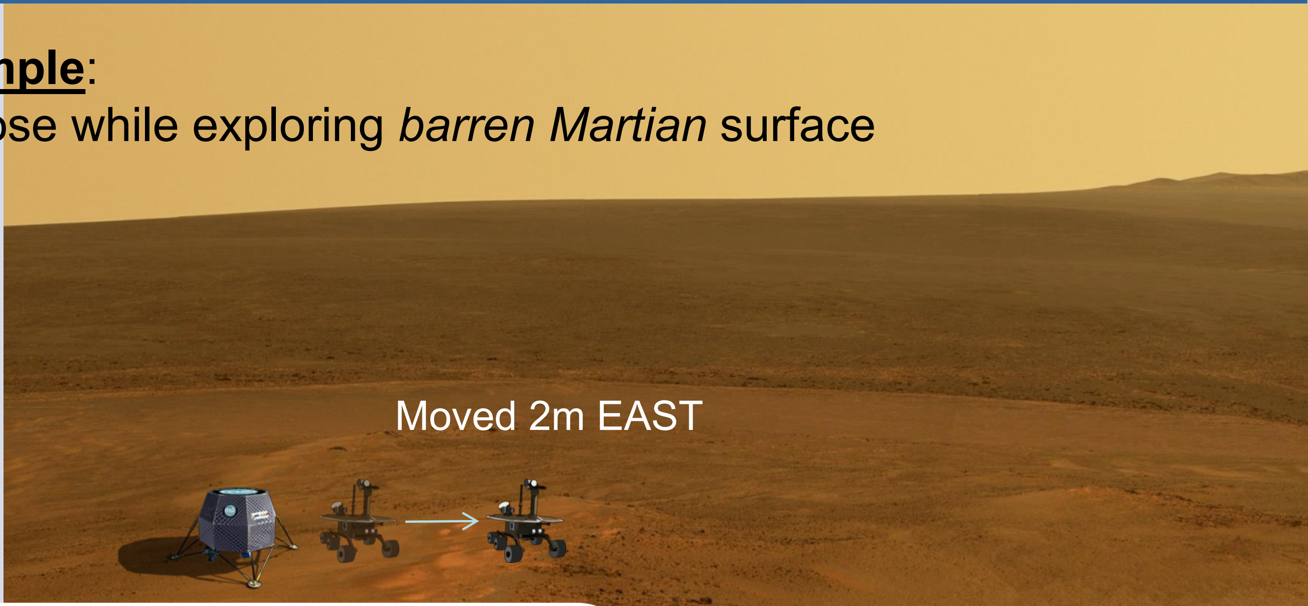
Worked Example:
Track robot pose while exploring *barren Martian* surface



Worked Example:

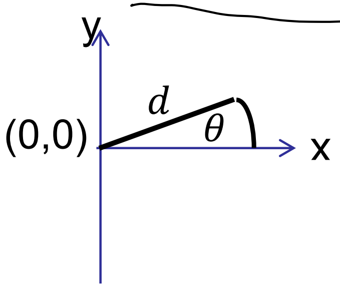
Track robot pose while exploring *barren Martian surface*

用相对参考系

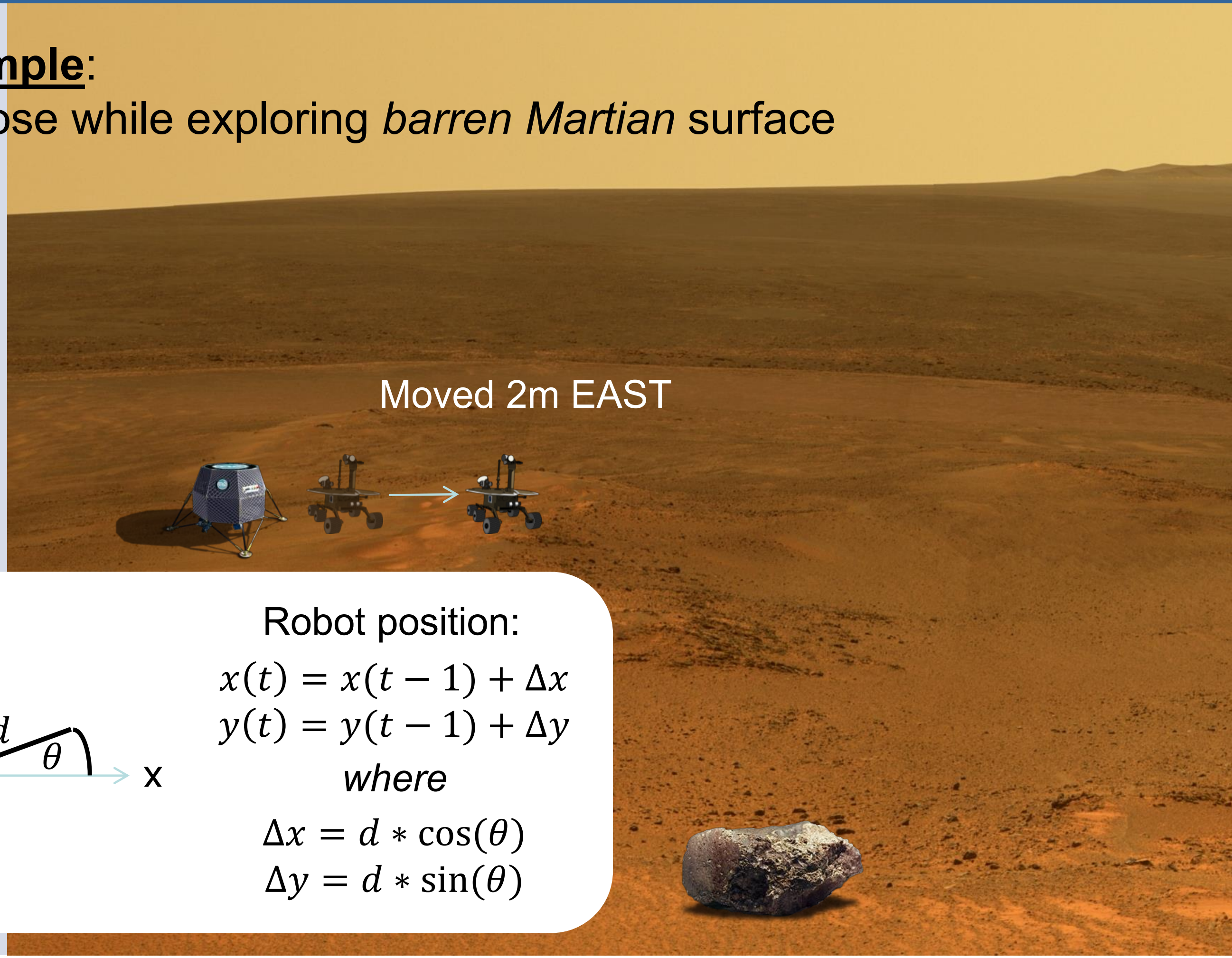


Define frame of reference:

参考系



<u>Worked Example</u>:
Track robot pose while exploring *barren Martian* surface



Robot position:

$$x(t) = x(t-1) + \Delta x$$
$$y(t) = y(t-1) + \Delta y$$

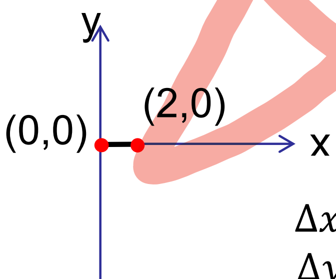
where

$$\Delta x = d * \cos(\theta)$$
$$\Delta y = d * \sin(\theta)$$

Worked Example:
Track robot pose while exploring *barren Martian* surface

Moved 2m EAST

$d = 2$
$\theta = 0$



Robot position:

$$x(1) = 0 + 2 = 2$$
$$y(1) = 0 + 0 = 0$$

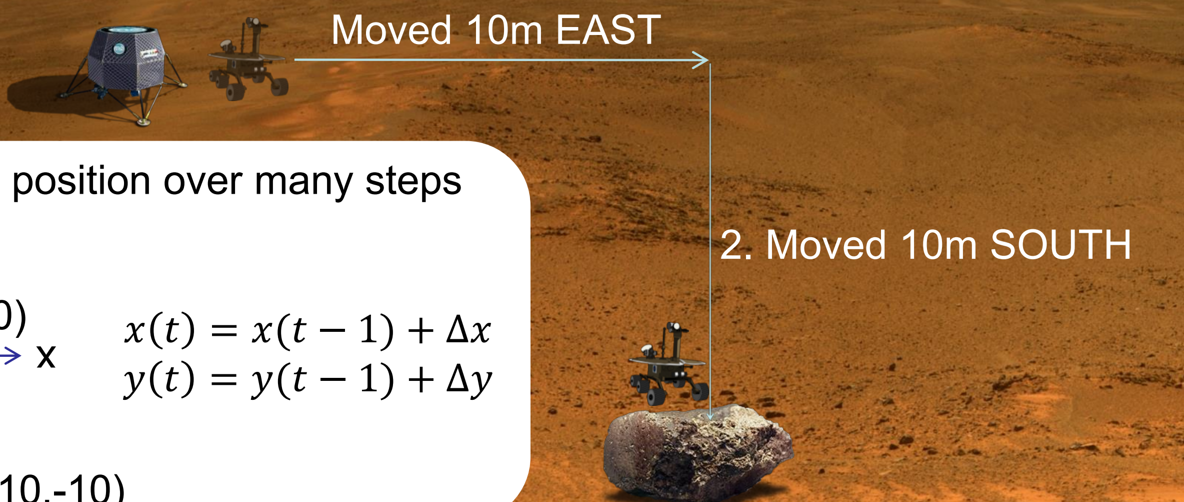
where

$$\Delta x = 2 * \cos(0) = 2 * 1 = 2$$
$$\Delta y = 2 * \sin(0) = 2 * 0 = 0$$

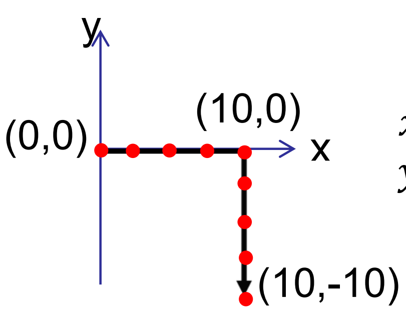
Worked Example:

Track robot pose while exploring *barren Martian surface*

Now you try:
See worksheet



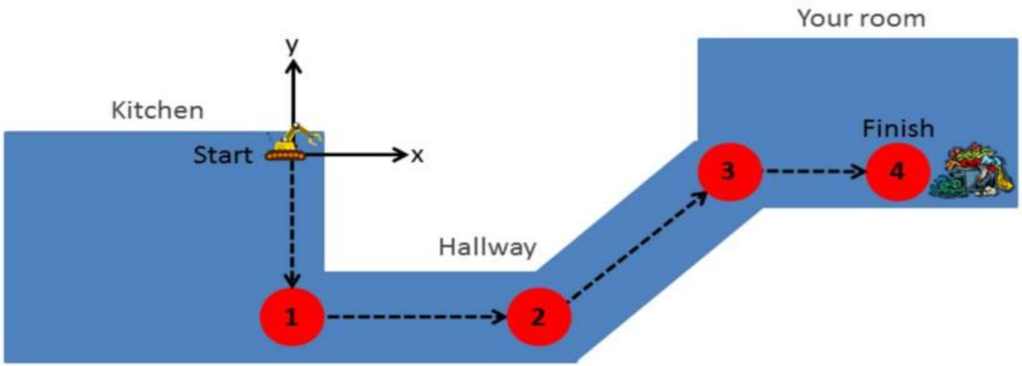
Iterate to track position over many steps



$$x(t) = x(t-1) + \Delta x$$
$$y(t) = y(t-1) + \Delta y$$

Odometry Worked Example

Dead-reckoning (Odometry) Worksheet



Task: Your new flat cleaning robot has moved from its base in the kitchen, down the hall, into your room and collected a pile of dirty clothes. Every 3 meters the robot updates its position using its dead-reckoning system. See the example below for a system which provides perfect readings and then try to calculate the positioning accuracy when realistic noise is added to the sensor readings...

			Perfect Odometry Readings	You have just mopped the floor in the hall causing your robot's wheels to slip, giving false distance readings...	The robot's magnetic compass is receiving electrical interference from the motors, creating noisy orientation readings...	Units
1	Odometry Readings	Distance	3	2.75	3	m
		Orientation	-90	-90	-98	degrees
	Change in Position	Δx	0.00			m
		Δy	-3.00			m
	New Position	x(1)	0.00			m
		y(1)	-3.00			m
	Accumulated Error (Euclidian)		0			m

$$x(t) = x(t-1) + \Delta x$$

$$\Delta x = d\cos\theta$$

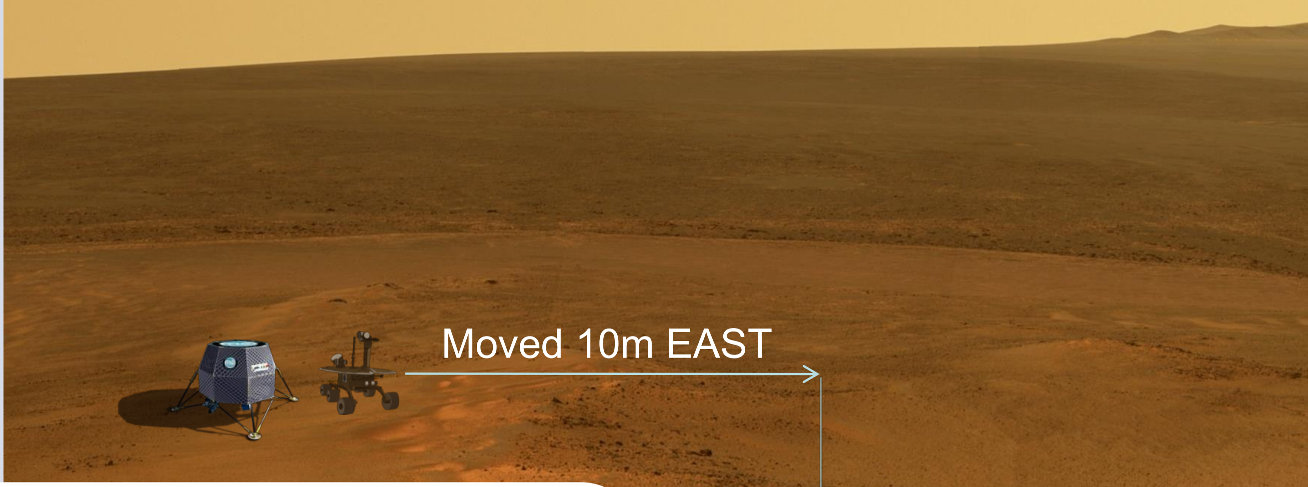
$$y(t) = y(t-1) + \Delta y$$

$$\Delta y = d\sin\theta$$

Odometry Worked Example

Waypoint	Parameter		Perfect Odometry Readings	You have just mopped the floor in the hall causing your robot's wheels to slip, giving false distance readings...	The robot's magnetic compass is receiving electrical interference from the motors, creating noisy orientation readings...	Units
1	Odometry Readings	Distance	**3**	**2.75**	**3**	**m**
		Orientation	**-90**	**-90**	**-98**	**degrees**
	Change in Position	Δx	0.00	0.00	-0.42	m
		Δy	-3.00	-2.75	-2.97	m
	New Position	x(1)	0.00	0.00	-0.42	m
		y(1)	-3.00	-2.75	-2.97	m
	Accumulated Error (Euclidian)		0	0.250	0.421	m
2	Odometry Readings	Distance	**3**	**3.5**	**3**	**m**
		Orientation	**0**	**0**	**-6**	**degrees**
	Change in Position	Δx	3.00	3.50	2.98	m
		Δy	0.00	0.00	-0.31	m
	New Position	x(2)	3.00	3.50	2.56	m
		y(2)	-3.00	-2.75	-3.28	m
	Accumulated Error (Euclidian)		0	0.559	0.522	m
3	Odometry Readings	Distance	**3**	**3.25**	**3**	**m**
		Orientation	**45**	**45**	**38**	**degrees**
	Change in Position	Δx	2.12	2.30	2.36	m
		Δy	2.12	2.30	1.85	m
	New Position	x(3)	5.12	5.80	4.92	m
		y(3)	-0.88	-0.45	-1.43	m
	Accumulated Error (Euclidian)		0	0.805	0.585	m
4	Odometry Readings	Distance	**3**	**2.9**	**3**	**m**
		Orientation	**0**	**0**	**9**	**degrees**
	Change in Position	Δx	3.00	2.90	2.96	m
		Δy	0.00	0.00	0.47	m
	New Position	x(4)	8.12	8.70	7.88	m
		y(4)	-0.88	-0.45	-0.96	m
	Accumulated Error (Euclidian)		0	0.722	0.253	m

Odometry: *issues*

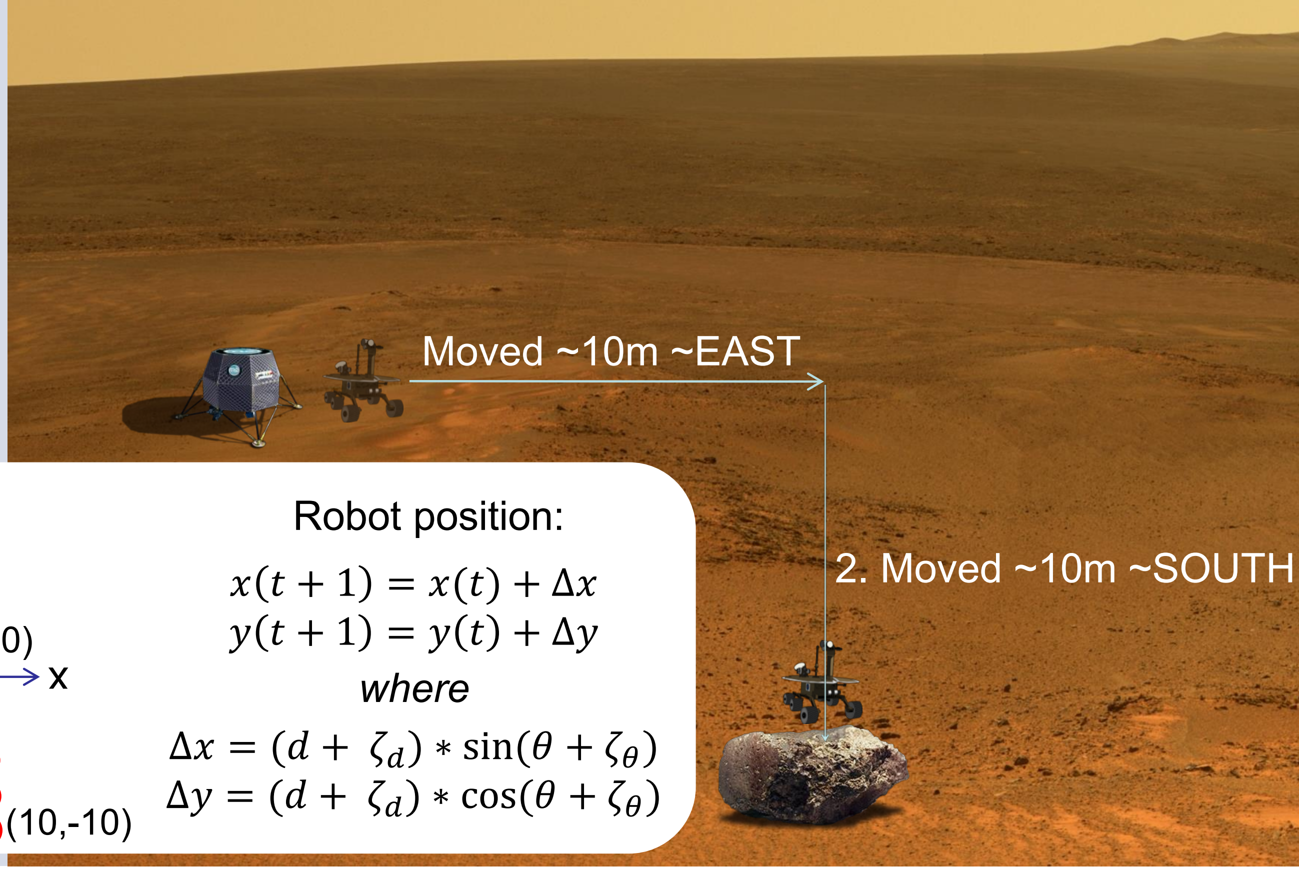


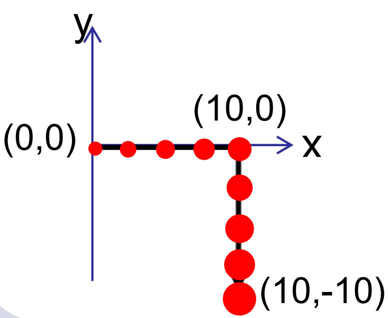
Sensing Errors Accumulate


e.g. wheel slip


e.g. interference

Odometry: *issues*





Robot position:

$$x(t+1) = x(t) + \Delta x$$
$$y(t+1) = y(t) + \Delta y$$

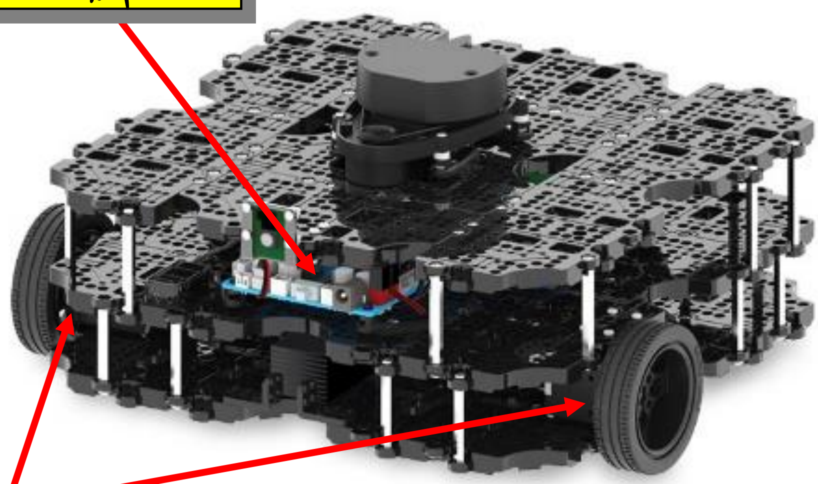
where

$$\Delta x = (d + \zeta_d) * \sin(\theta + \zeta_\theta)$$
$$\Delta y = (d + \zeta_d) * \cos(\theta + \zeta_\theta)$$

On-board Positioning: *Odometry*

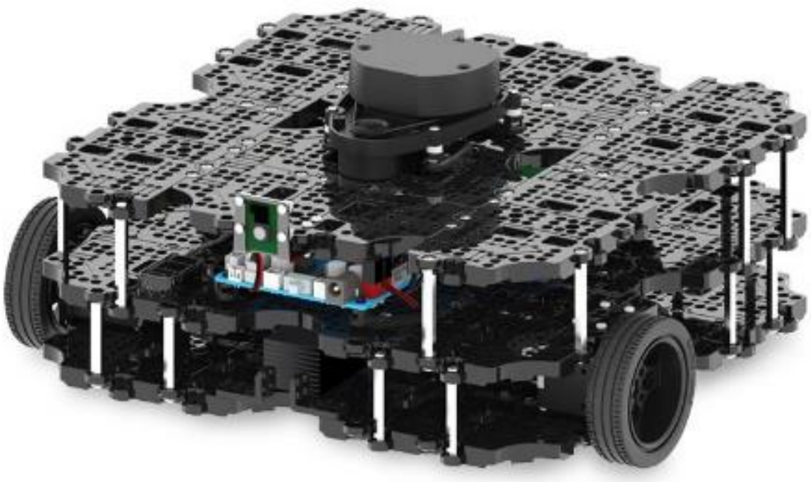
Inertial Measurement Unit (IMU):
- **Gyroscope:** Angular Velocity 陀螺仪：角速度
- **Accelerometer:** Linear Acceleration 加速度计：线性加速度
- **Compass:** Orientation 指南针：方向

TurtleBot3 Odometry System



Wheel encoders

Odometry issues: ***Demo***



Odometry: *issues*

See: https://www.instructables.com/id/RB2-running-squares/

Summary of Odometry

1. **Odometry == Dead Reckoning == Path Integration**

2. **Requires two sources of information, monitored *incrementally* (step by step):**
   1. **Distance** travelled between updates
   2. **Orientation** followed between updates

受到 累积/传播误的影响

3. **Suffers from accumulated/propagated errors due to iterative process limiting its use to short ranges.**

The University Of Sheffield.

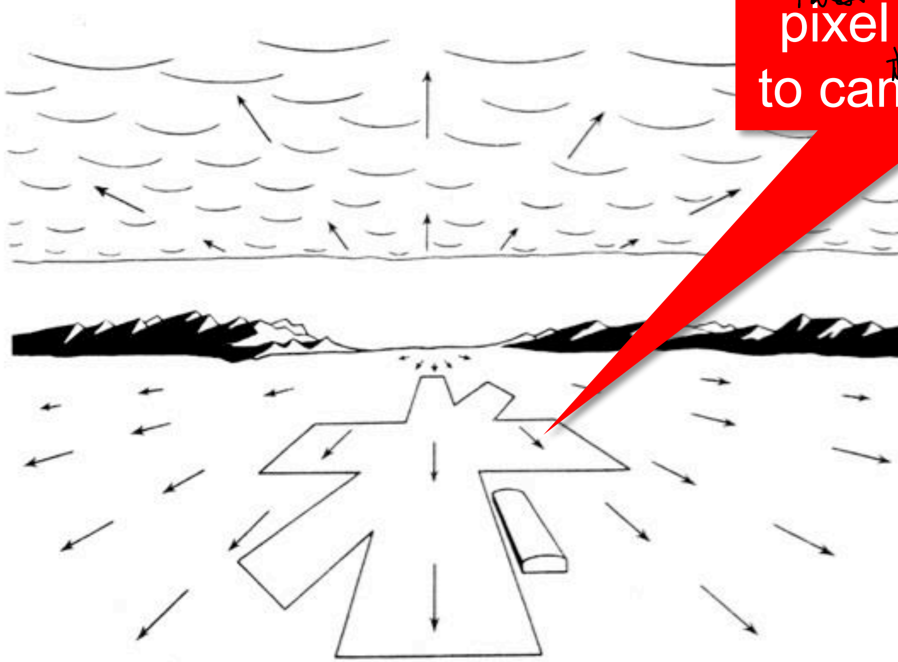
SHEFFIELD ROBOTICS

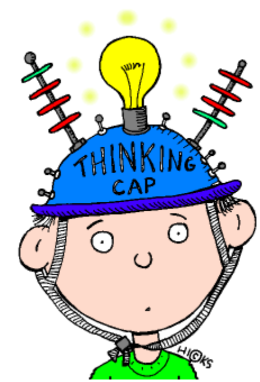
光 流

Optical Flow

像素 运动 相机 运动

Can we relate pixel movements to camera motion?





J.J. Gibson – The Ecological Approach to Visual Perception

视觉 里程计

Visual Odometry

$$\begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} -\hat{f}/Z & 0 & u/Z & -uv/\hat{f} & -(\hat{f}+\frac{u^2}{\hat{f}}) & v \\ 0 & -\hat{f}/Z & v/Z & \hat{f}+v^2/\hat{f} & -uv/\hat{f} & -u \end{pmatrix} \begin{pmatrix} v_x \\ v_y \\ v_z \\ \varpi_x \\ \varpi_y \\ \varpi_z \end{pmatrix}$$

像素 速度

Pixel velocity

Can be computed using optical flow

图像 雅可比

Image Jacobian.

All parameters (u,v, $\hat{f}$,Z) can be defined

相机 速度

Camera velocities

Visual Odometry

$$\begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} -\hat{f}/Z & 0 & u/Z & -uv/\hat{f} & -(\hat{f} + \frac{u^2}{\hat{f}}) & v \\ 0 & -\hat{f}/Z & v/Z & \hat{f} + v^2/\hat{f} & -uv/\hat{f} & -u \end{pmatrix} \begin{pmatrix} v_x \\ v_y \\ v_z \\ \varpi_x \\ \varpi_y \\ \varpi_z \end{pmatrix}$$

Pixel velocity

Can be computed using optical flow

Image Jacobian.

All parameters (u,v, $\hat{f}$,Z) can be defined

In most circumstances u/Z=v/Z=0

Camera velocities

v_z=0 for UAV at fixed height.

Can measure angular velocities (ω) using gyro

Visual Odometry

将角速度聚集成 w

Clustered angular velocities into ω

$$\begin{pmatrix}\dot{u}\\ \dot{v}\end{pmatrix} = \begin{pmatrix}-\hat{f}/Z & 0 & 0 & J_u\\ 0 & -\hat{f}/Z & 0 & J_v\end{pmatrix}\begin{pmatrix}v_x\\ v_y\\ 0\\ \omega\end{pmatrix}$$

Clustered last three terms into Ju and Jv.

多个括号外

Multiple out brackets

$$\dot{u} = -\frac{v_x\hat{f}}{Z} + J_u\,\omega$$

$$\dot{v} = -\frac{v_y\hat{f}}{Z} + J_v\,\omega$$

重排 相机V

Re-arrange for camera velocities in x and y

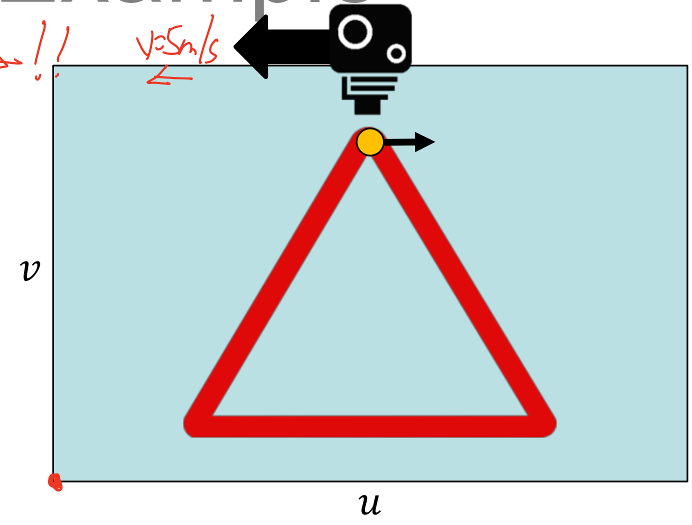
$$v_x = \frac{z(J_u\omega - \dot{u})}{\hat{f}}$$

$$v_y = \frac{z(J_v\omega - \dot{v})}{\hat{f}}$$

From Lecture 7

Worked Example

→对应上面的公式　考题!!



Task:

For a camera with $\hat{f} = 1$ what is the velocity of a pixel at [u,v]=[0,0], Z = 100cm when the camera moves left at 5 cm/s?

Solution:

Define camera motion

$$\begin{pmatrix} v_x \\ v_y \\ v_z \\ \varpi_x \\ \varpi_y \\ \varpi_z \end{pmatrix} = \begin{pmatrix} -5 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Compute Jacobian

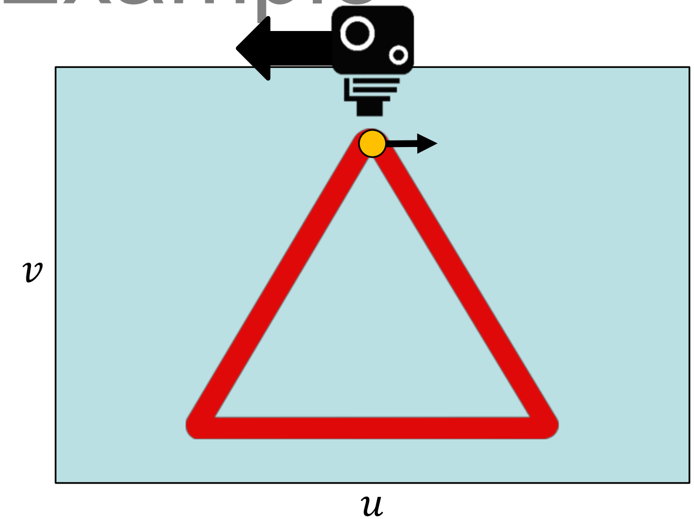
$$J_p(u, v, Z) = \begin{pmatrix} -0.01 & 0 & 0 & 0 & -1 & 0 \\ 0 & -0.01 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Compute pixel velocity

$$\begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} -0.01 & 0 & 0 & 0 & -1 & 0 \\ 0 & -0.01 & 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} -5 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0.05 \\ 0 \end{pmatrix}$$

Worked Example

Task:
For a camera with $\hat{f}$ = 1 what is the velocity of a pixel at [u,v]=[0,0], Z = 100cm when the camera moves left at 5 cm/s?



Check with new equations:

$$v_x = \frac{z(J_u\omega - \dot{u})}{\hat{f}} \qquad v_y = \frac{z(J_v\omega - \dot{v})}{\hat{f}}$$

No angular rotations therefore $\omega = 0$

$$v_x = \frac{-z * \dot{u}}{\hat{f}} \qquad v_y = \frac{-z * \dot{v}}{\hat{f}}$$

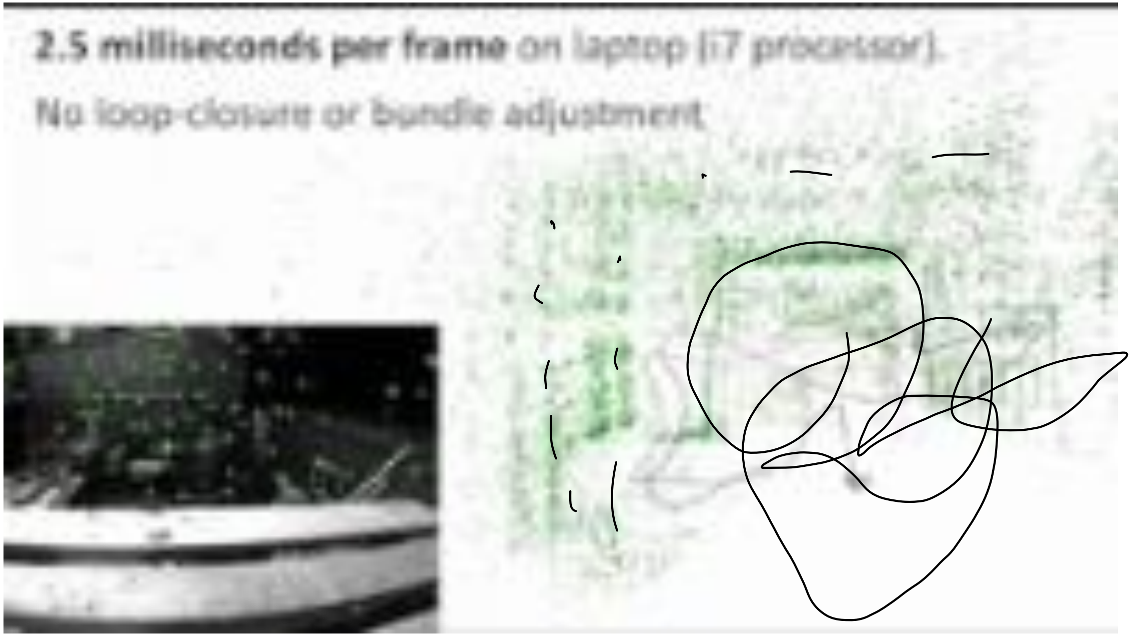
From previous slide $\begin{pmatrix}\dot{u}\\ \dot{v}\end{pmatrix} = \begin{pmatrix}0.05\\ 0\end{pmatrix}$

$$v_x = \frac{-100 * 0.05}{1} \qquad v_y = 0$$

$$v_x = \frac{-100 * 0.05}{1} = -5$$

The same as in previous lecture

Visual Odometry on Robots



SVO 2.0 Semi-visual odometry; Forster et al, 2016
https://www.youtube.com/watch?v=hR8uq1RTUfA

Where to find out more

1. **Dr Mike Mangan, University of Sheffield:**
   Final year projects in this area

2. **Reading Materials:**
   - Textbook: Peter Corke, Robotics, Vision and Control
   - Animal PI Review: Heinze, Stanley, Ajay Narendra, and Allen Cheung. "Principles of insect path integration." Current Biology 28.17 (2018): R1043-R1058.

3. **Online Materials:**
   Robot Academy MOOC (visual odometry)

This lecture has covered …

1. External positioning systems

2. On-board positioning

   – **Classic** Odometry

   – **Visual** Odometry

   (a.k.a: Dead-reckoning, Path Integration)