ROS for Educators

Part 0: The Plan

Mac Mason:	How ROS makes robots go
Hands-on session	("Making robots go")
Michael Ferguson:	Advanced ROS
Sharon Small:	An undergraduate pilot course
Coffee Break!	
Zach Dodds:	ROS and the ARDrone in CS2
Hands-on session	("Making robots do more than go")

http://www.cs.duke.edu/~mac/sigcse12.pdf

Part 1: Why ROS?

Part 2: The Big Idea

Robots cause problems.

- Will my robot work?
- Will my camera work?
- Can I use a Kinect?
- Will my joystick work?
- I have to synchronize everything manually!
- ...and so I end up with spaghetti.
- My students don't know {C++, Python, Java, LISP, Haskell, JavaScript, Scratch}!

Part 2: The Big Idea

Nodes and *Topics*

Details: *many-to-many, peer-to-peer, strongly-typed (a)synchronous language-agnostic network IPC*

Part 2: The Big Idea

Nodes and *Topics*

Details: *many-to-many, peer-to-peer, strongly-typed (a)synchronous language-agnostic network IPC*

Part 2: The Big Idea

Nodes and *Topics*

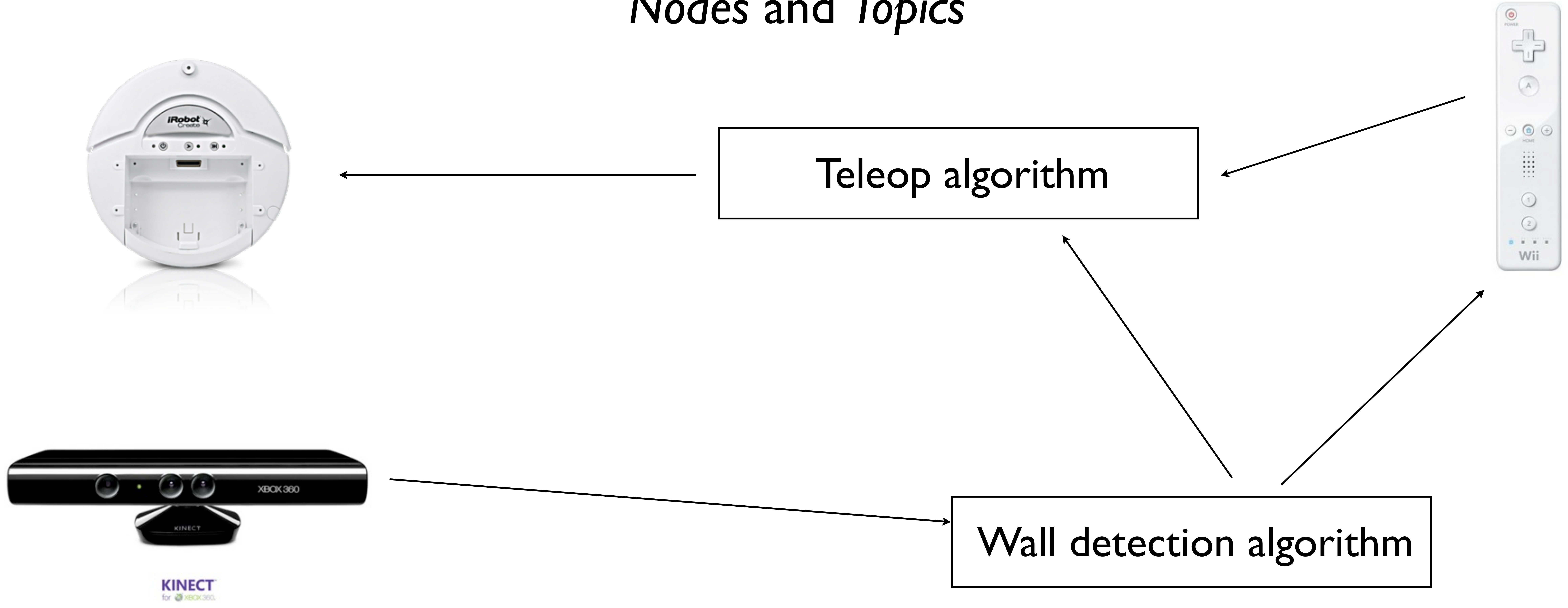
Teleop algorithm

Wall detection algorithm

Details: *many-to-many, peer-to-peer, strongly-typed (a)synchronous language-agnostic network IPC*

Part 2: The Big Idea

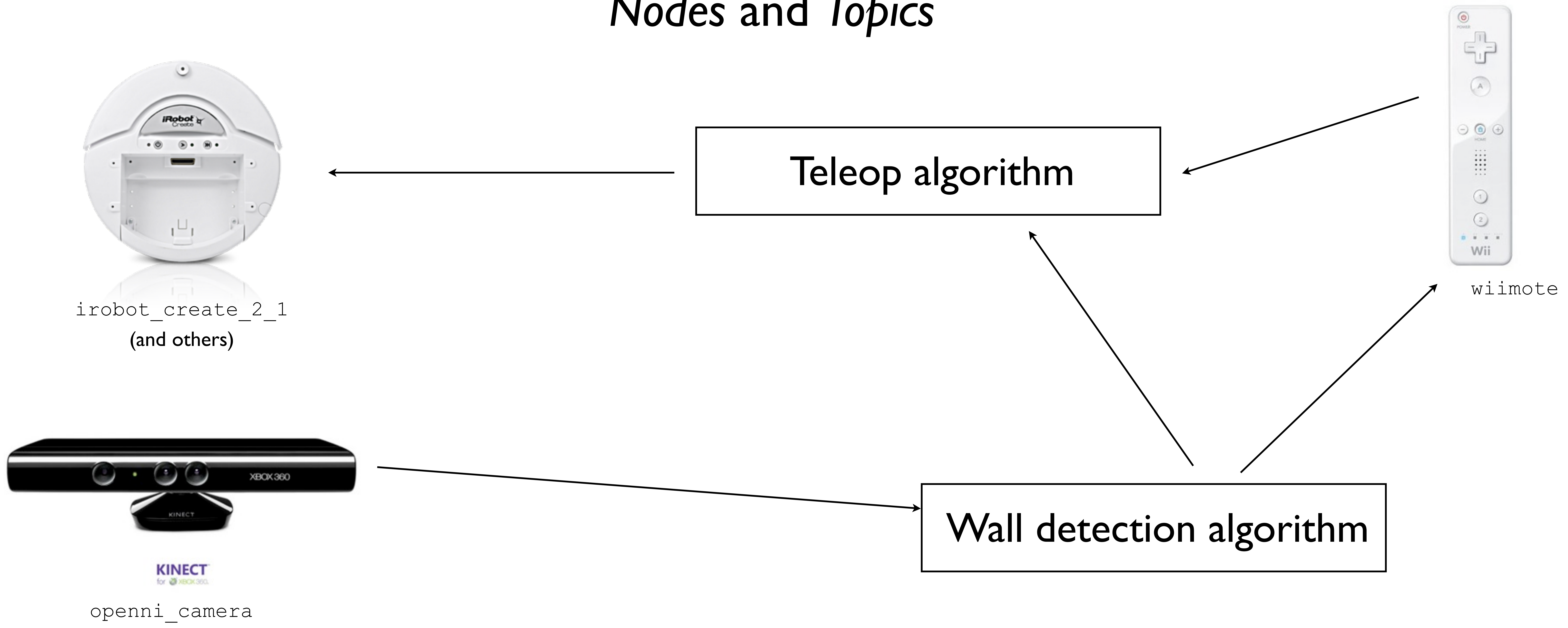
Nodes and *Topics*



Details: *many-to-many, peer-to-peer, strongly-typed (a)synchronous language-agnostic network IPC*

Part 2: The Big Idea

Nodes and *Topics*



Details: *many-to-many, peer-to-peer, strongly-typed (a)synchronous language-agnostic network IPC*

Part 2: The Big Idea

Topics do many-to-many transmission of ROS *messages*

Publishing messages

```
#!/usr/bin/env python
import roslib
roslib.load_manifest('beginner_tutorials')
import rospy
from std_msgs.msg import String

def talker():
    pub = rospy.Publisher('chatter', String)
    rospy.init_node('talker')
    while not rospy.is_shutdown():
        str = "hello world %s" % rospy.get_time()
        pub.publish(String(str))
        rospy.sleep(1.0)
if __name__ == '__main__':
    talker()
```

Subscribing to messages

```
#!/usr/bin/env python
import roslib
roslib.load_manifest('beginner_tutorials')
import rospy
from std_msgs.msg import String

def callback(data):
    print "I heard", data.data

def listener():
    rospy.init_node('listener')
    rospy.Subscriber("chatter", String, callback)
    rospy.spin()

if __name__ == '__main__':
    listener()
```

(Code from the truly excellent ROS tutorials at ros.org)

Details: *many-to-many, peer-to-peer, strongly-typed (a)synchronous language-agnostic network IPC*