

Pushing

and

Pulling

*Computing push plans for disk-shaped robots
and dynamic labelings for moving points*

Dirk Gerrits



“I do Computer Science.”

“I do Computer Science.”

“Cool!”

“I do Computer Science.”

“Cool! So I have this problem with my computer...”

```
A problem has been detected and windows has been shut down to prevent damage to your computer.

DRIVER_IRQL_NOT_LESS_OR_EQUAL

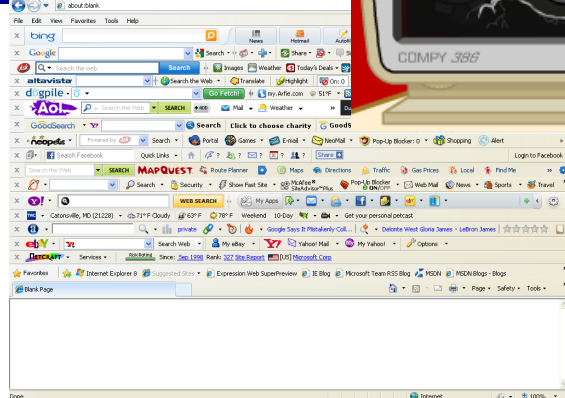
If this is the first time you've seen this stop error screen,
restart your computer. If this screen appears again, follow
these steps:

Check to make sure any new hardware or software is properly installed.
If this is a new installation, ask your hardware or software
manufacturer for any windows updates you might need.

If problems continue, disable or remove any newly installed
hardware or software. Disable BIOS memory options such as caching or
shadowing. If you need to use Safe Mode to remove or disable components,
restart your computer, press F8 to select Advanced Startup Options, and
then select Safe Mode.

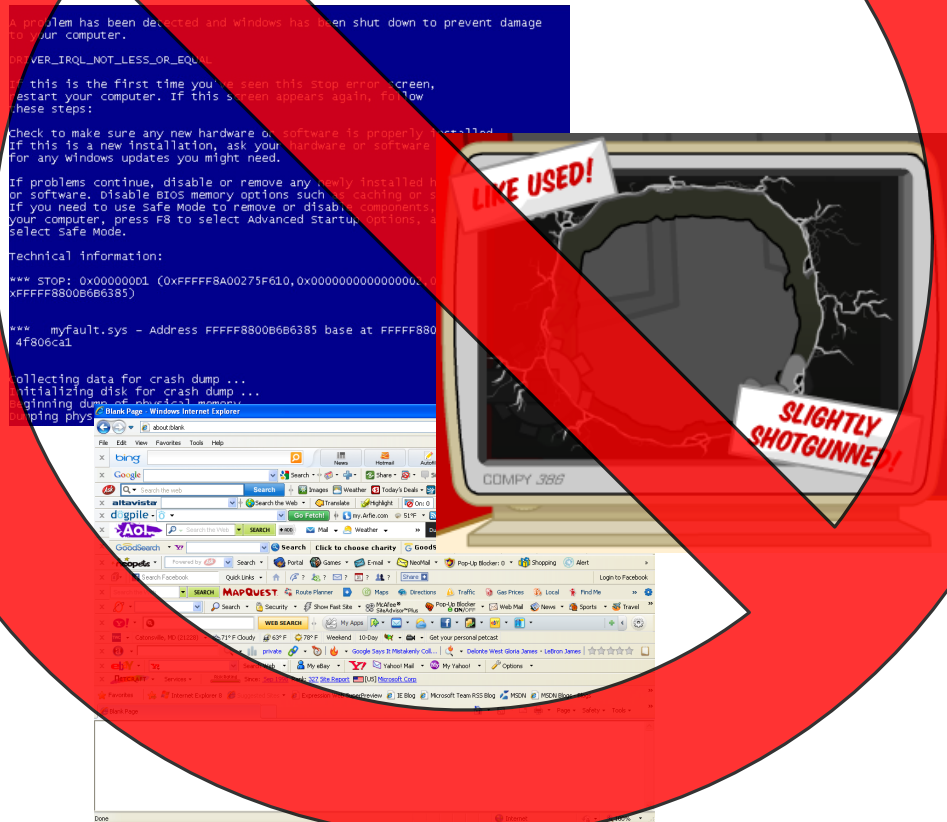
Technical information:
*** STOP: 0x00000001 (0xFFFFF8A0275F610, 0x0000000000000002, 0
0xFFFFF8B00B6B385)
*** myfault.sys - Address FFFFF8B00B6B385 base at FFFFF8B0
4F806ca1

collecting data for crash dump ...
initializing disk for crash dump ...
beginning dump of memory ...
dumping physical memory ...
```



"I do Computer Science."

"Cool! So I have this problem with my computer..."

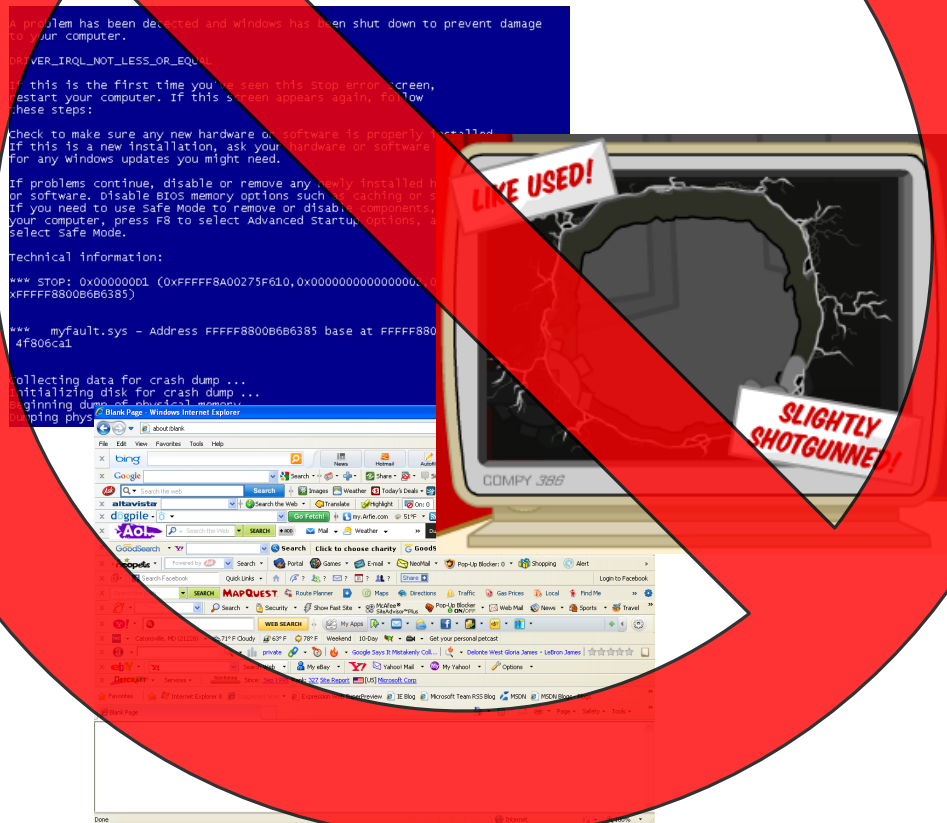


Not: humans solving computer problems

But: computers solving human problems

"I do Computer Science."

"Cool! So I have this problem with my computer..."



Not: humans solving computer problems

But: computers solving human problems

- What problems can they solve?
- How efficiently can they solve them?

Problem 1: Pushing Objects with Robots



Problem 1: Pushing Objects with Robots



1. build a strong robot (easy)

Problem 1: Pushing Objects with Robots



1. build a strong robot (easy)
2. make it find the right pushing motion (hard)

Problem 1: Pushing Objects with Robots

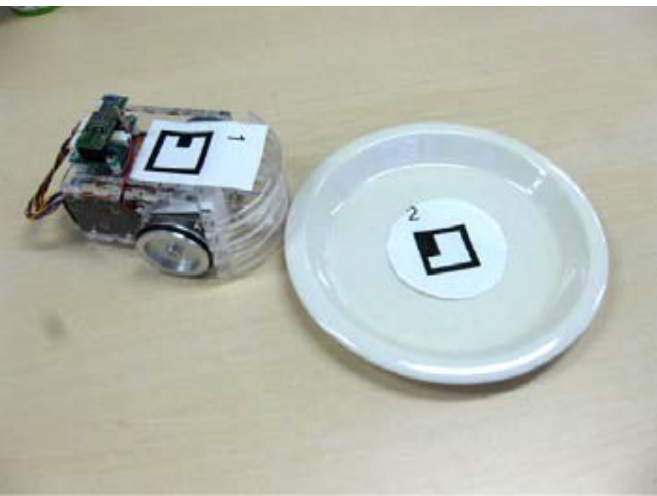


1. build a strong robot (easy)
2. make it find the right pushing motion (hard)

“How hard could it be?
A child can do it!”

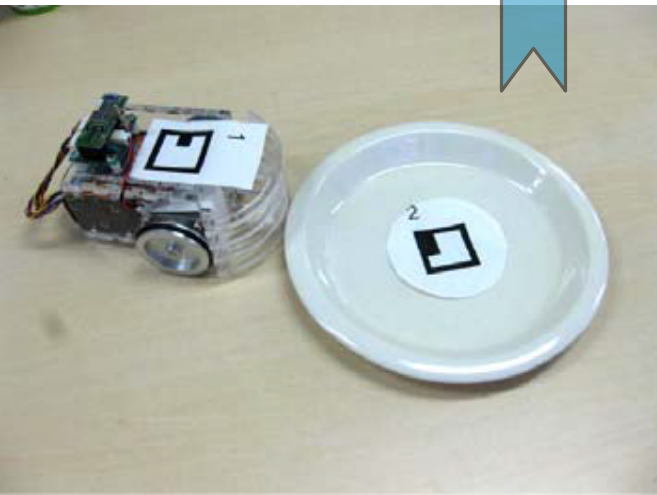
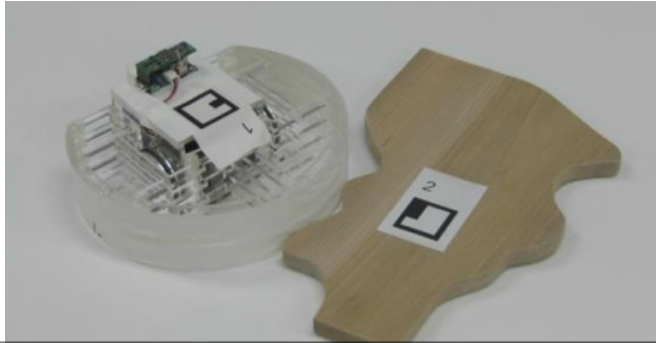


Baby Steps in Pushing



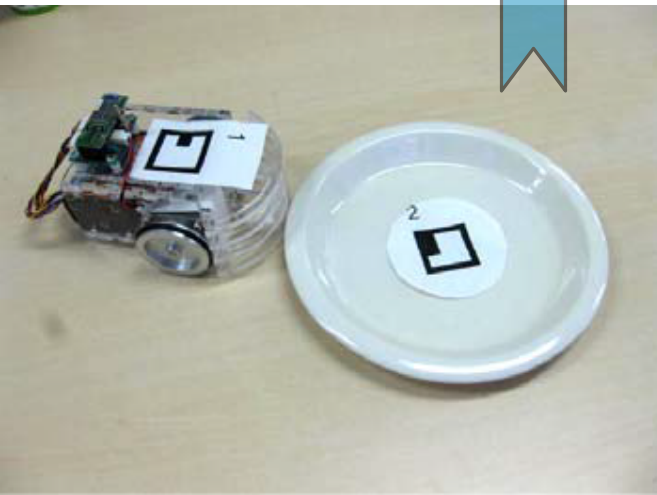
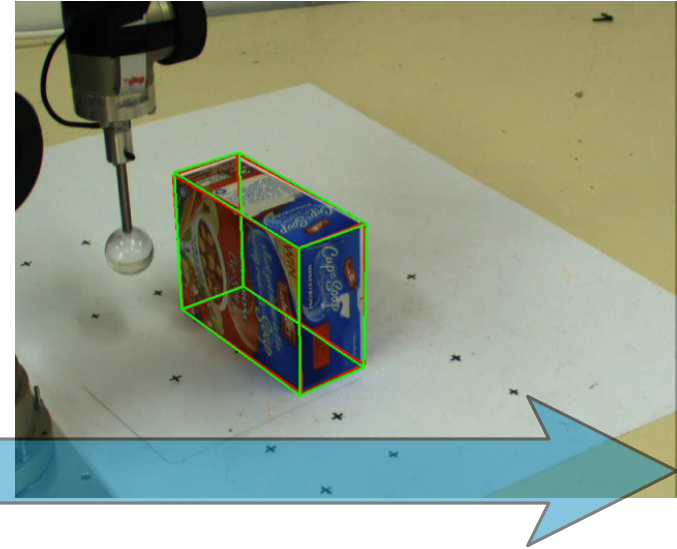
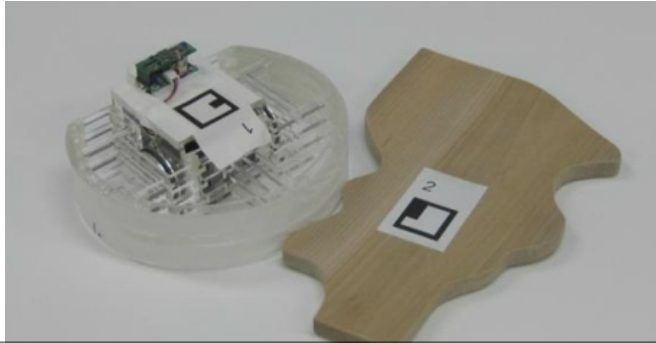
Baby Steps in Pushing

more complex
shapes



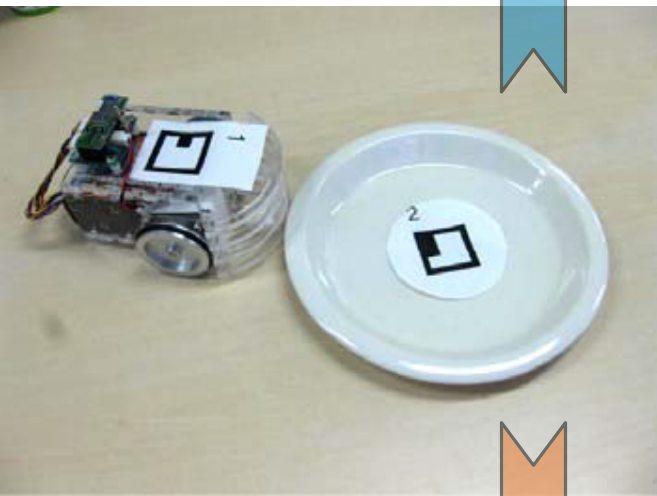
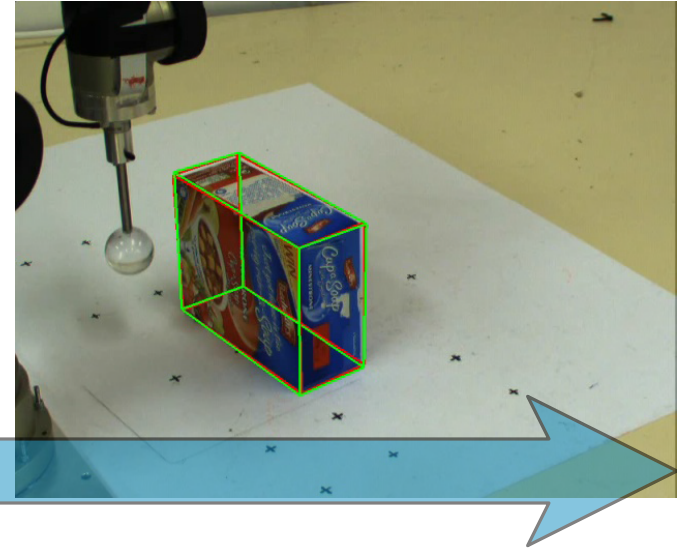
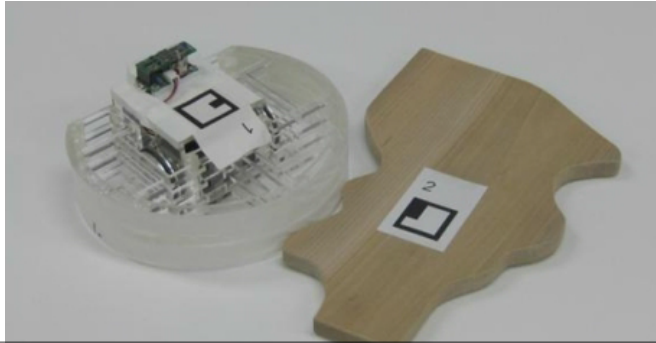
Baby Steps in Pushing

more complex
shapes

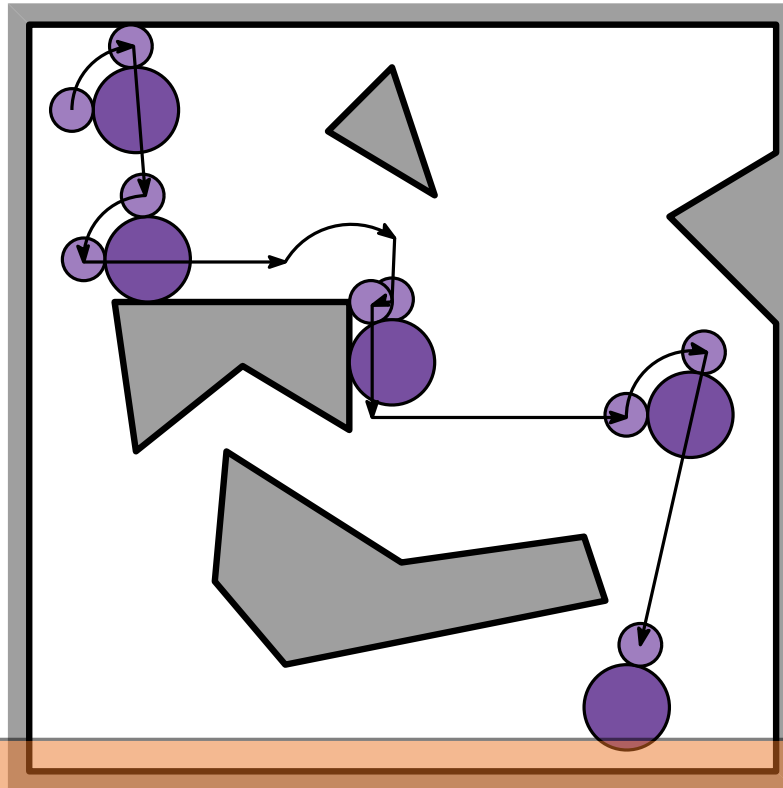


Baby Steps in Pushing

more complex
shapes

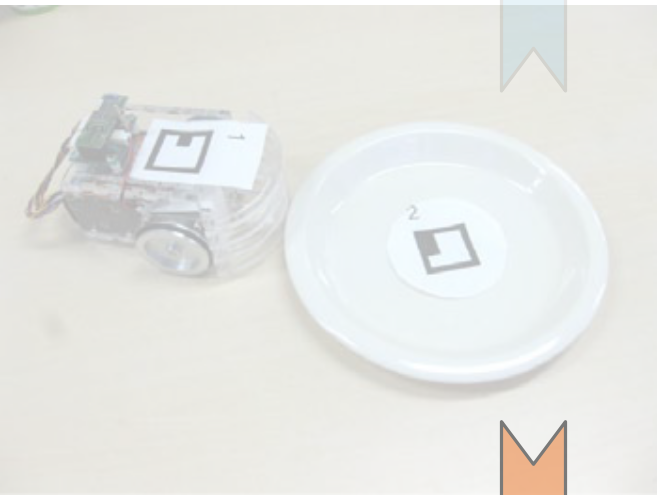
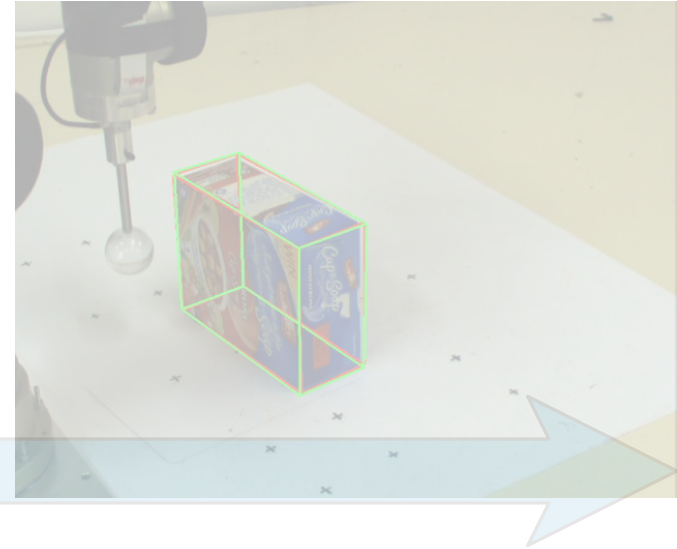


more complex
pushing tasks

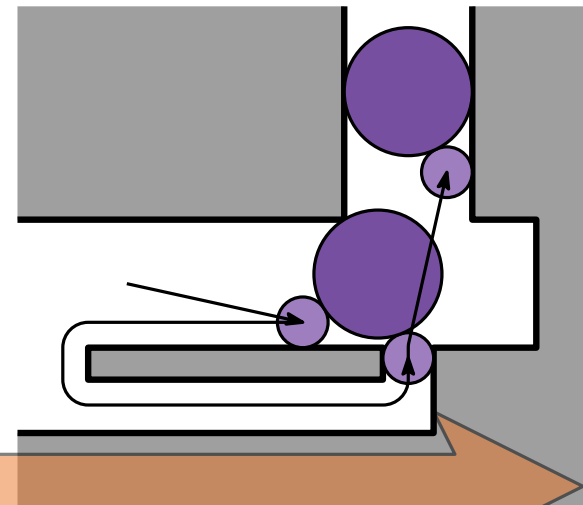
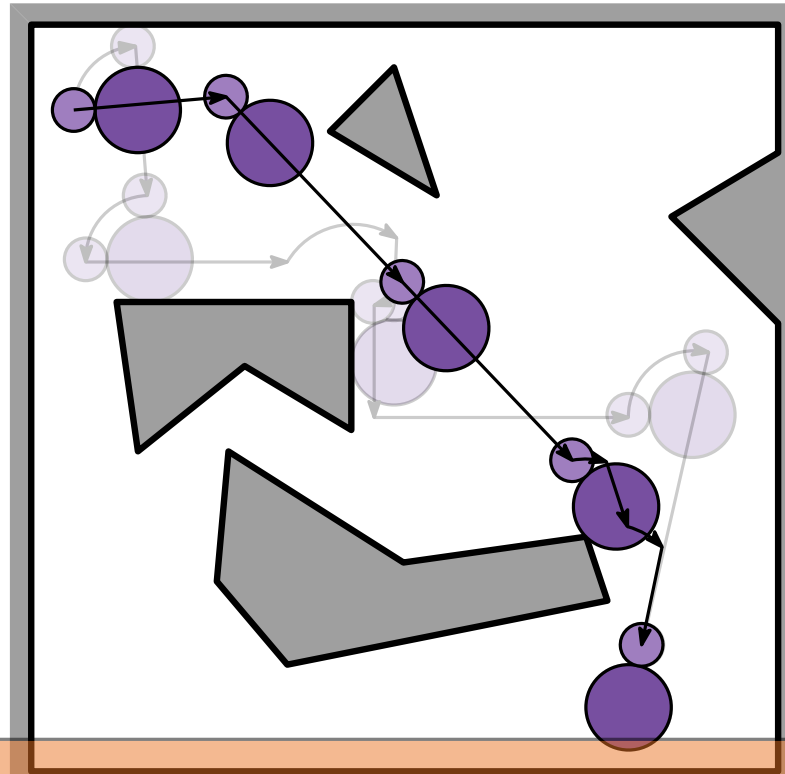


Baby Steps in Pushing

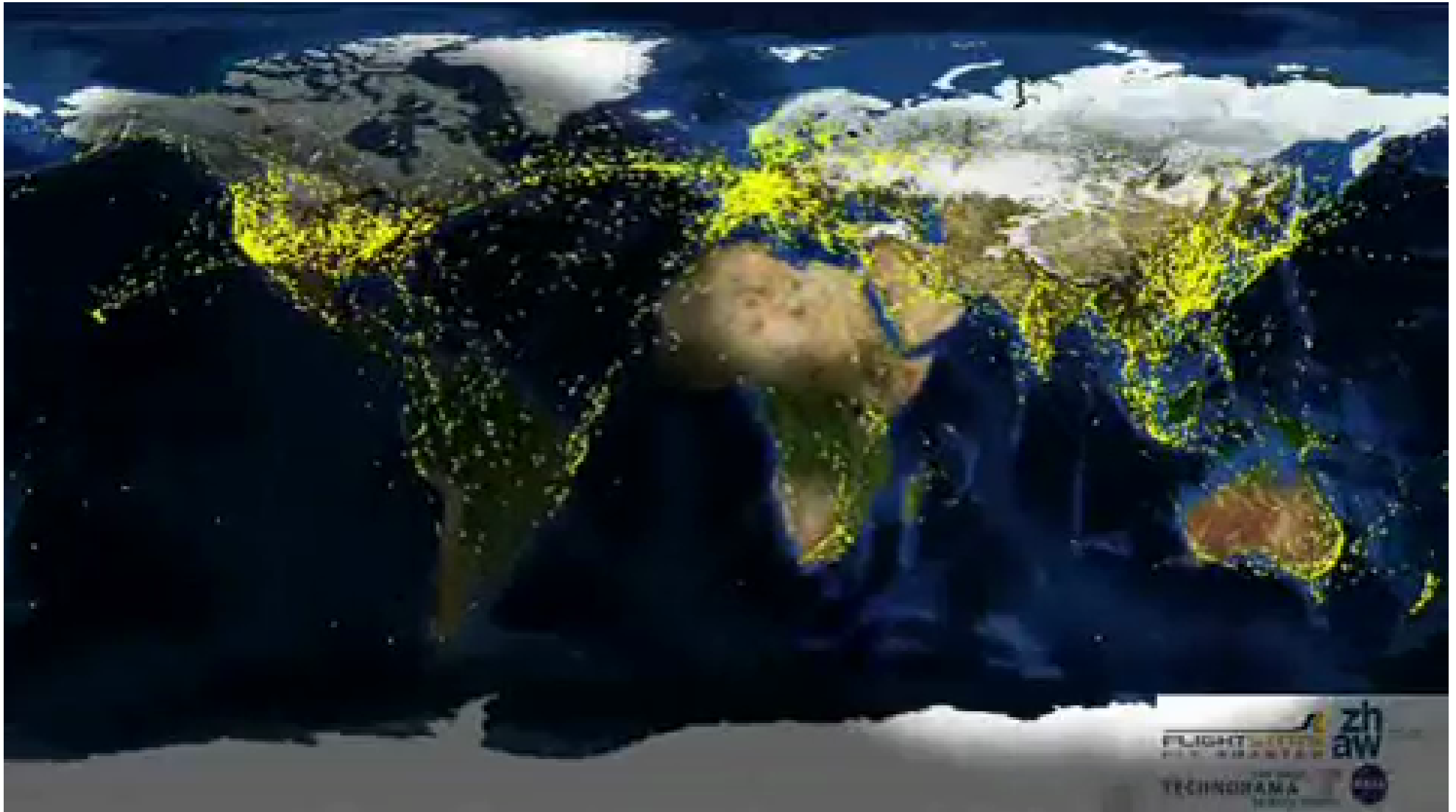
more complex
shapes



more complex
pushing tasks



Intermezzo: Air-Traffic Control

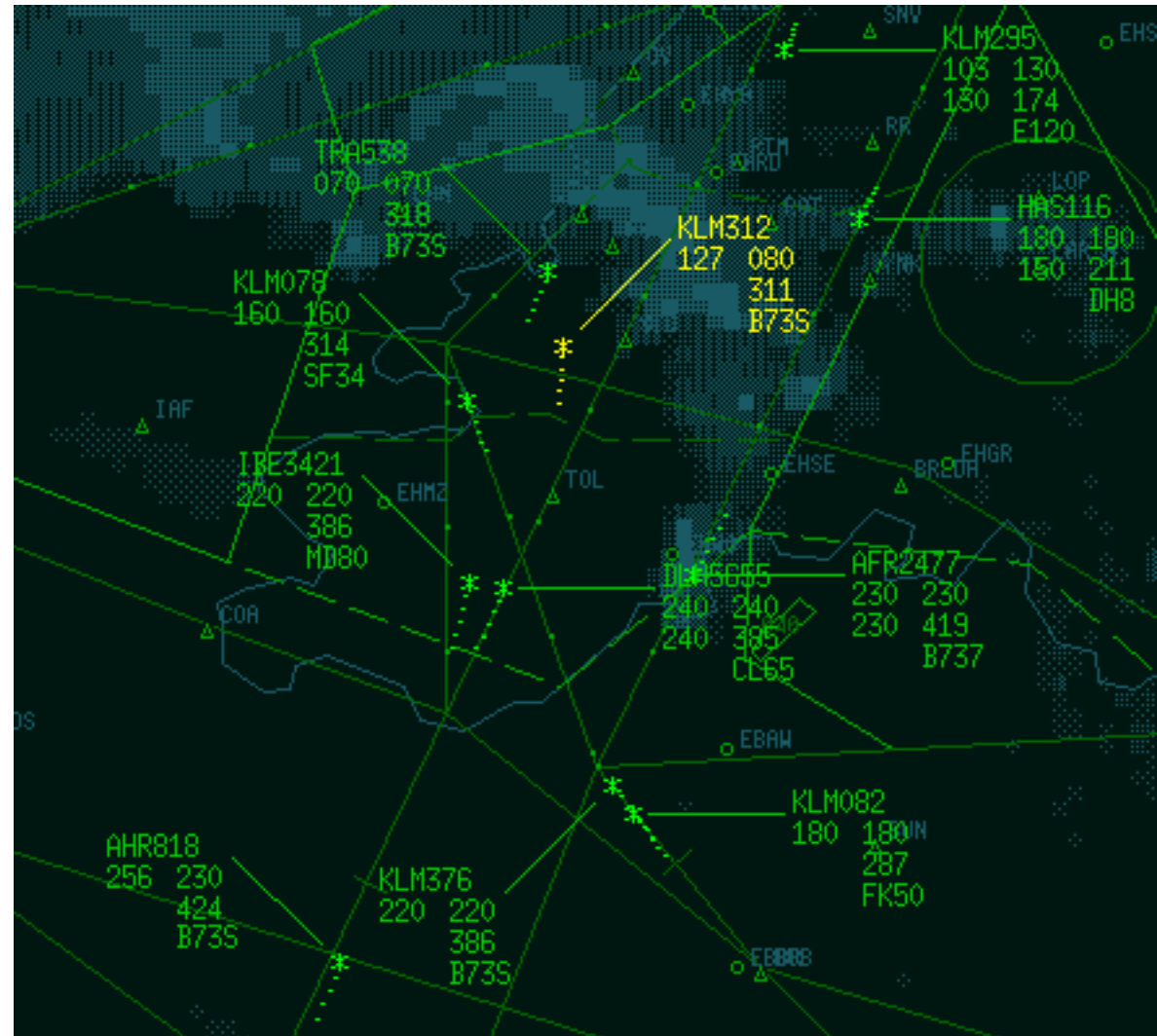


~100,000 flights/day, kept safe by air-traffic controllers.

Intermezzo: Air-Traffic Control

Airplanes → moving points + information labels

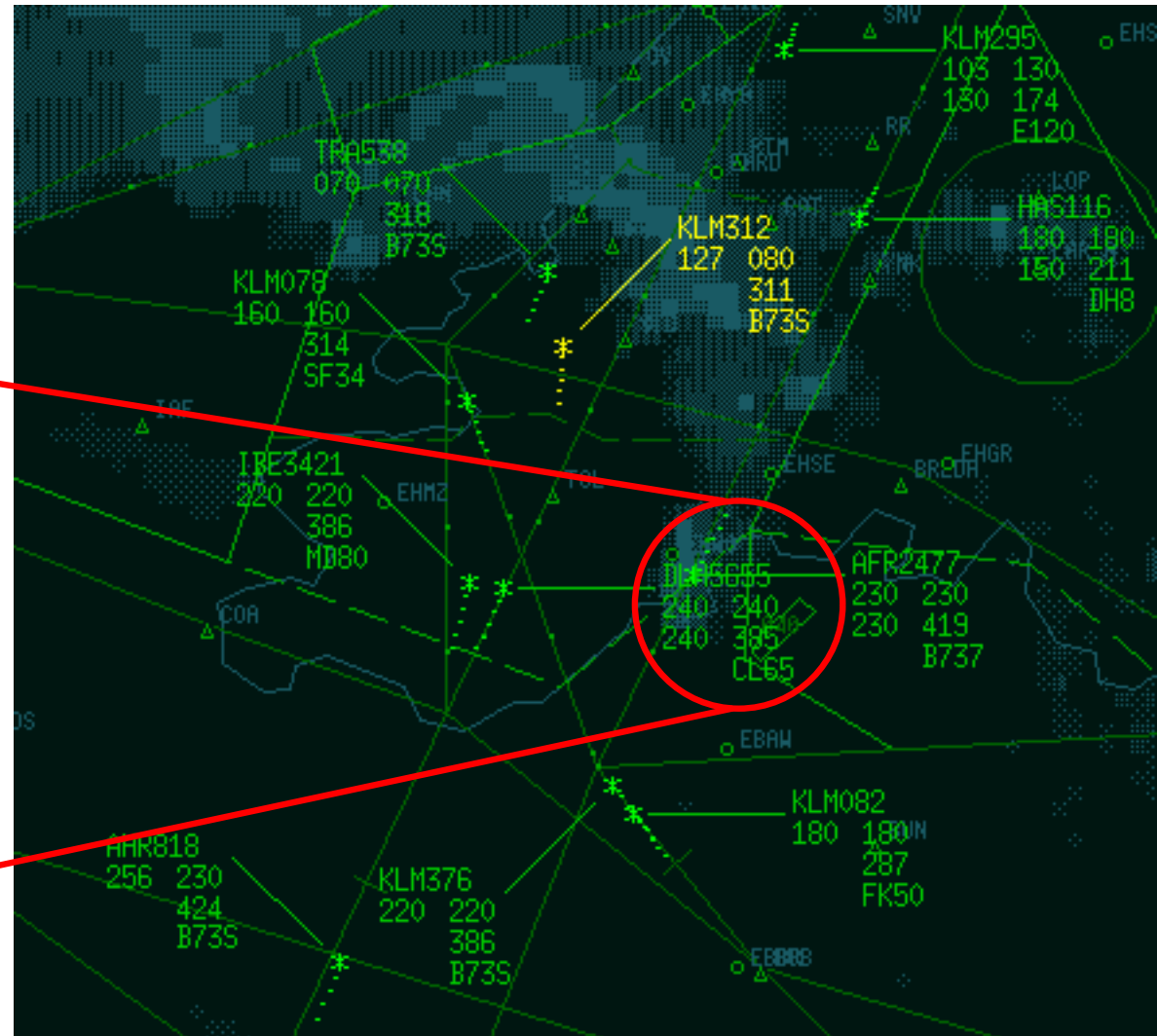
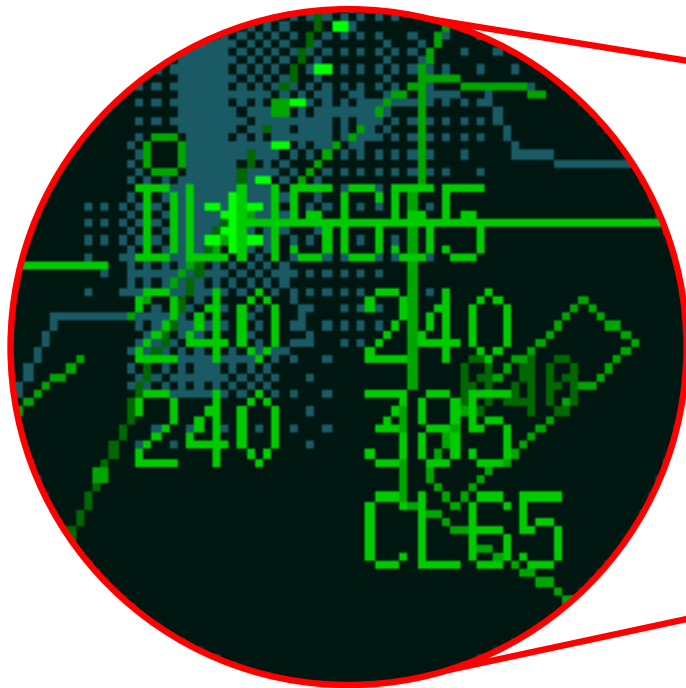
Watch for potential crashes



Intermezzo: Air-Traffic Control

Airplanes → moving points + information labels

Watch for potential crashes

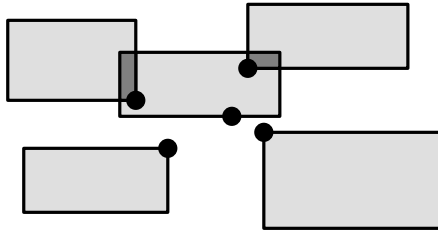


...while moving labels around for readability!

Problem 2: Labeling Moving Points

We want to:

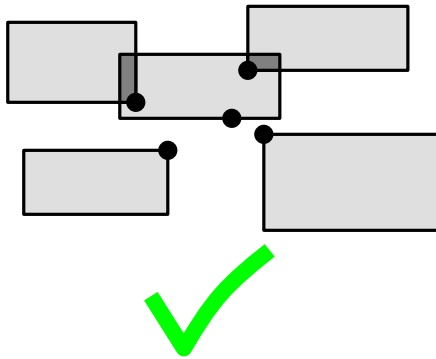
- minimize overlap,
labeling all points



Problem 2: Labeling Moving Points

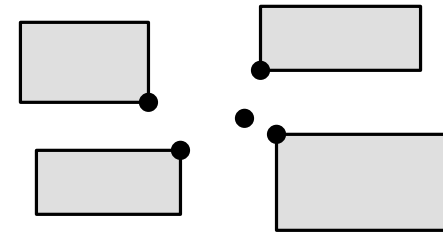
We want to:

- **minimize** overlap, labeling **all** points



We don't want to:

- **avoid** overlap by only labeling **some** points



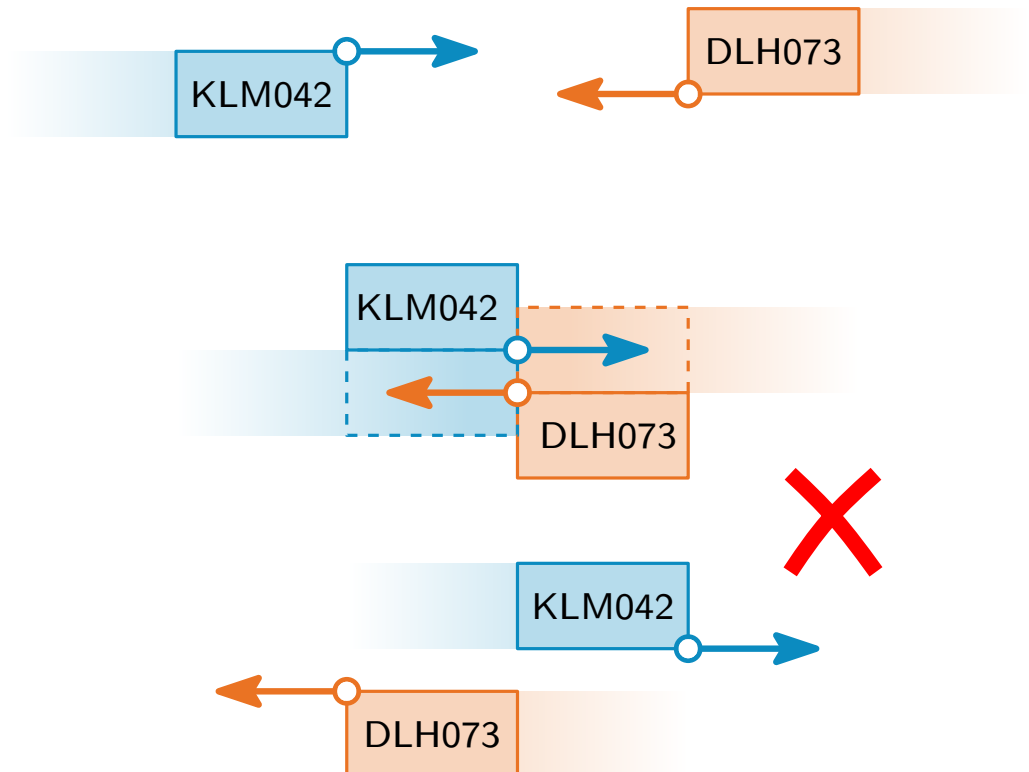
Problem 2: Labeling Moving Points

We want to:

- **minimize** overlap, labeling **all** points

We don't want to:

- **avoid** overlap by only labeling **some** points
- relabel abruptly



Problem 2: Labeling Moving Points

We want to:

- **minimize** overlap, labeling **all** points
- move labels smoothly

We don't want to:

- **avoid** overlap by only labeling **some** points
- relabel abruptly

