Rozpoznávanie chodcov na báze agentovo-orientovaného programovania

Andrej Lúčny

MicroStep-MIS & KAI FMFI UK

andy@microstep-mis.com

http://www.microstep-mis.com/~andy

Automobil je vlastne zlé pomenovanie. Auto síce netreba ťahať, ale treba ho riadiť a preto nazvať ho „samochod“ bolo dosť odvážne (zaslúži si skôr názov „bezkoňochod“ = sans-cheval-mobil). Nepochybne ale príde doba keď autá nebudú potrebovať ani vodiča.


hoci aj renomovaní tvorcovia sci-fi to asi pokladajú za náročnejšiu úlohu než skonštruovať androida-gigola. Tak zlé to ale hádam nebude...

... a ani nebude. Takmer každá automobilka už disponuje vlastným výskumným programom, ktorý je zameraný na čiastočné ciele tejto úlohy.

vedie k tomu hlavne znižovanie ceny senzorov, napr. digitálnych kamier.

Mali sme príležitosť zapojiť sa do projektu DENSO CORPORATION v ktorom išlo o použitie jednej digitálnej kamery na rozpoznávanie chodcov z auta s dôrazom na riešenie problému zákrytu s cieľom predikcie nebezpečných situácii.

Biomimetický prístup

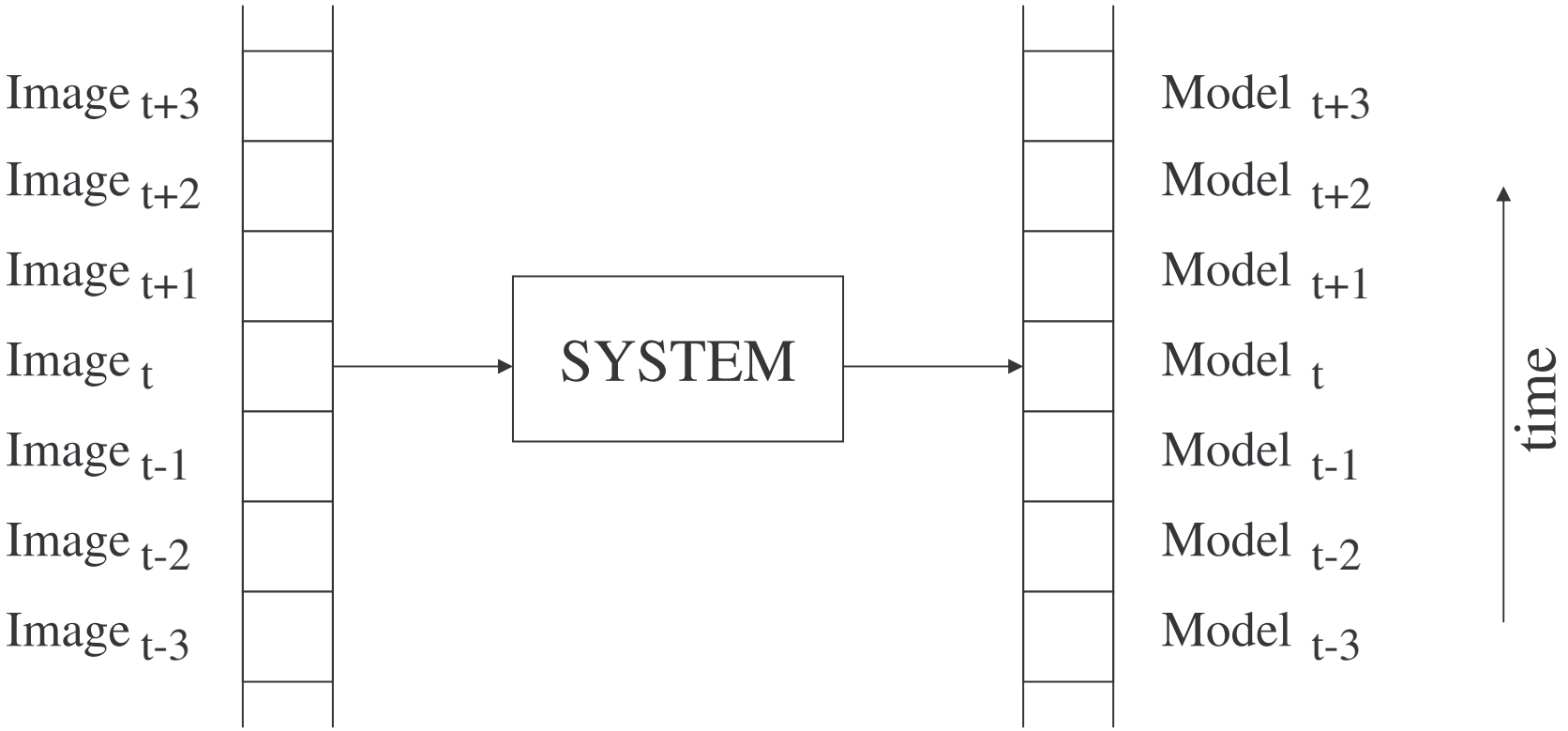
snažili sme sa problém riešiť spôsobom, akým si predstavujeme, že sa rieši v našej mysli

- komplexnosť riešenia
- práca v reálnom čase
- predstava o mysli
- technológia na modelovanie tejto predstavy

Komplexnost’


```
<recognition>
  <object type=‘pedestrian’ key=‘132’>
    <rectangle2D x1=‘522’ y1=‘290’ x2=‘551’ y2=‘376’/>
    <speed2D dx=‘0’ dy=0’/>
    <position3D x=‘2.12’ y=‘0’ z=‘8.54’/>
    <size3D dy=‘1.67’/>
    <speed3D dx=‘0’ dy=‘0’ dz=‘0’/>
  </object>
</recognition>
```

Práca v reálnom čase



Predstava o mysli a jej modelovanie

- za predstavu o mysli sme zvolili Minského societný model mysle
- a za technológiu na jeho implementáciu agentovo-orientované programovanie (programovanie motivované multiagentovými systémami)

Multi-agentové systémy

oblasť zaoberajúca systémami, ktoré sa vyznačujú špecifickým druhom modularity

Táto modularita je postavená na distribúcii a decentralizácii. Systém sa tu skladá z modulov, ktoré nazývame agentmi. Agent sa vyznačuje schopnosťou:

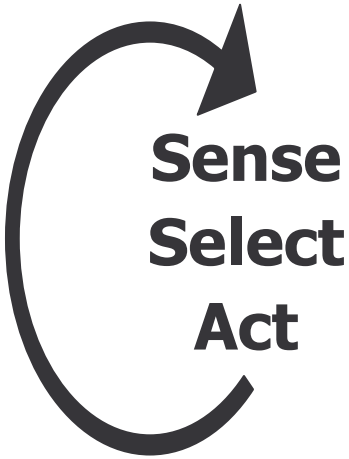
- konať nezávisle na ostatných agentoch
- komunikovať s nimi spôsobom, ktorý túto autonómnosť nenarúša

Multi-agentové systémy (MAS)

- Typickou ukážkou MAS je napr. robosoccer: Aký program vložiť do jednotlivých robotov aby vyhrali zápas ?

Povaha agenta

- Všimnime si, že do takéhoto robota budeme vkladať program tvaru



- kód agenta je vykonávaný v jeho vlastnom vlákne
- vlákno je potrebné blokovať (časovač, spúšťač)

„Slabý“ a „silný“ agent

„Inteligencia“ je zabezpečená špeciálnou súčiastkou v agentovi

Silné agenty

(Inteligentné agenty)

(Deliberatívne agenty)

„Inteligencia“ povstáva z interakcie agentov bez špeciálnych súčiastok

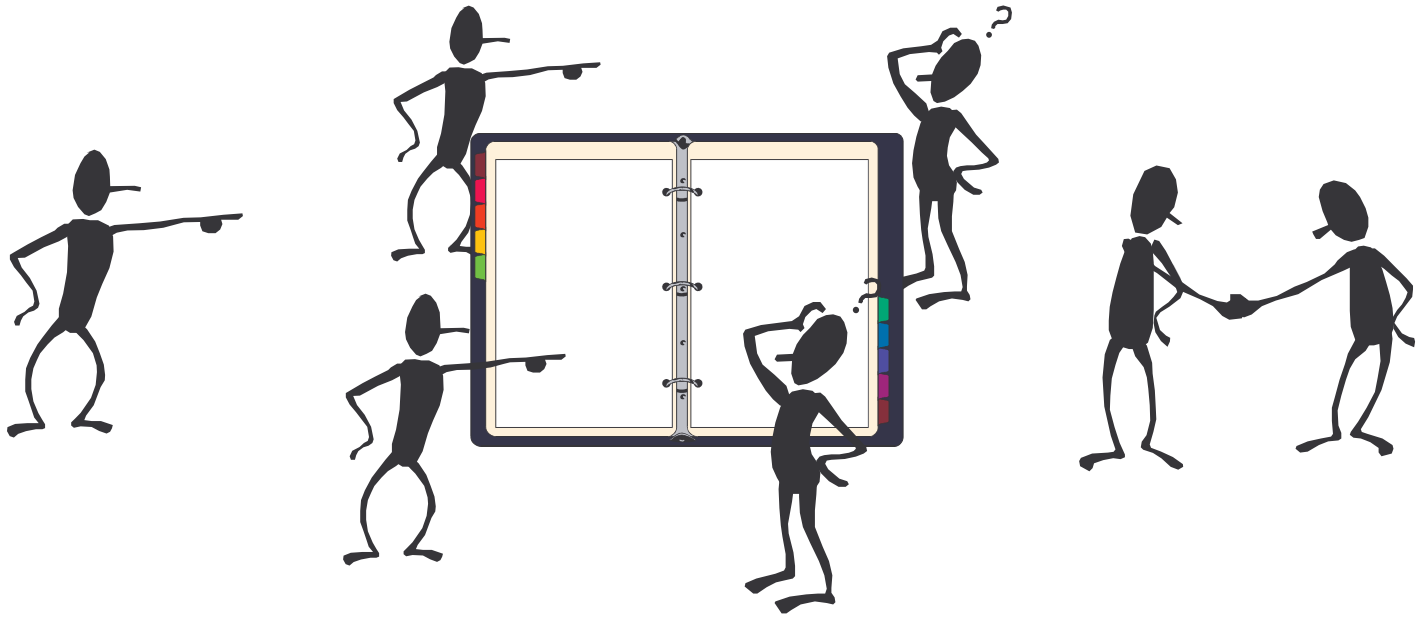
Slabé agenty

(reaktívny agenty)

(emergencia)

Komunikácia medzi agentami

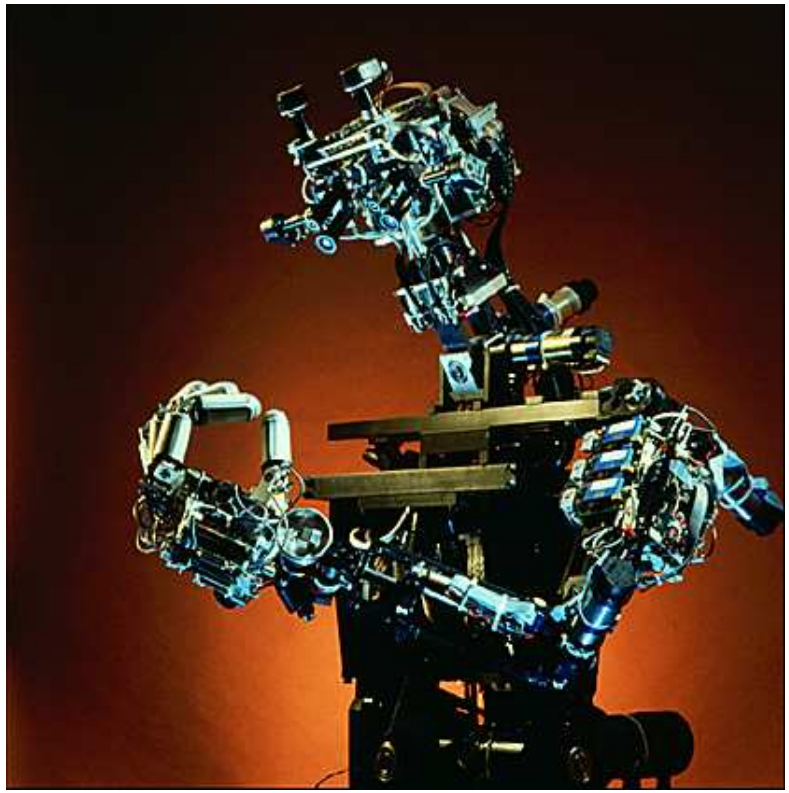
- a že agenty budú vzájomne komunikovať a to buď priamo, alebo nepriamo cez nejakú „čiernu tabuľu“ (blackboard, space)



Agentovo - orientované programovanie (AOP)

- Poznatky o danom druhu modularity získané na takýchto distribuovaných systémoch je možné generalizovať a aplikovať aj pre tvorbu akýchkoľvek systémov.

- Techniky, ktoré toto realizujú nazývame Agentovo-orientovaným programovaním



MAS

Network is
a computer

AOP

Computer is
a network

Agent-Space

- z možných implementácii AOP sme si samozrejme:) zvolili tú svoju a to architektúru agent-space. Vyznačuje sa tým, že stavia na slabých agentoch a nepriamej komunikácii v reálnom čase, pričom nastavuje detaily tejto komunikácie natoľko šikovne, že komunikáciu medzi dvomi agentami možno vždy infiltrovať iným agentom. Vďaka tomu je možné vyvíjať systém inkrementálne zdola – nahor (Brooks).

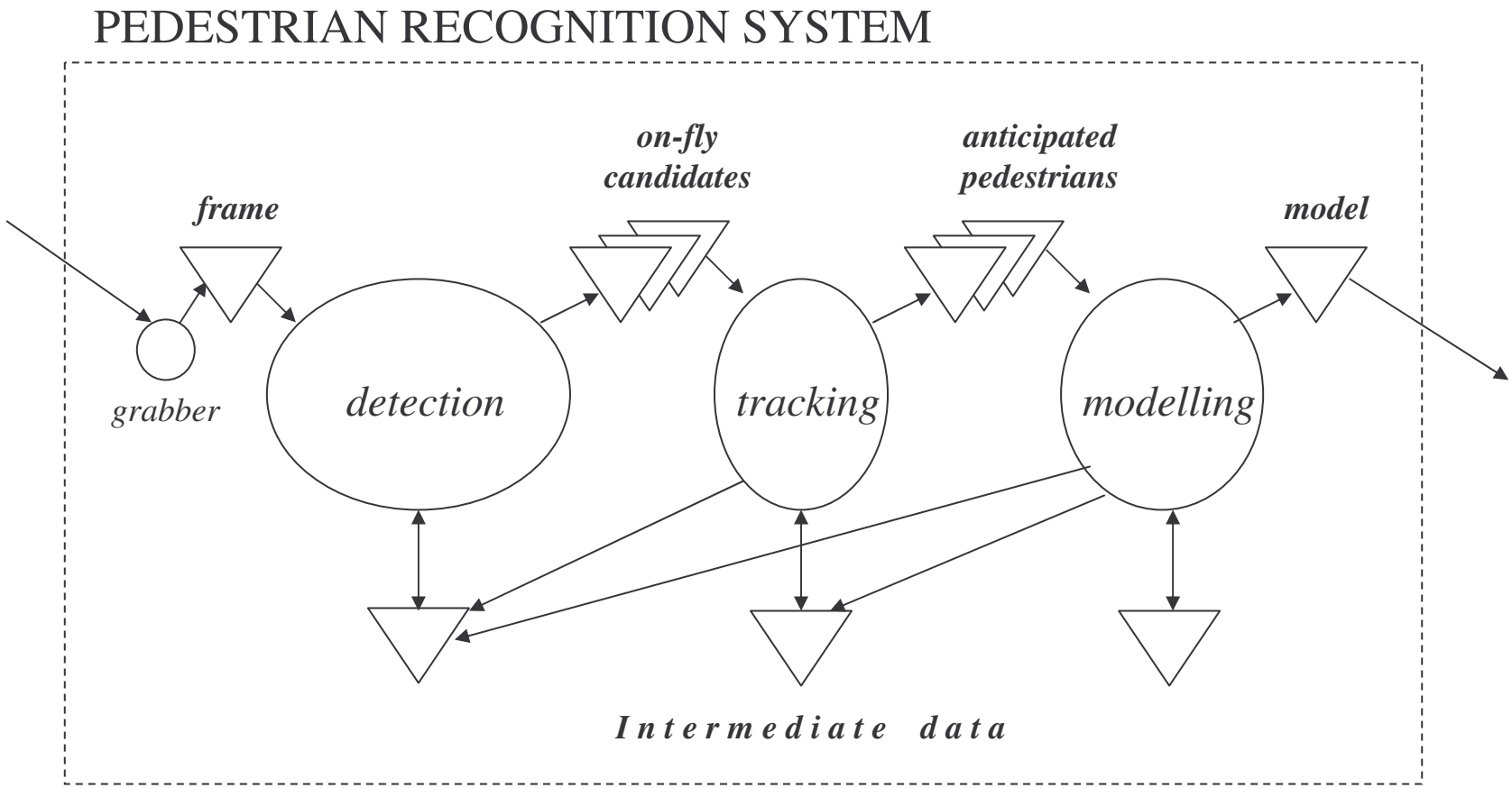
Ukážka

```
package com.microstepmis.agentspace.demo;
import com.microstepmis.agentspace.*;

public class Agent1 extends Agent {

  int i = 0;

  public void init(String[] args) {
    attachTimer(1000);
  }

  public void senseSelectAct() {
    System.out.println("write: "+i);
    write("a",i++);
  }

}
```

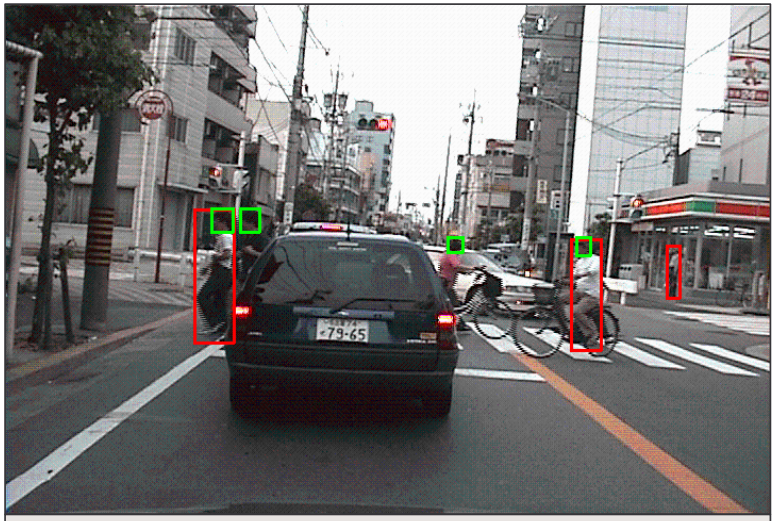
```
public class Agent2 extends Agent {

  int i;

  public void init(String args[]) {
    attachTrigger("a");
  }

  public void senseSelectAct() {
    i =  (Integer) read("a",-1);
    System.out.println("read "+i);
  }

}
```

```
public class Starter {
  public static void main(String[] args) {
    new SchdProcess("space","com.microstepmis.agentspace.SpaceFactory",new String[]{"DATA"});
    new SchdProcess("agent1","com.microstepmis.agentspace.demo.Agent1",new String[]{});
    new SchdProcess("agent2","com.microstepmis.agentspace.demo.Agent2", new String[]{});
  }
}
```

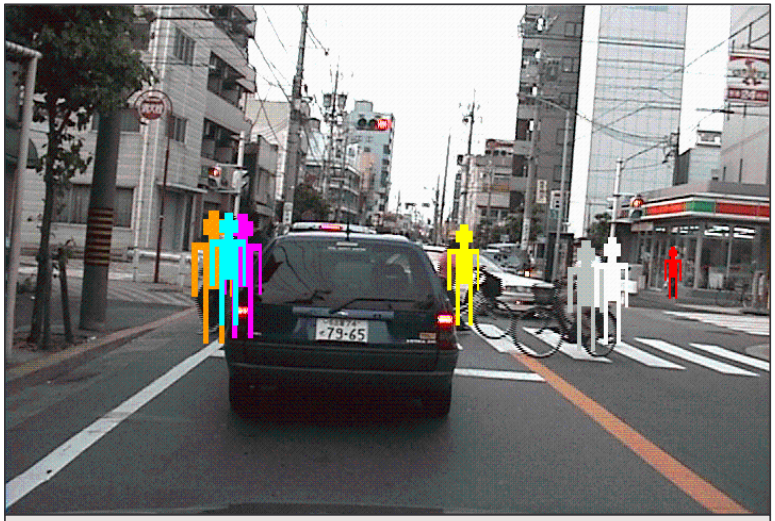
Rozpoznávanie chodcov



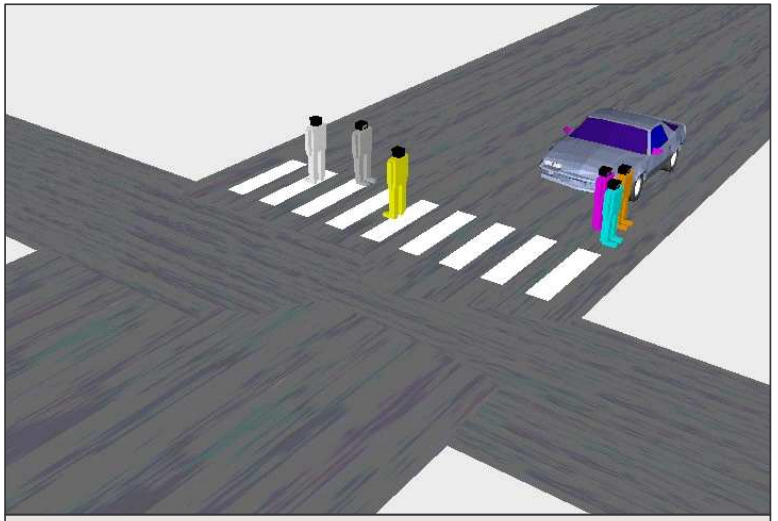
raw image



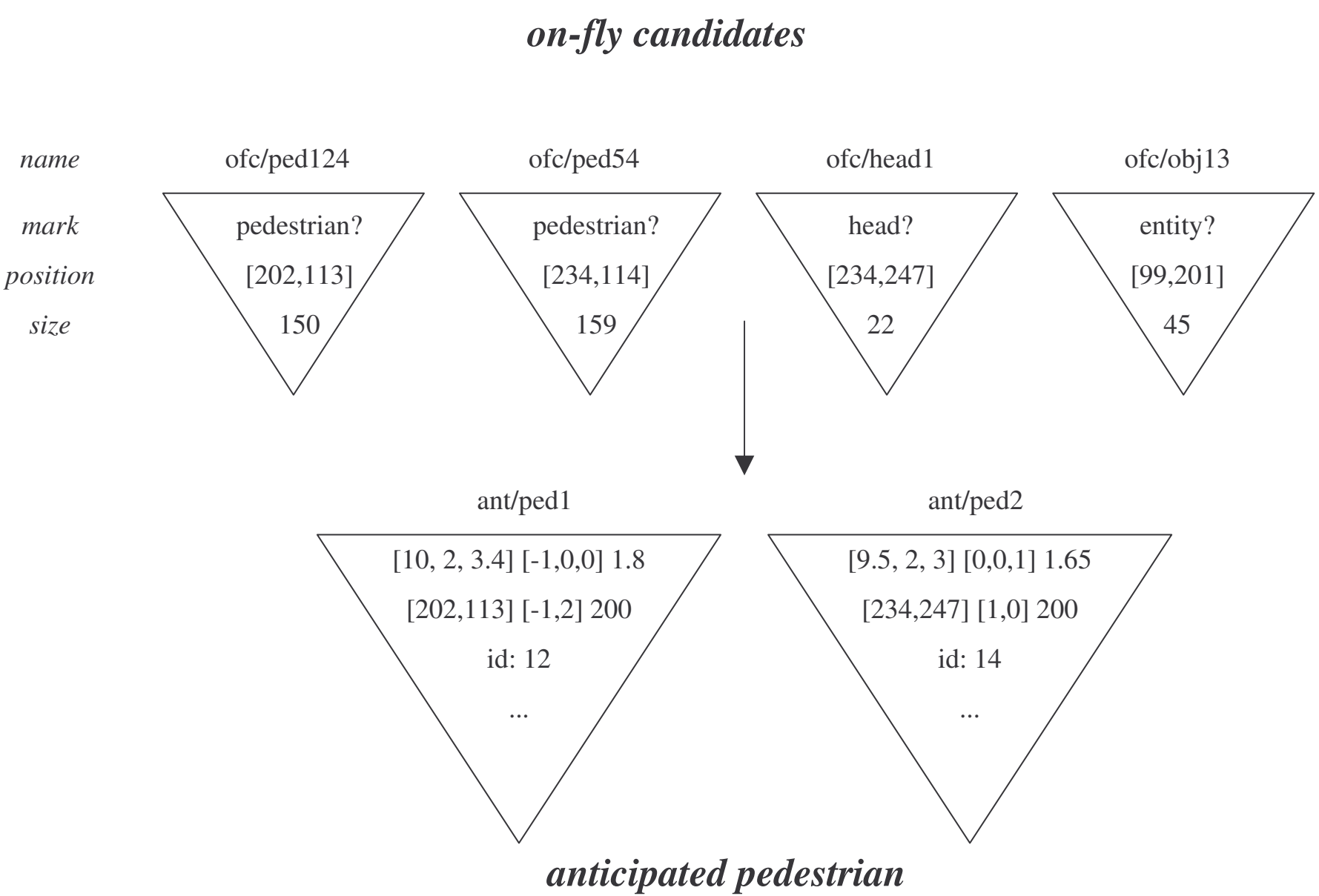
on-fly candidates (pedestrians, heads)

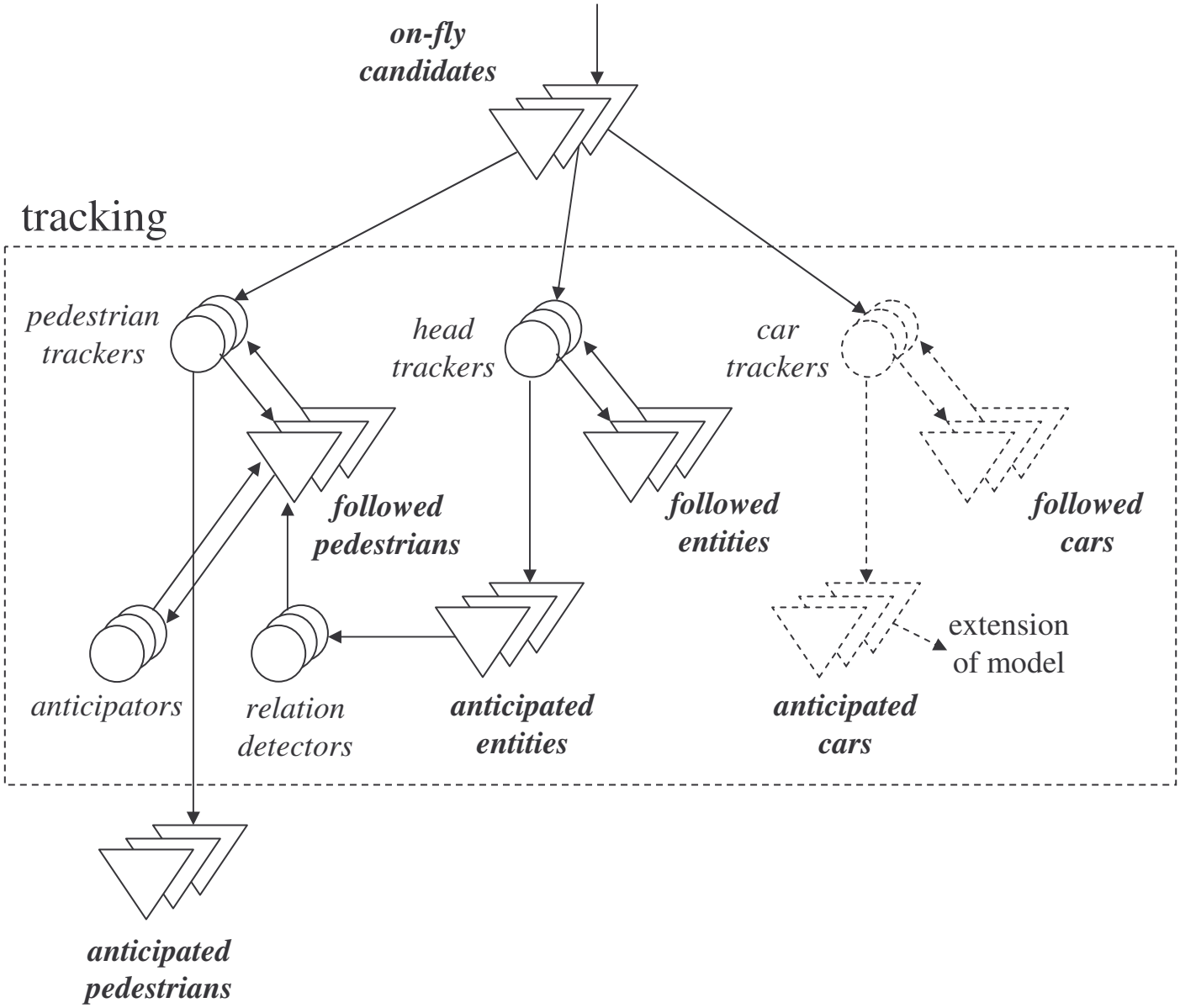


anticipated pedestrians



model


on-fly candidates
name
mark
position
size
ofc/ped124
pedestrian?
[202,113]
150
ofc/ped54
pedestrian?
[234,114]
159
ofc/head1
head?
[234,247]
22
ofc/obj13
entity?
[99,201]
45
ant/ped1
[10, 2, 3.4] [-1,0,0] 1.8
[202,113] [-1,2] 200
id: 12
...
ant/ped2
[9.5, 2, 3] [0,0,1] 1.65
[234,247] [1,0] 200
id: 14
...
anticipated pedestrian


on-fly
candidates
tracking
pedestrian
trackers
head
trackers
car
trackers
followed
pedestrians
followed
entities
followed
cars
anticipators
relation
detectors
anticipated
entities
extension
of model
anticipated
cars
anticipated
pedestrians

Výsledky

- video
- nenechajte sa oklamať zdanlivou kvalitou:
  - on-fly kandidáti sú v ňom ručne vybraní
  - konštanty sú nastavené pre konkrétne video
  - čas je 1000 krát spomalený
  - chodci v scéne sú dostatočne rozdielni

Ďakujem za pozornosť

Rozpoznávanie chodcov na báze agentovo-orientovaného programovania

Andrej Lúčny

MicroStep-MIS & KAI FMFI UK

andy@microstep-mis.com

http://www.microstep-mis.com/~andy