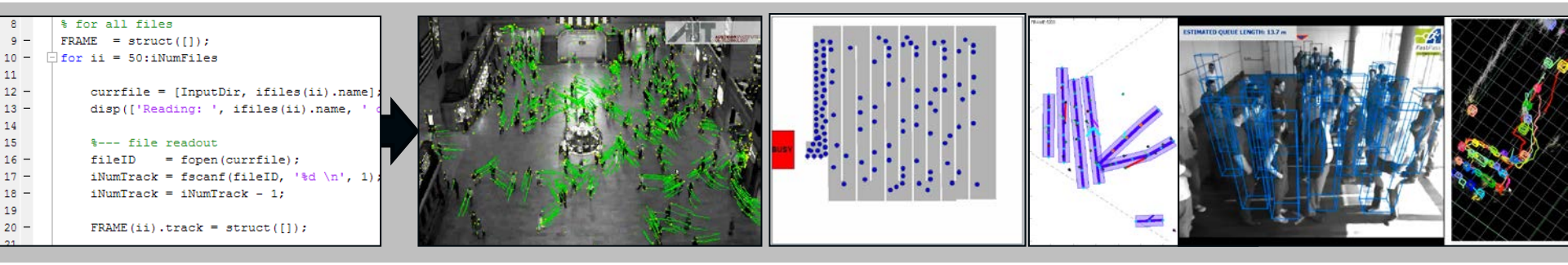
AUSTRIAN INSTITUTE OF TECHNOLOGY
TOMORROW TODAY



Algorithmic development for 2D and 3D vision systems using Matlab

Csaba Beleznai

Csaba Beleznai
Senior Scientist
Video- and Safety Technology
Safety & Security Department
AIT Austrian Institute of Technology GmbH
Vienna, Austria

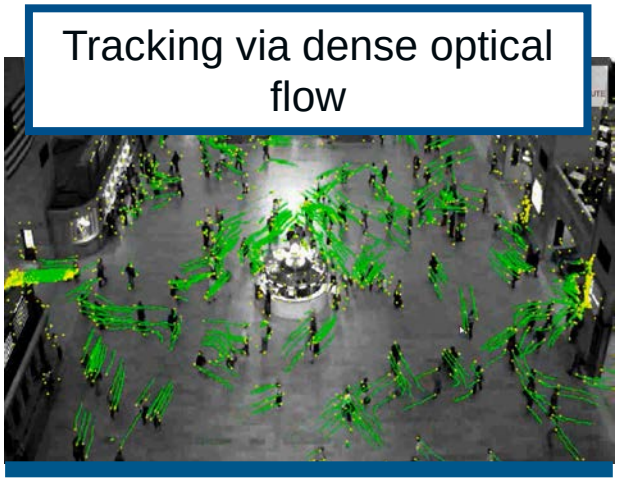
Co-Authors:

Michael Rauter, Christian Zinner, Andreas Zweng, Andreas Zoufal, Julia Simon, Daniel Steininger, Markus Hofstätter und Andreas Kriechbaum

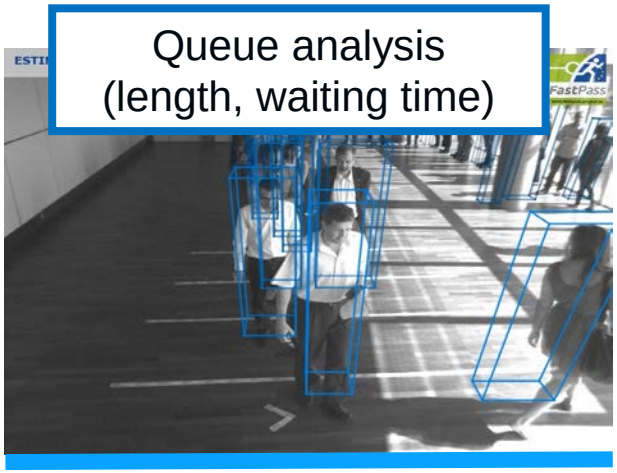
Vision+ FastPass www.fastpass-project.eu D-SenS SEVENTH FRAMEWORK PROGRAMME FFG bmvit

Content

- **Brief intro**- Austrian Institute of Technology
- **Motivation** – Development of complex HW/SW systems
- **Concepts – Interplay between Matlab and C/C++**
  - Matlab →C++ and C++ → Matlab
- **Applied cases: visual analysis of crowding phenomena**



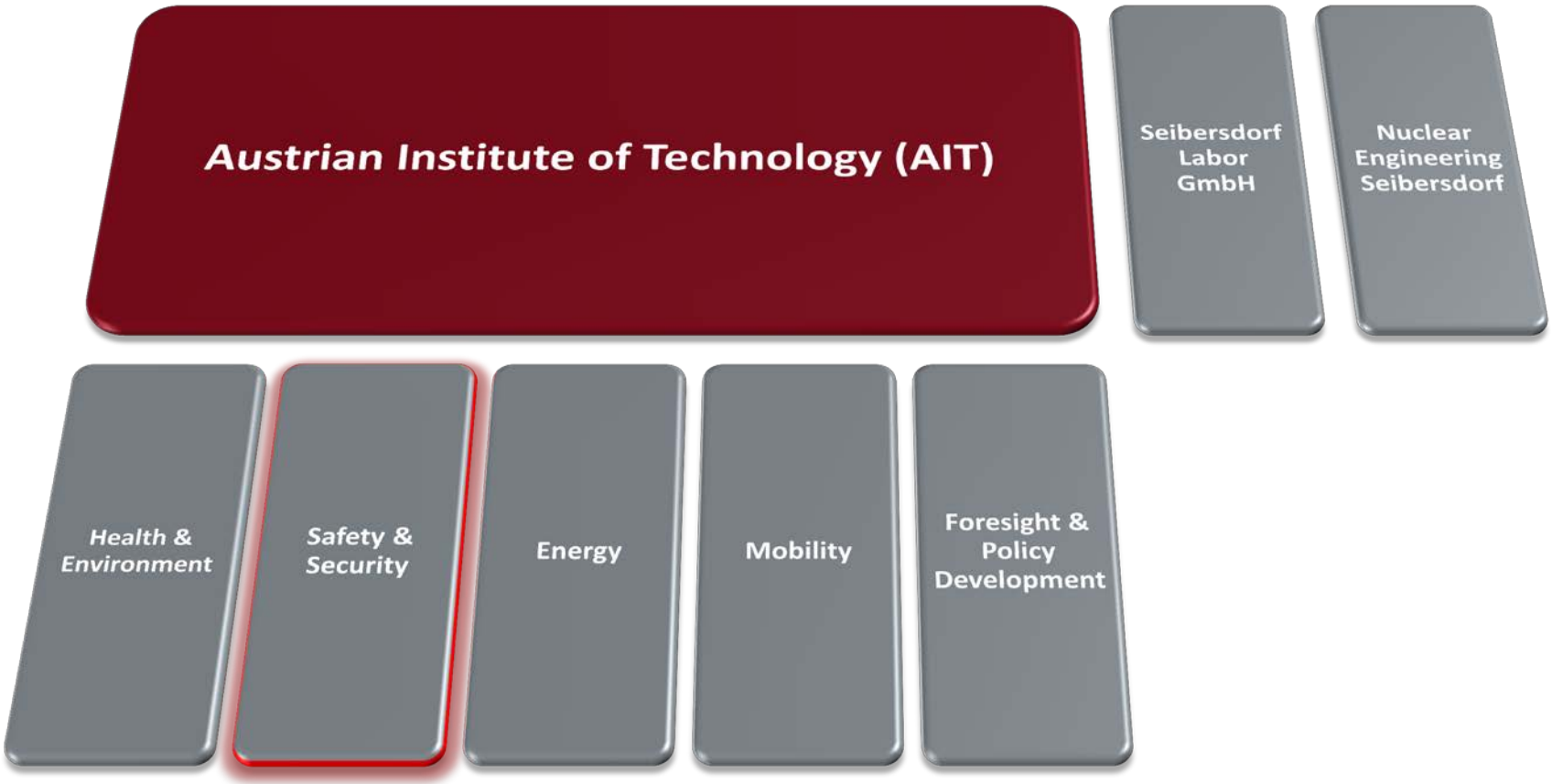
2D



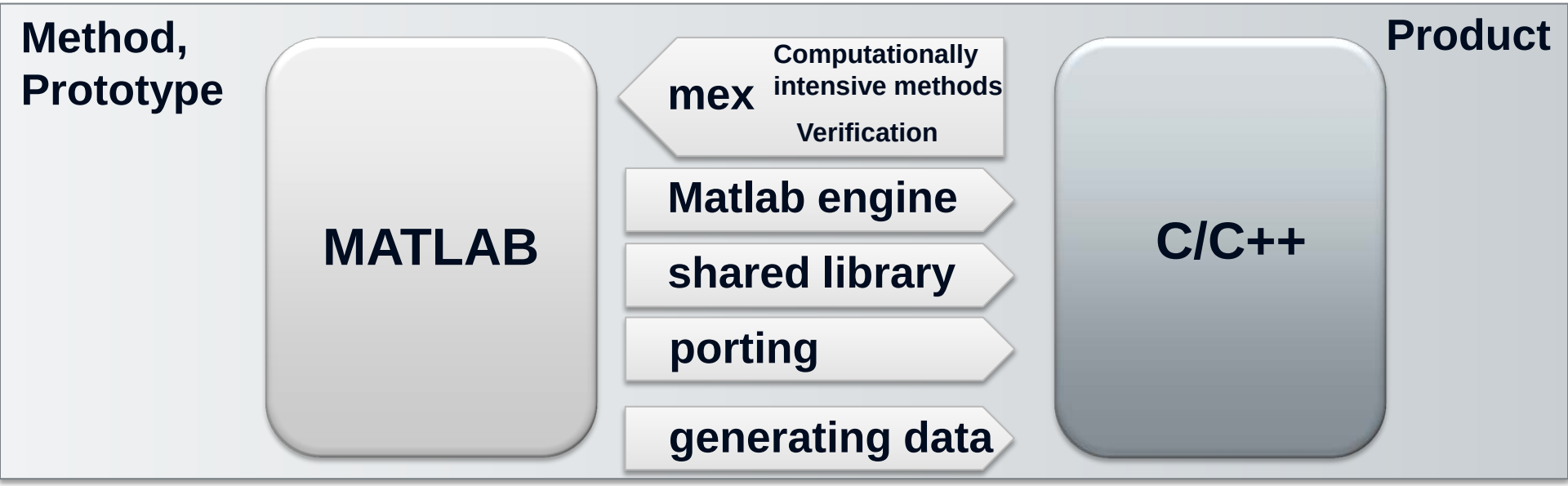
3D

- **Summary**

Introduction



Employed development concept

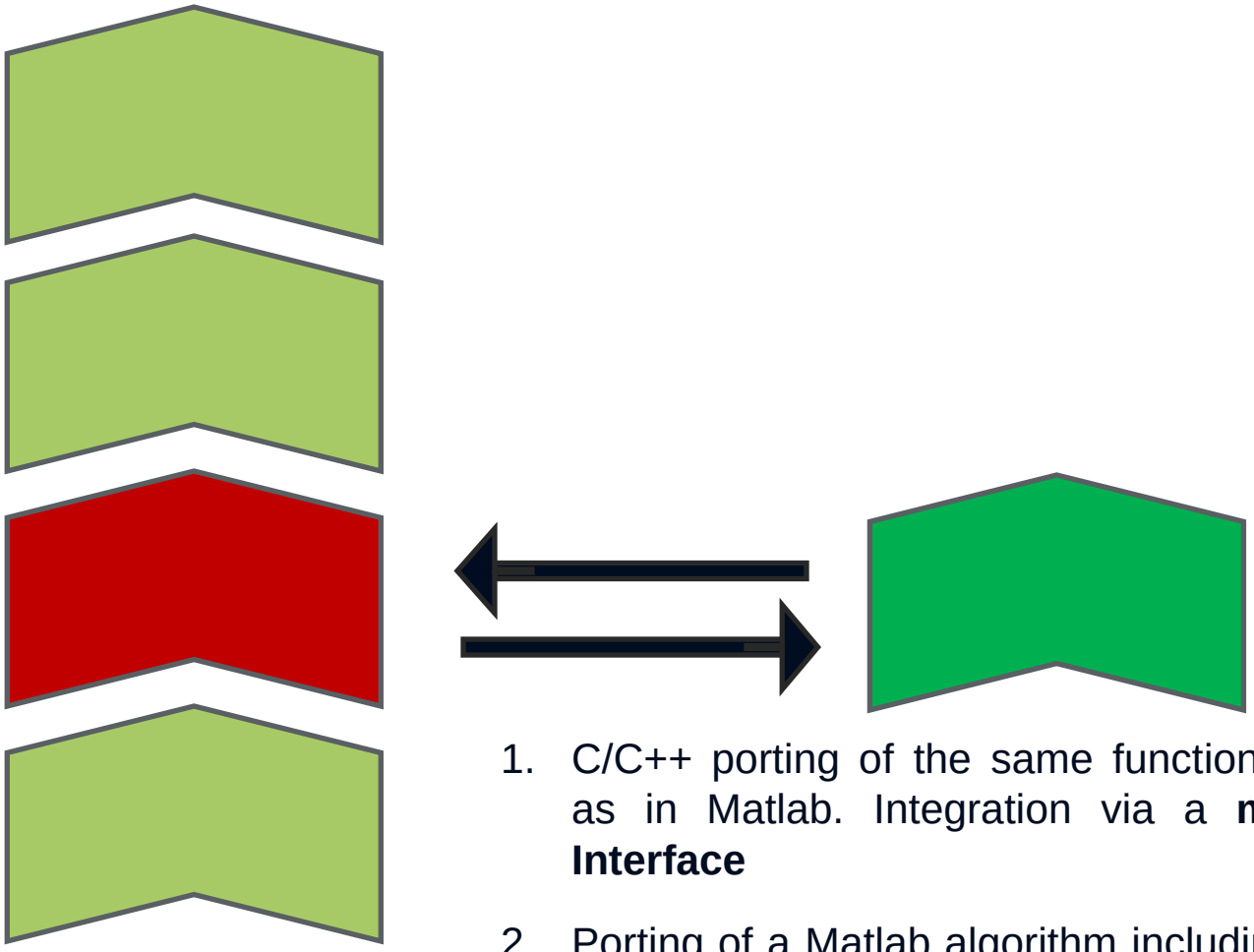


- **MATLAB:**
  - Broad spectrum of algorithmic functionalities,
  - Image analysis prototypes can be done easily and fast,
  - Large set of visualization and debugging options,
  - Rapid development → method, prototype, demonstrator
- **C/C++**
  - Real-time capability

Rapid creation of prototypes / verification

MATLAB **C/C++**

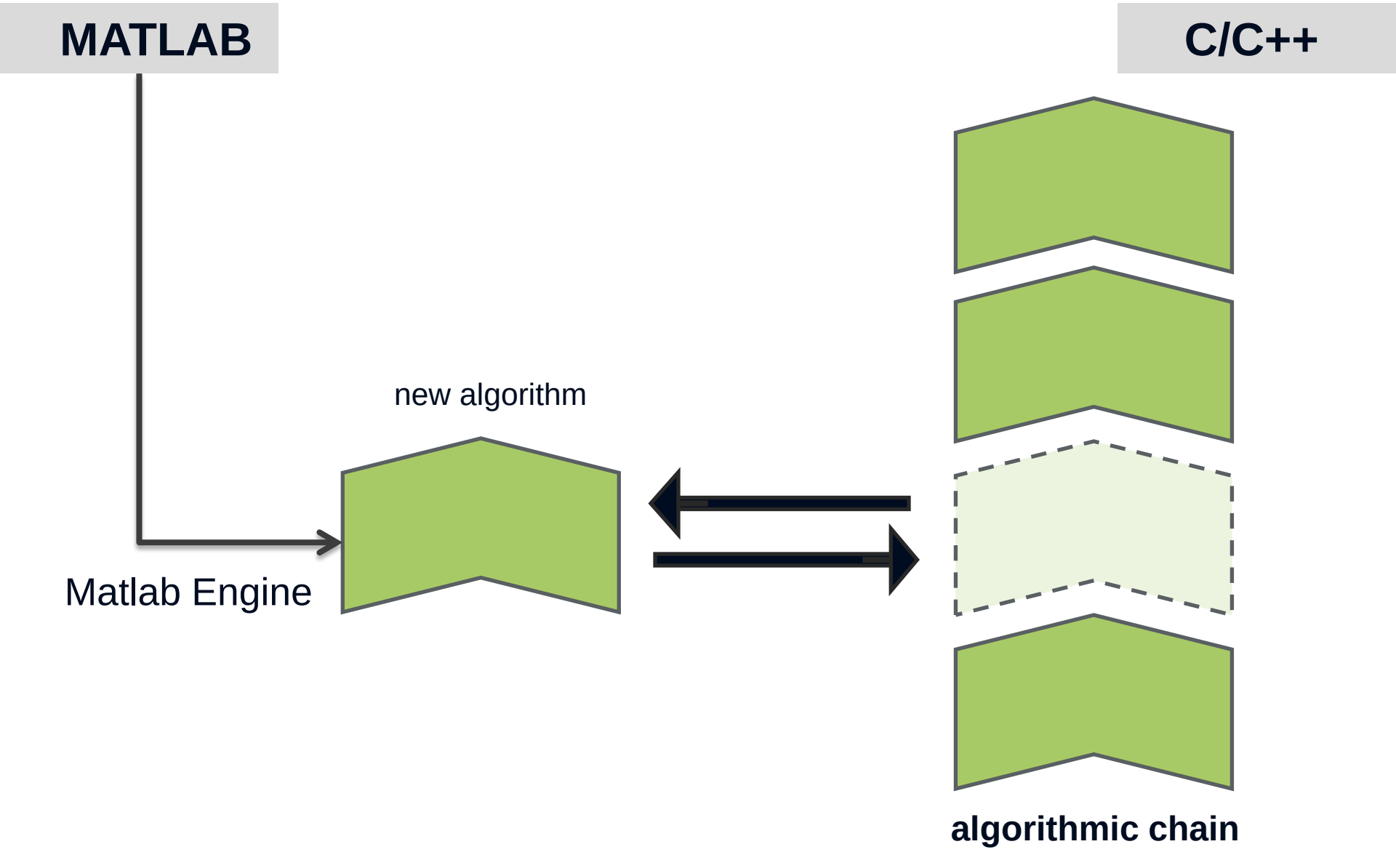
Computationally intensive algorithm or part of an algorithm



algorithmic chain

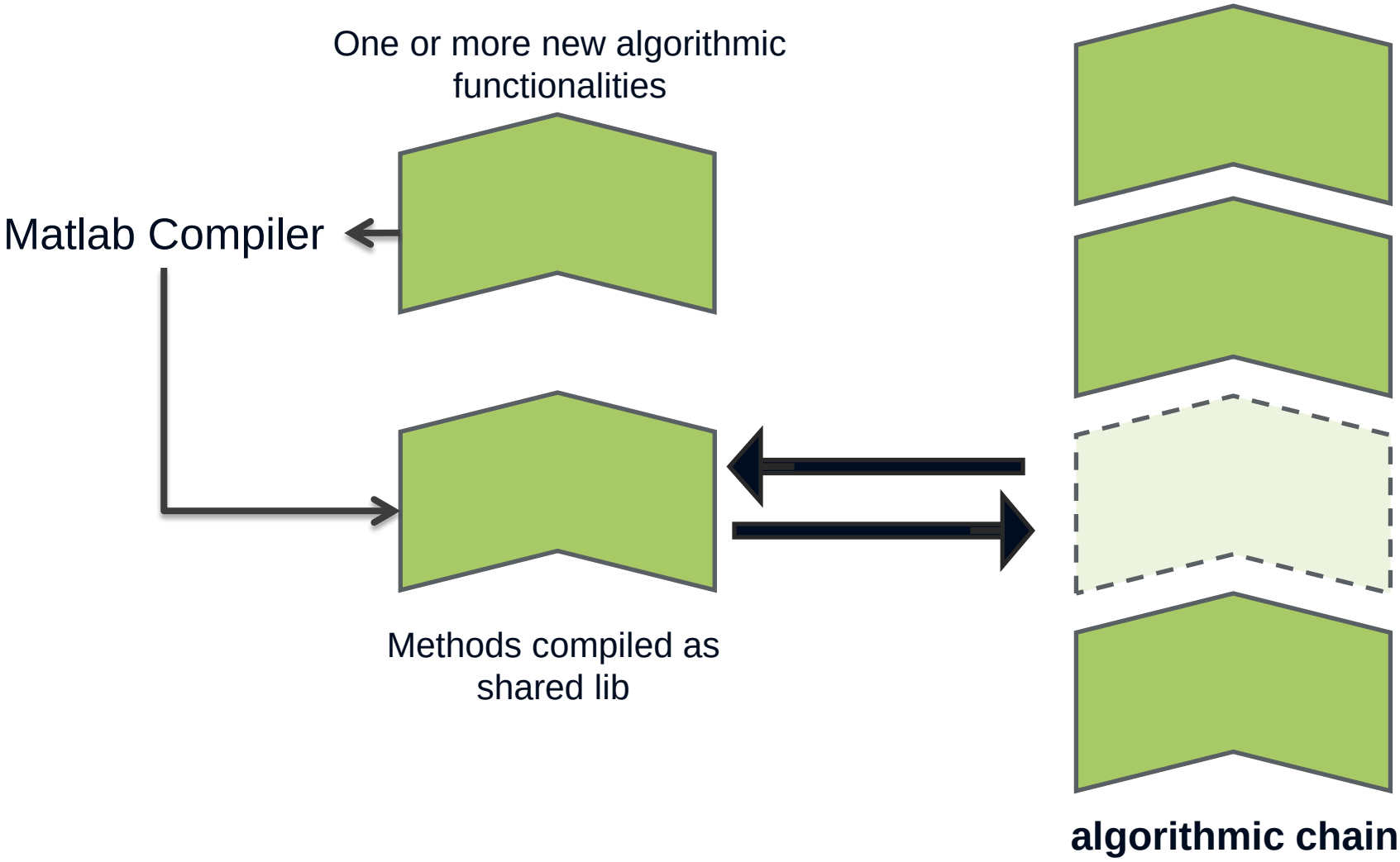
1. C/C++ porting of the same functionality as in Matlab. Integration via a **mex-Interface**
2. Porting of a Matlab algorithm including a **mex-Interface** to verify the functionality in terms of identical results

Fast integration of new algorithms (1)

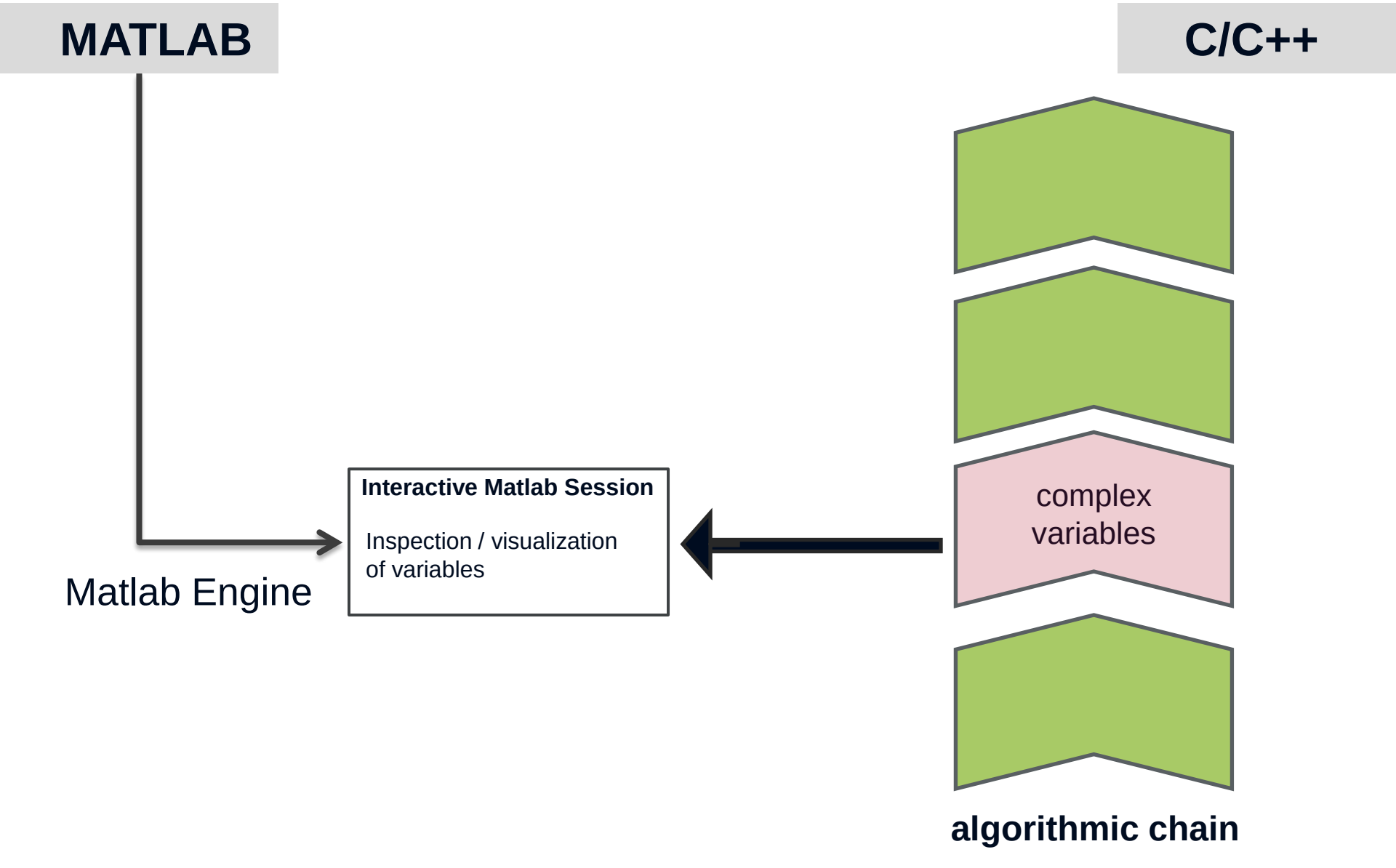


Fast integration of new algorithms (2)

MATLAB **C/C++**



Matlab Engine supporting C/C++ Debugging



Visual Surveillance - Motivating example

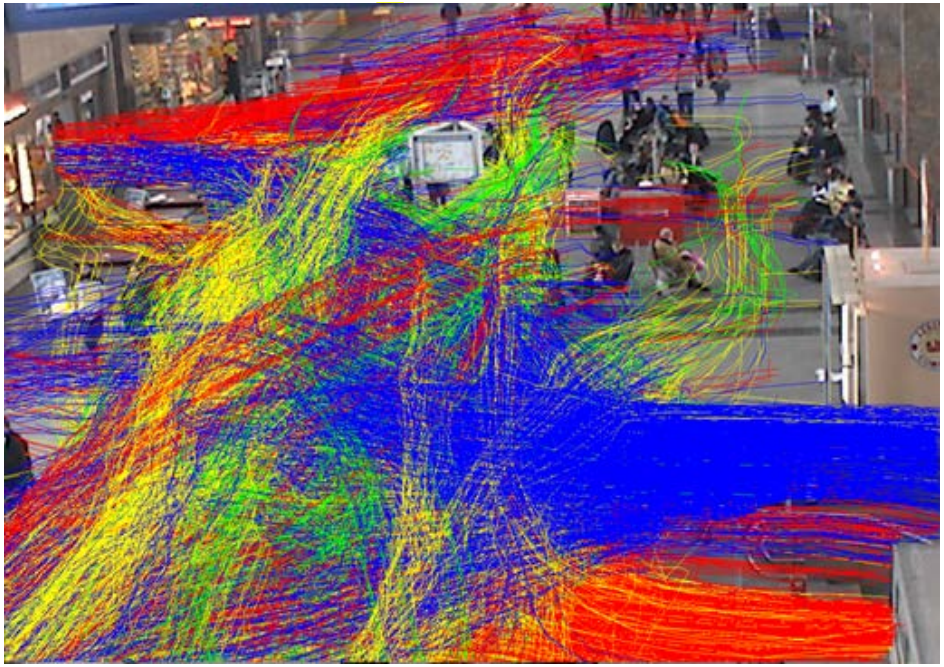
Algorithmic units:

- Object detection and classification
- Tracking
- Activity recognition

Typical surveillance scenario:

Who : people, vehicle, objects, …
Where is their location, movement?
What is the activity?
When does an action occur?

Visual Surveillance - Motivating example



Typical surveillance scenario:

Who : people, vehicle, objects, …
Where is their location, movement?
What is the activity?
When does an action occur?

Algorithmic units:

- Object detection and classification
  - Counting, Queue length, Density, Overcrowding
  - Abandoned objects
  - Intruders
- Tracking
  - Single objects
  - Video search
  - Flow
- Activity recognition
  - Near-field (articulation)
  - Far-field (motion path)

AUSTRIAN INSTITUTE OF TECHNOLOGY

Visual analysis of pedestrian flows

Number, position, dynamics - Quantities characterizing the context and behavior of/in the scene **OUTPUTS**

HIGH-LEVEL

Estimating characteristic quantities

Characteriztaion of dynamics

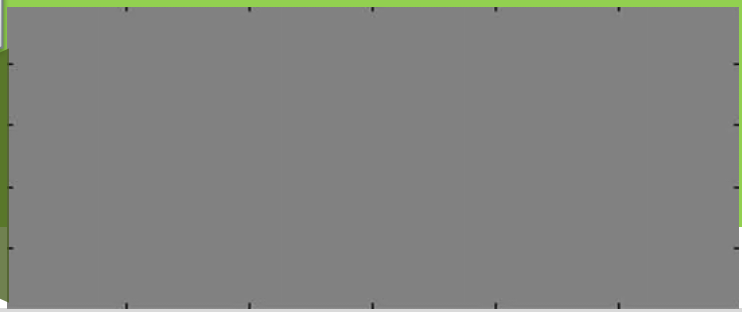
Crowd segmentation

LOW-, MID-LEVEL

Dense motion estimation

Tracking

Human detection



Intensity/RGB image

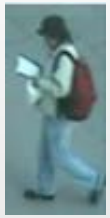
Disparity (depth) image + parameters

INPUTS

Matlab → C++: Matlab-Engine

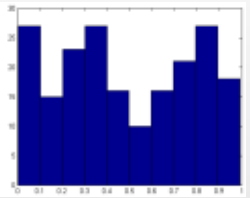
official example: *engdemo.c*

Sample computation:



MATLAB Function

function [DescrTempl] = ComputeDescr(iminTempl, Params)



Input: Image + Parameter

Output: Signature (e.g. histogram)

- MATLAB operates in the background as a powerful programmable algorithmic library

```
#include "engine.h"                              // including the Matlab engine
Engine   *ep;                                   // instancing the Matlab engine
//======== 1. Initializing the Matlab engine ==============================
   if (!(ep = engOpen("\0")))
        return STATUS_MATLAB_INIT_ERROR;    // otherwise return error code

engPutVariable(ep, "Params", mxParams);   // Place variable Params into the MATLAB workspace
engPutVariable(ep, "iminTempl", mxImT);   // Inserting image data into Matlab

// Evaluating the expression in Matlab
engEvalString(ep, "DescrTempl = ComputeDescr(iminTempl, Params);");

// Deallocating Matlab-specific C-variables
mxDestroyArray(mxParams); mxParams = NULL;
mxDestroyArray(mxImT);    mxImT = NULL;
// closing the Matlab engine
engClose(ep);
```

Matlab → C++: shared library

- Shared Library:
  Set of functions loaded into a C/C++ application during run-time dynamically
- MATLAB code → MATLAB compiler → shared library

Compiler call:

```
mcc -W lib:matchlib -T link:lib ComputeDescr.m
```

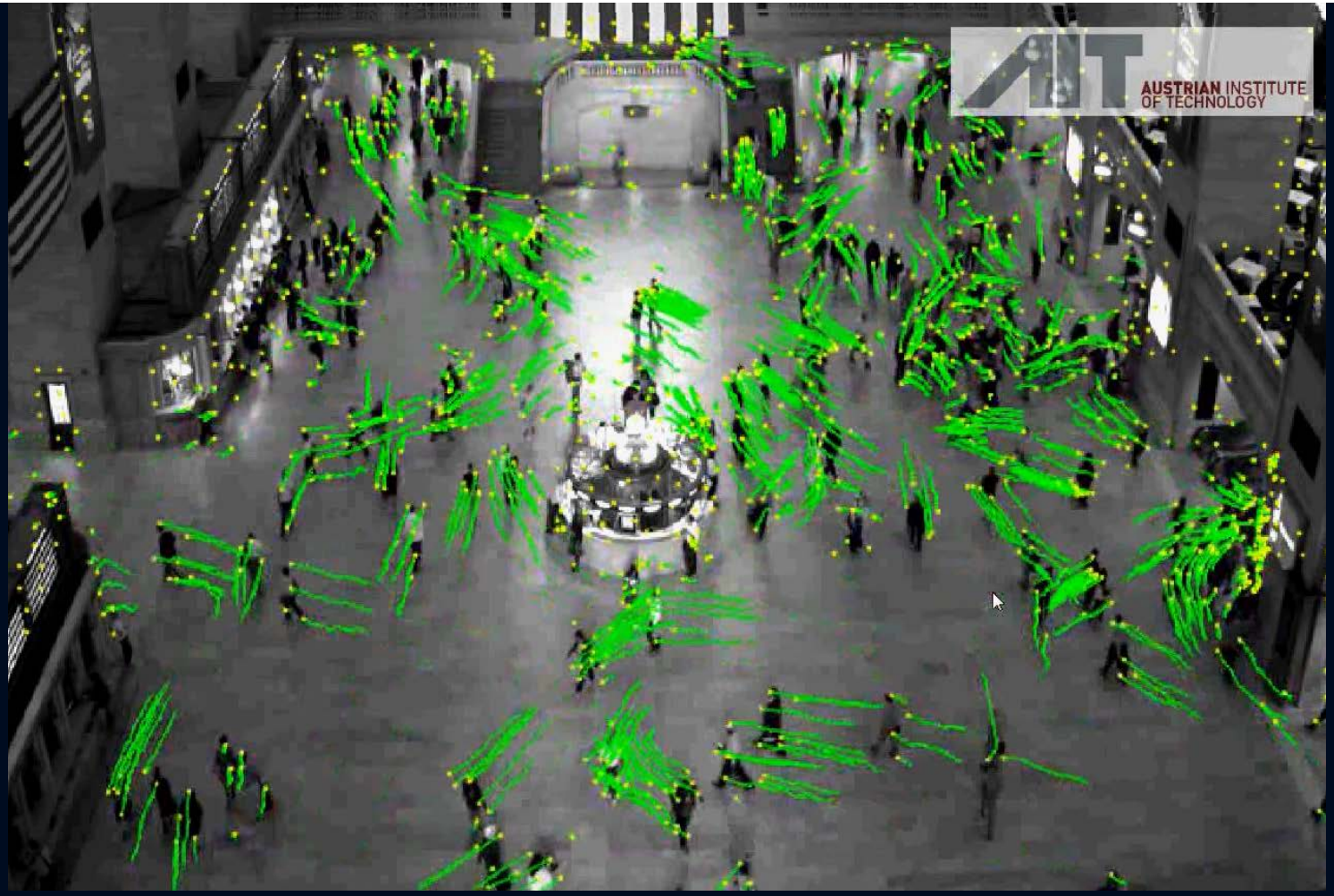
```
#include "matchlib.h"          // Compiled interface of Matlab code

//============== 1. MCR and library initialization functions ========
if( !mclInitializeApplication(NULL, 0) )
{
     fprintf(stderr, "Could not initialize the application.\n");
     exit(1);
}
if (!matchlibInitialize())
{
    fprintf(stderr, "Could not initialize the library.\n");
    exit(1);
}

// compiled function call
mlfComputeDescr(1, &mxDescrT, mxImT, mxParams);// first argument is the number of outputs

matchlibTerminate();               // library termination
mclTerminateApplication();         // application-level resource termination
```

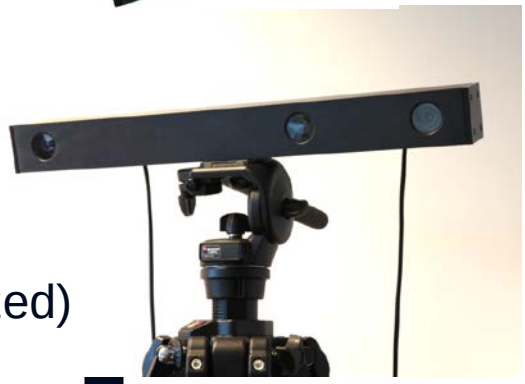
Pedestrian flow analysis in 2D

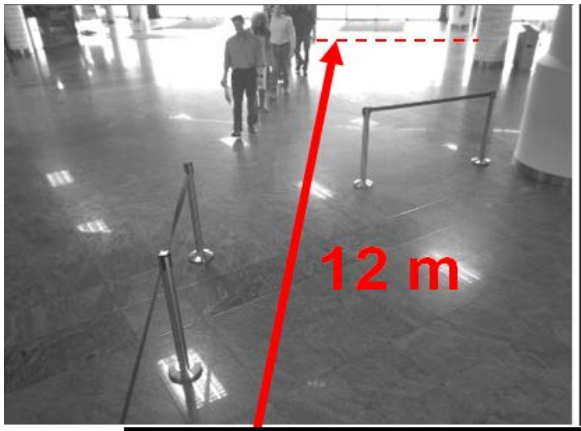


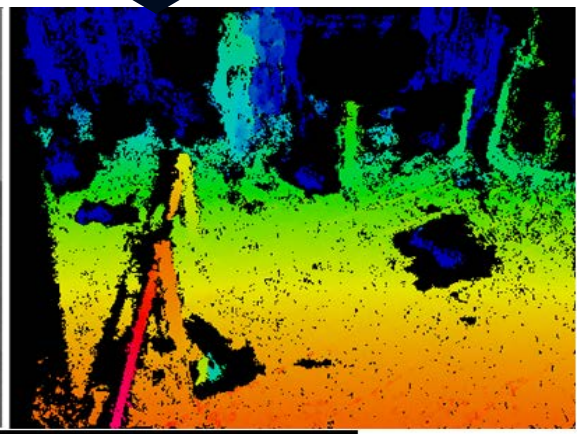
Public dataset: Grand Central Station, NYC: 720x480 pixels, computational speed: 35 *fps*

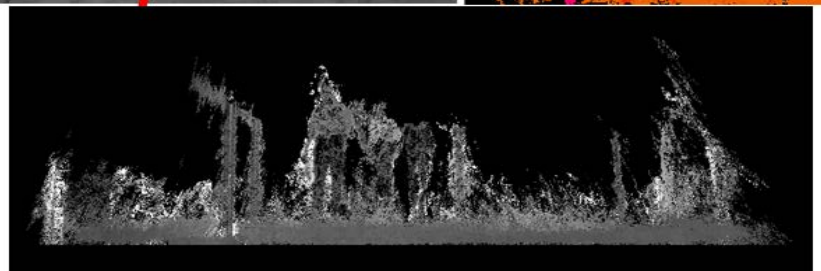
Passive stereo based depth measurement

- 3D stereo-camera system developed by AIT
  - Area-based, local-optimizing, correlation-based stereo matching algorithm
  - Specialized variant of the Census Transform
  - Resolution: typically ~1 Mpixel
  - Run-time: ~ 14 fps (Core-i7, multithreaded, SSE-optimized)
  - Excellent “depth-quality-vs.-computational-costs” ratio
  - USB 2 interface









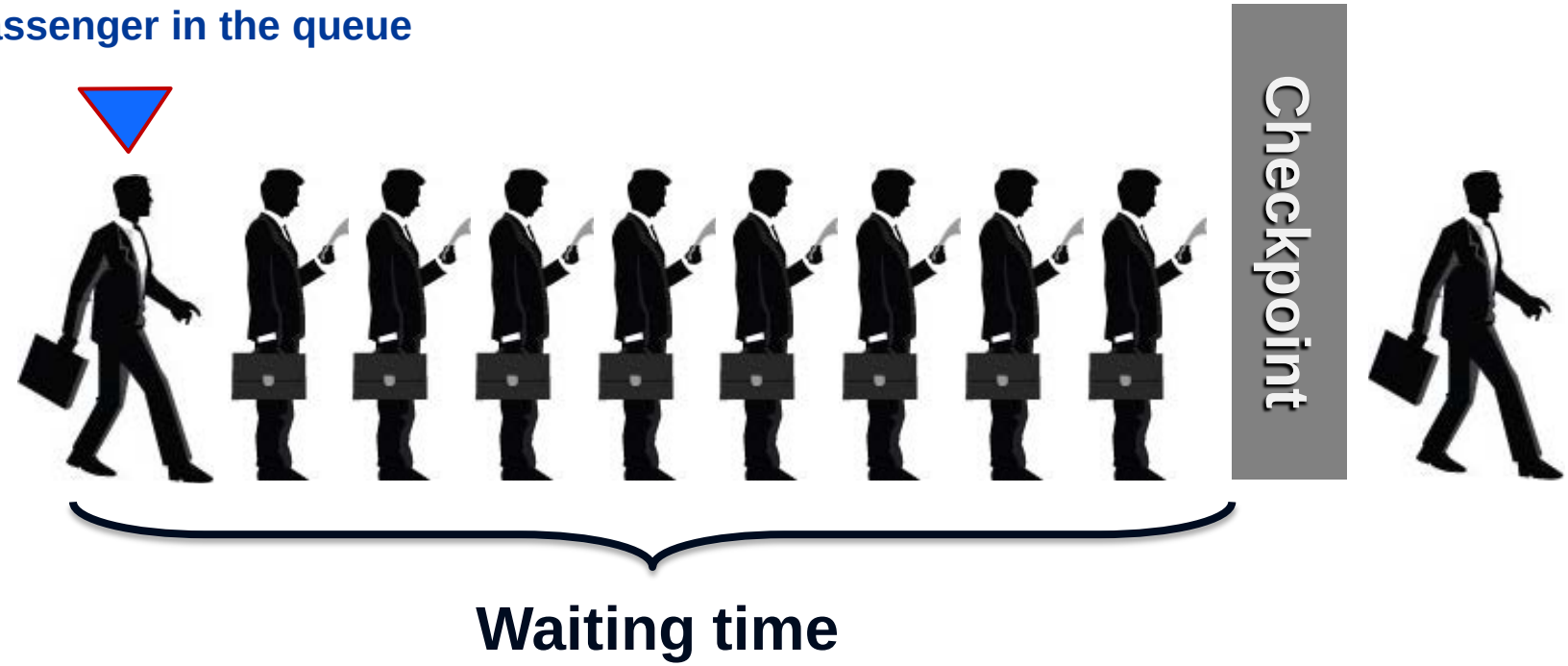
Advantage:

- Depth ordering of people
- Robustness against illumination, shadows,
- Enables scene analysis

Fast Detection Framework:
Queue Length + Waiting Time estimation

What is waiting time in a queue?

Time measurement relating to last passenger in the queue

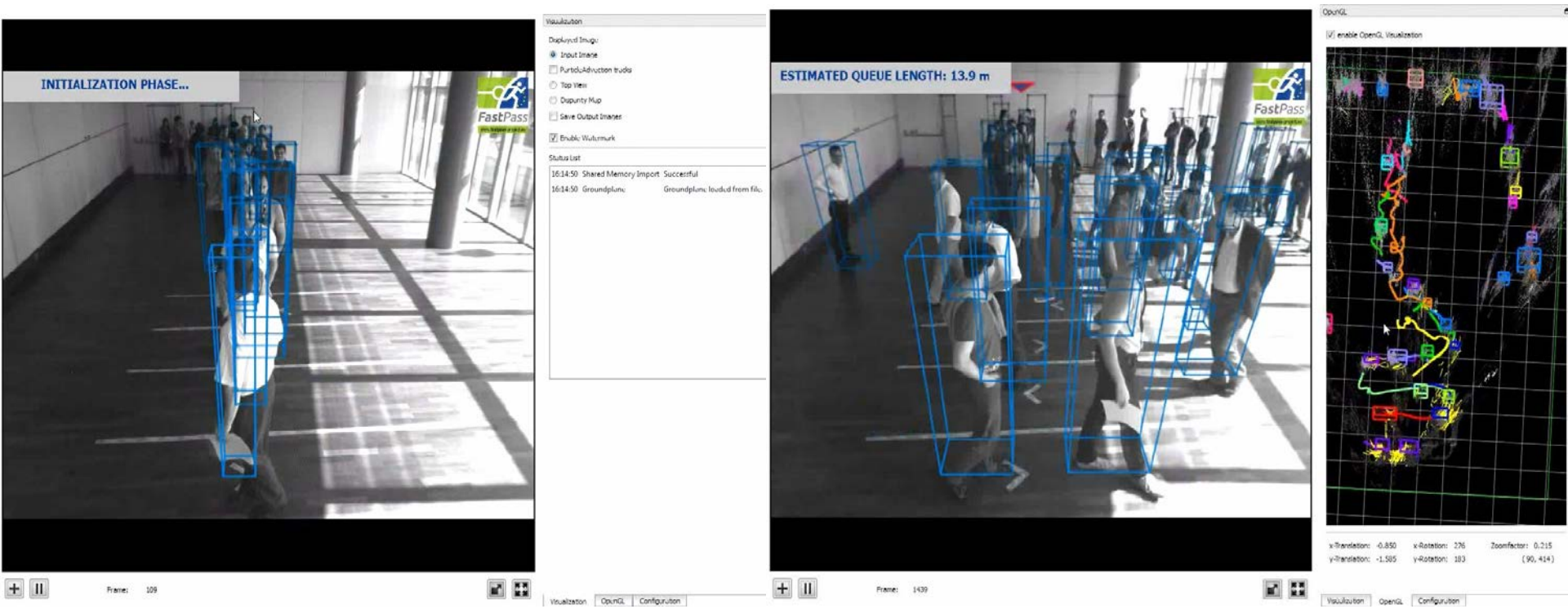


Why interesting?

Example: Announcement of waiting times (e.g. mobile app) → customer satisfaction

Example: Infrastructure operator → load balancing

Queue analysis (length, dynamics)



Visual queue analysis (1)

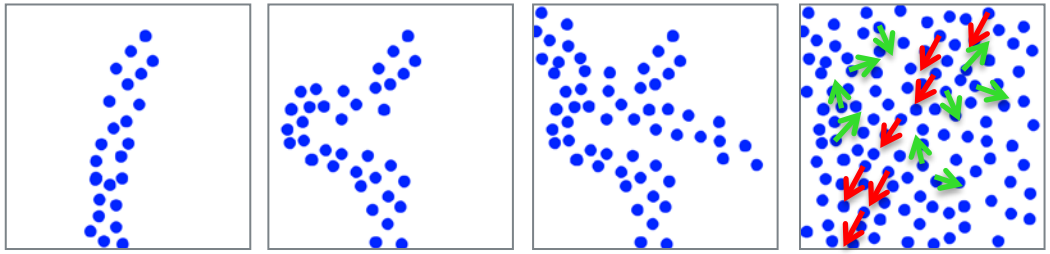
- Challenging problem

$$\textit{Waiting time} = \frac{\textit{Length}}{\textit{Velocity}}$$

→ 1. What is the shape and extent of the queue?

→ 2. What is the velocity of the propagation?

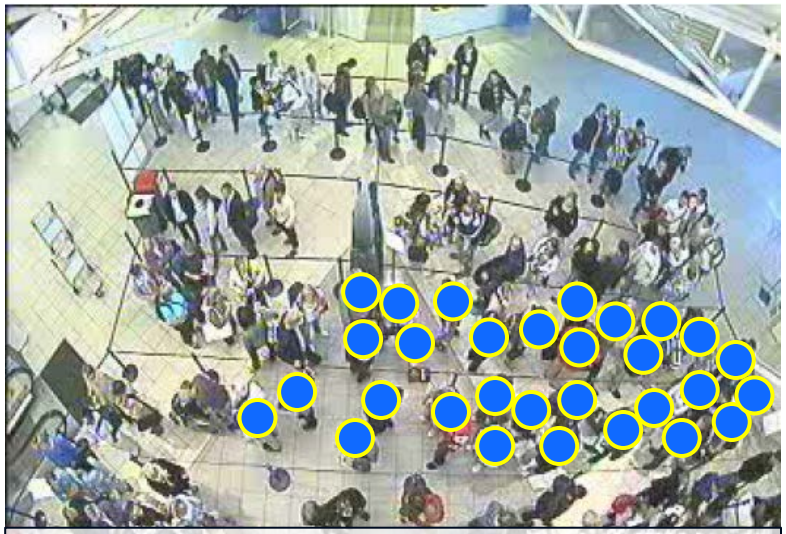
- Shape
  - No predefined shape (context/situation-dependent and time-varying)



- Motion→ not a pure translational pattern
  - Propagating s*top-and-go* behaviour with a noisy „background“
  - Signal-to-noise ratio depends on the observation distance

DEFINITION: Collective goal-oriented motion pattern of multiple humans exhibiting spatial and temporal coherence

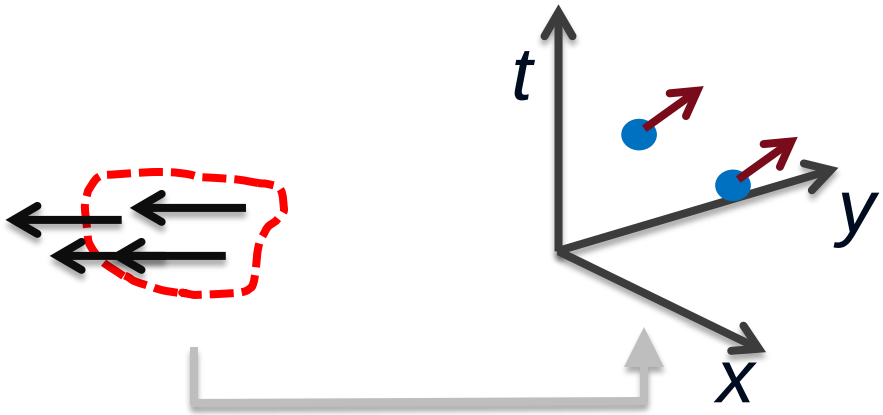
Visual queue analysis (2)

- 

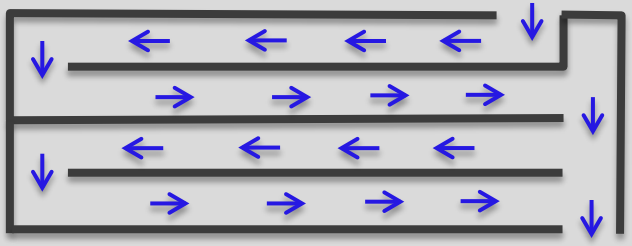
Source: Parameswaran et al. *Design and Validation of a System for People Queue Statistics Estimation*, Video Analytics for Business Intelligence, 2012

How can we detect (weak) correlation?

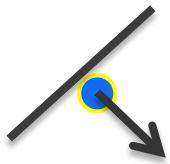
Correlation in space and time



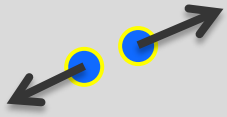
- Much data is necessary → Simulating crowding phenomena in Matlab
  - *Social force model* (Helbing 1998)



goal-driven kinematics – force field



repulsion by walls

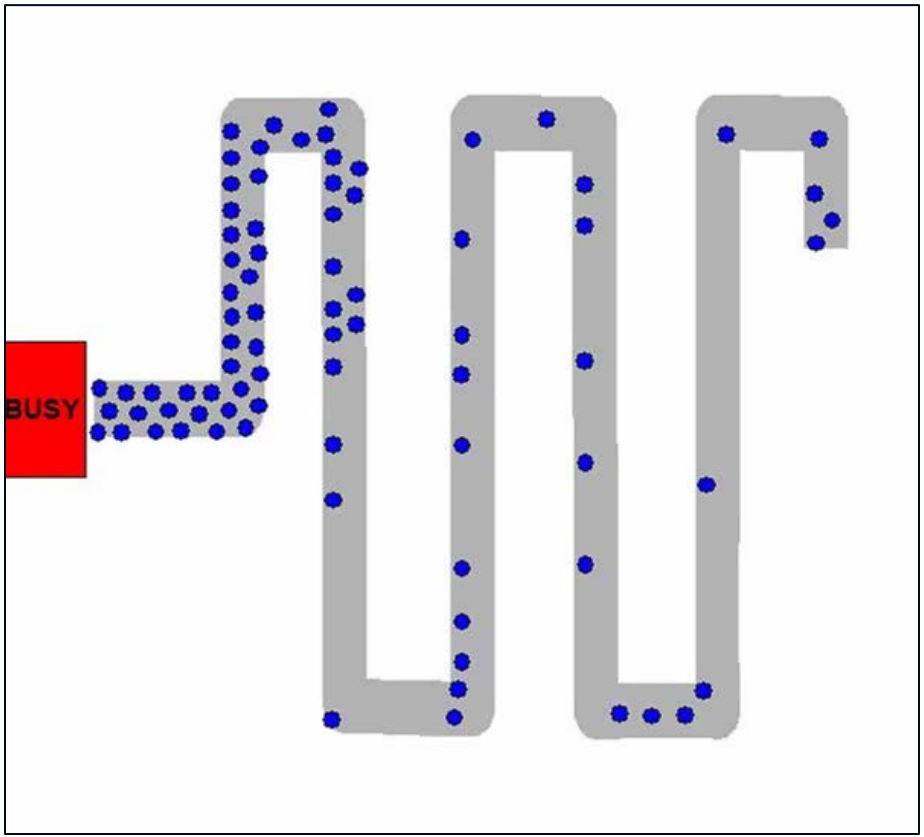


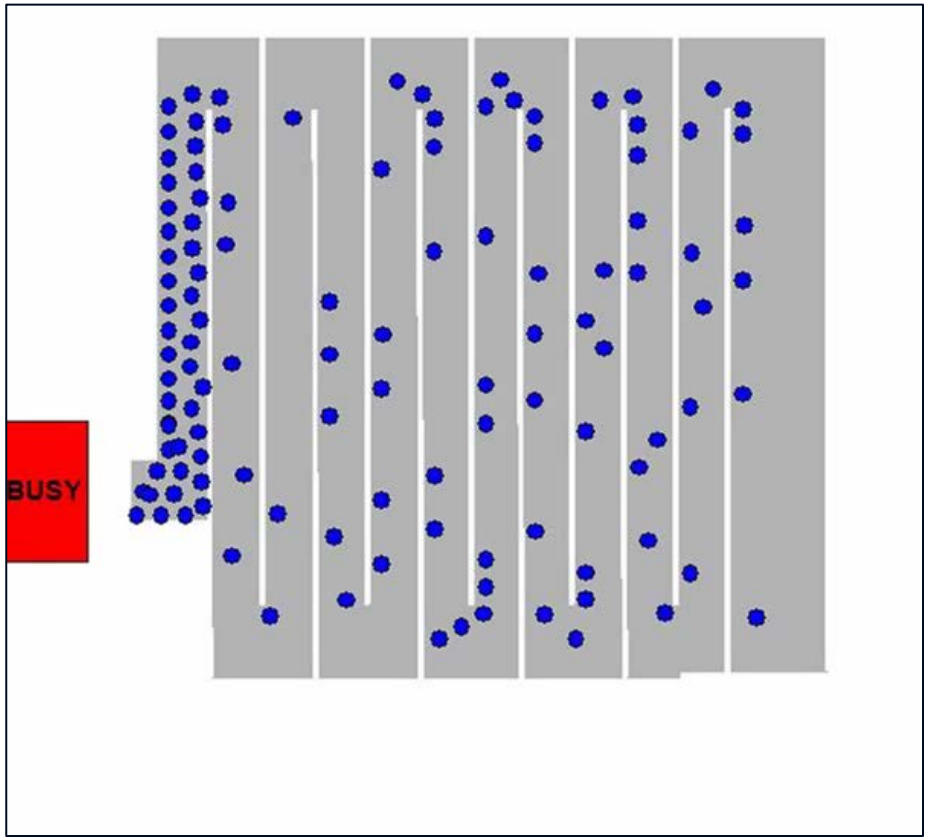
repulsion by „preserving privacy “

MATLAB simulation tool → Data with large variability

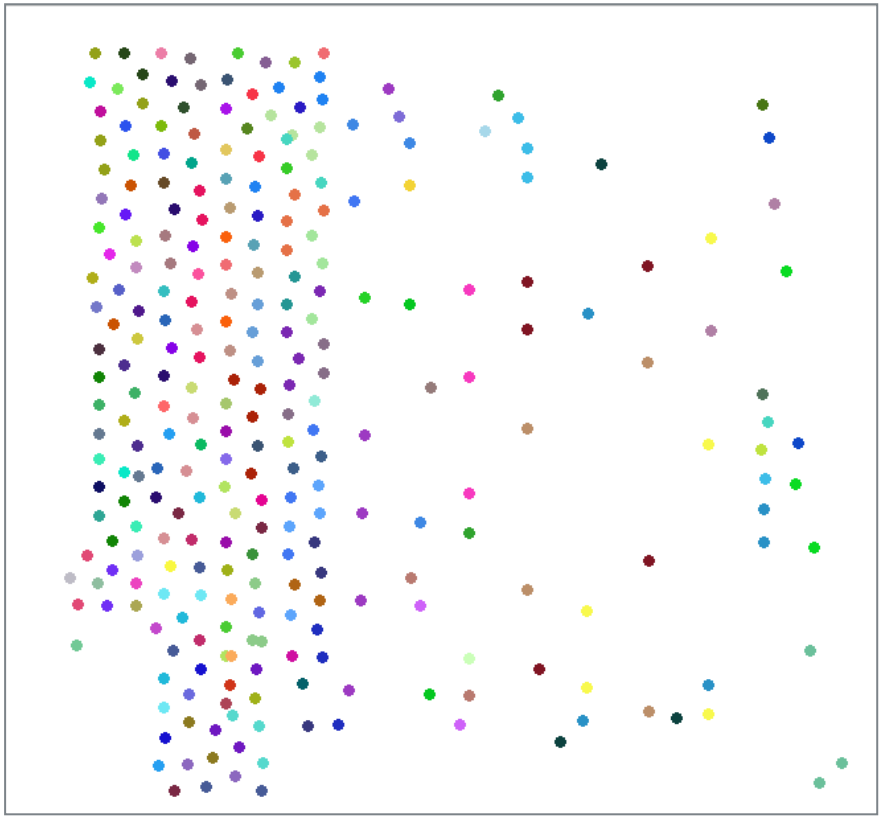
Creating queueing zones via MS Powerpoint as an Editor:

Two simulated examples (video) produced by Matlab:

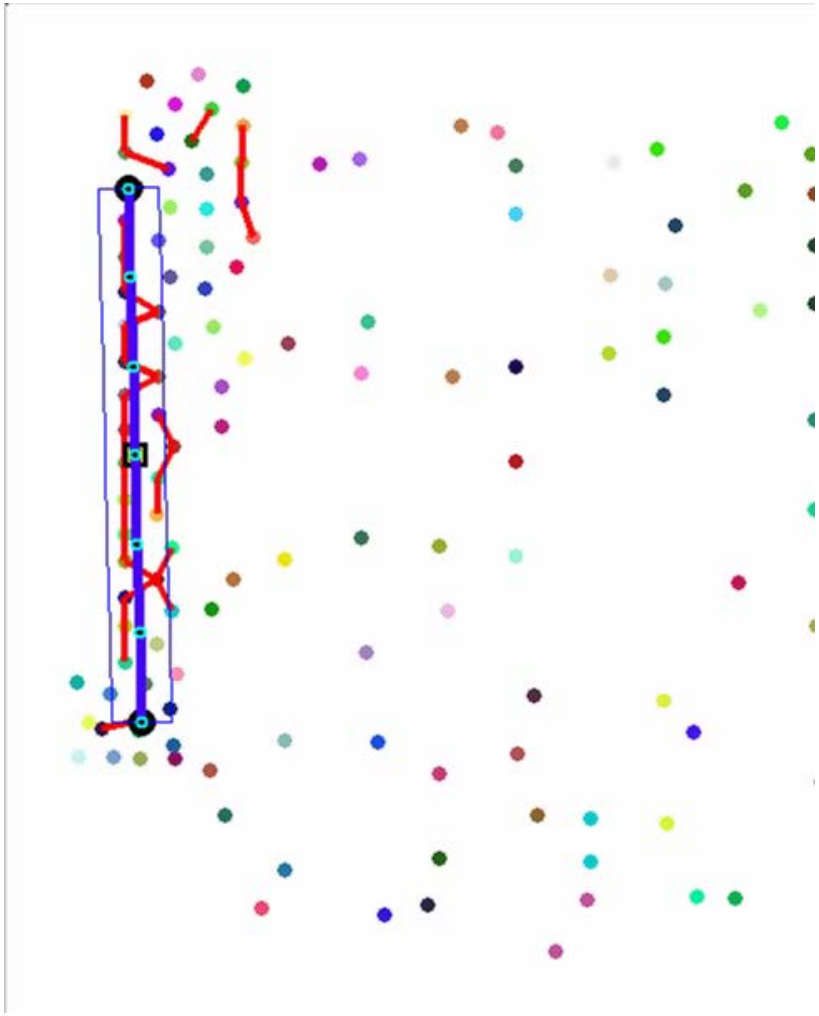




Queue analysis (length, dynamics)



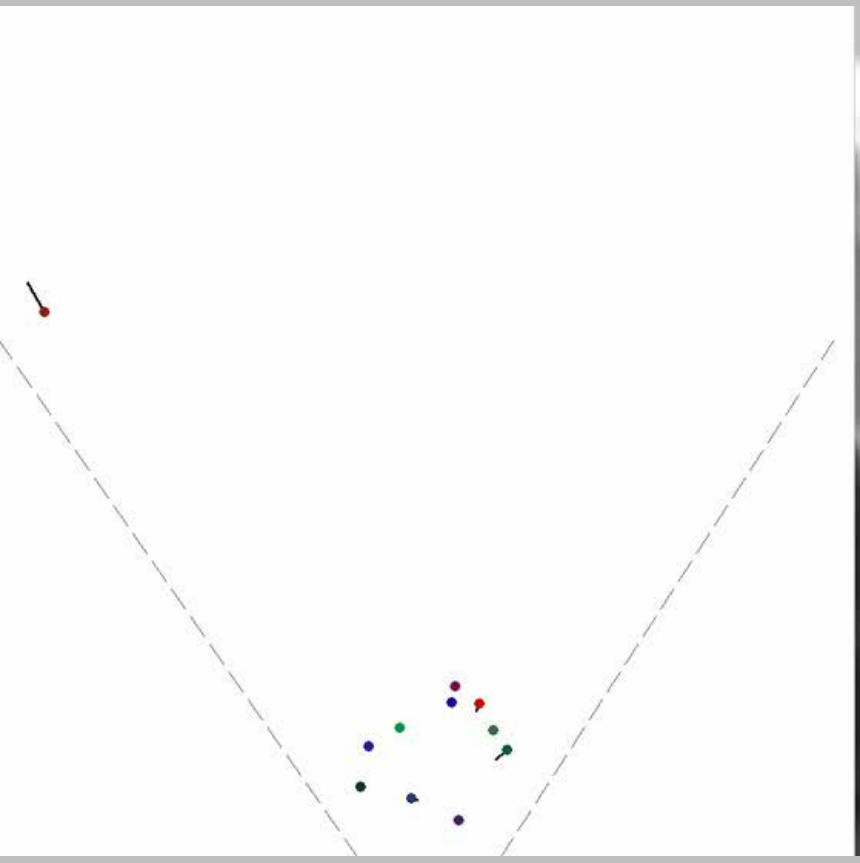
Pedestrian distribution: without movement



Video: Coherence analysis yielding the queue configuration

Adaptive estimation of the spatial extent of the queueing zone

Estimated configuration
(top-view)

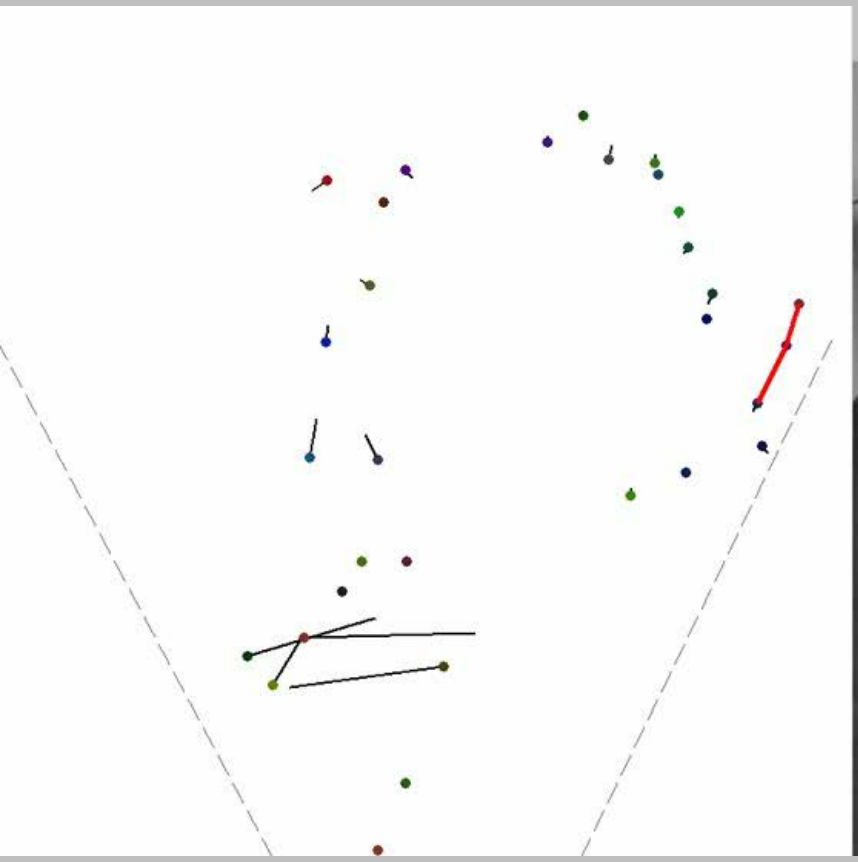


Detection results

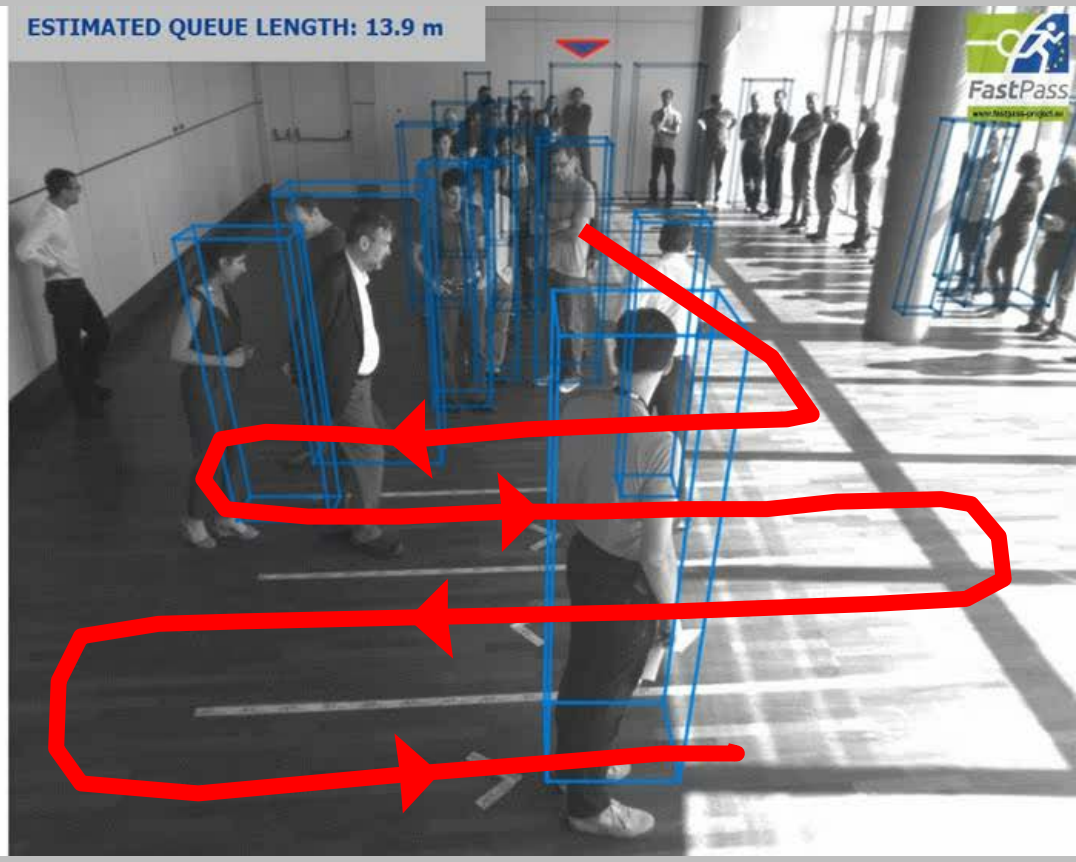
Left part of the image is intentionally blurred due to protecting the privacy of by-standers, who were not part of the experimental setup.

Adaptive estimation of the spatial extent of the queueing zone (meander-style queue)

Estimated configuration (top-view)



Detection results



Summary

- MATLAB is an essential tool for developing complex algorithmic units
- Achieving the same complexity in C/C++ is associated with significant development efforts

- Often, for a technical problem multiple solutions exist:
  - Enables rapid assessment of many alternatives by fast integration into an existing algorithmic chain.

- Further useful aspects not covered in the talk
  - *pcode* – protecting Matlab scripts
  - Built-in support for version control (Git, SVN) – 2014b
  - User interfaces allowing for tab-panels – 2014b
  - MatlabCentral und FileExchange

Thank you for your attention!

CSABA BELEZNAI
Senior Scientist
Safety & Security Department
Video- and Security Technology

AIT Austrian Institute of Technology GmbH
Donau-City-Straße 1 | 1220 Vienna | Austria
T +43(0) 664 825 1257 | F +43(0) 50550-4170
csaba.beleznai@ait.ac.at | http://www.ait.ac.at

http://www.d-sens.eu/