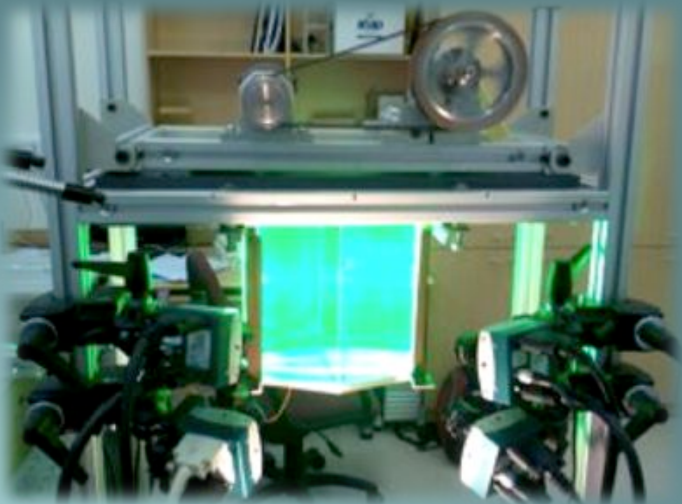
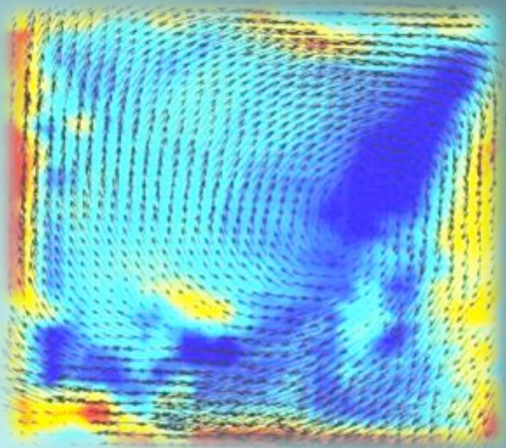
3D-PTV real-time image processing



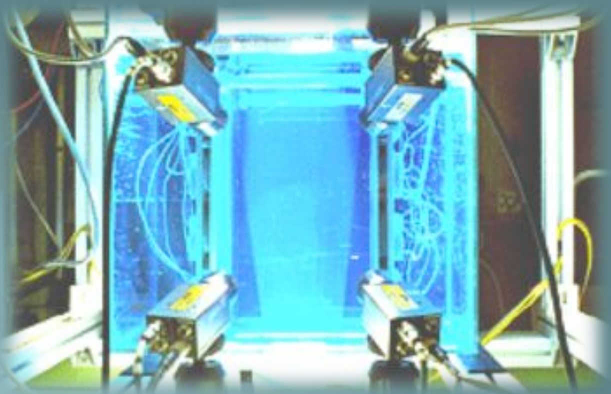
- By: **Mark Kreizer**
  - Mechanical Engineering, Tel-Aviv University



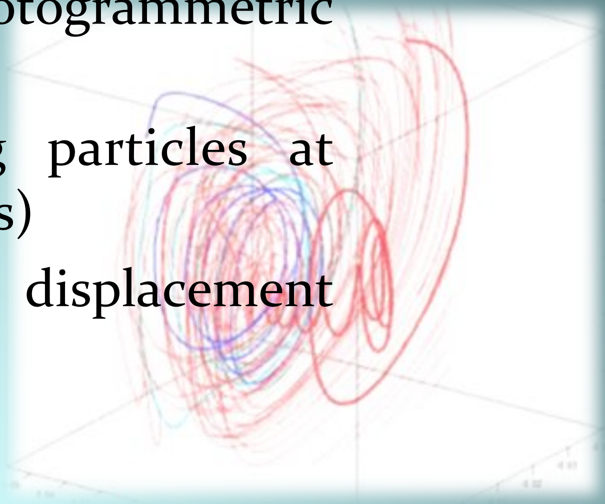
- Supervision of: **Dr. Alexander Liberzon**
  - Mechanical Engineering, Tel-Aviv University

Background - 3D-PTV – What is it?

- What is 3D-PTV?
  - Three Dimensional Particle Tracking Velocimetry (3D-PTV) is a **Lagrangian** 3D flow measuring technique – tracking particles in the flow (flow tracers)

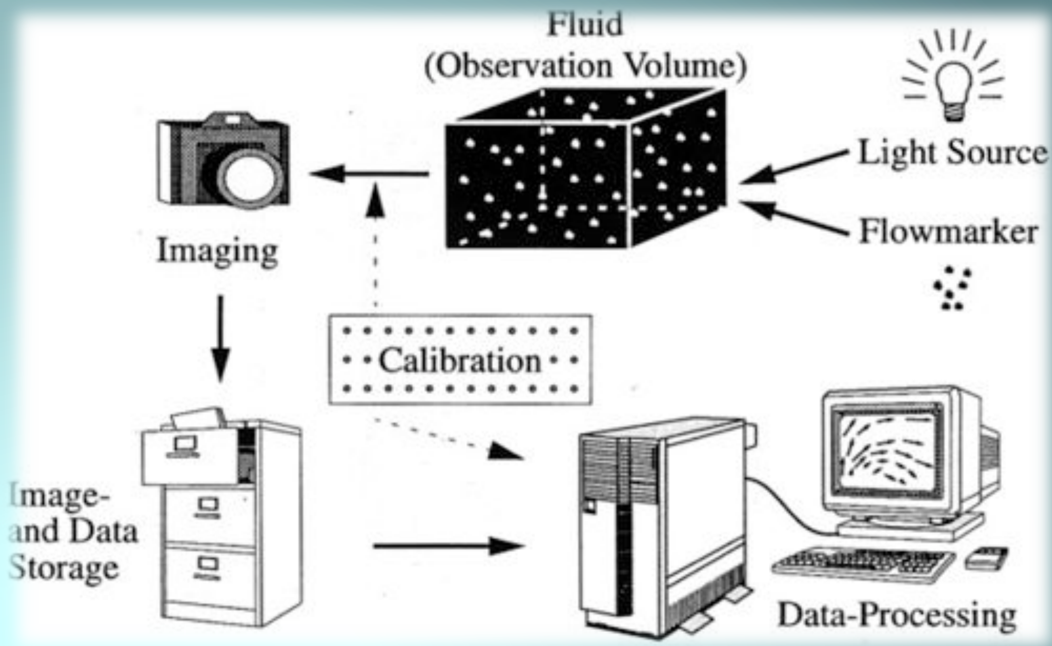


- What is the measuring principle?
  - Particles are **detected** by cameras
  - Particles' **3D location** is determined using photogrammetric principles
  - Particles' **trajectories** are built, by linking particles at following sequences (particle tracking algorithms)
  - Particles' **velocities** are determined by their displacement during a prescribed time interval.



Background - 3D-PTV – How?

- What is the technical principle?
  - Seeding **flow tracers** in the control volume
  - Illuminating the flow tracers using a **light source** (e.g. Laser, Led)
  - Multiple **cameras** recording the control volume
  - Software **post processing** of the data



Background - 3D-PTV – Problems

- 3D-PTV is not that commonly used because of:
  - Cost
  - Immobility
  - Cumbersomeness

- *“Could anything be done to solve these problems?”*

Background - 3D-PTV – Problems – Data rate

- Data rate for a typical experiment (2 minutes, 500 fps)

Frame size · Frame rate · Data rate for a single camera

$$1.3\frac{MB}{frame}\times 500\frac{frames}{sec}=650\frac{MB}{sec}$$

TOTAL DATA RATE

$$650\frac{MB}{sec}\times 4\ cameras=2.55\frac{GB}{sec}$$

- High-speed hard-drive continuous writing speed < 50 MB/s – *Less than required* (The main Bottleneck)

TOTAL DATA SIZE

$$2.55\frac{GB}{sec}\times 2\ minutes=300\ GB$$

Goals

- The main goal: Wide spread usage of 3D-PTV.

"I can do low-budget 3D Lagrangian measurements with my cool system, and everywhere !"

- Our Goals
  1. Increasing **mobility**
     - Allow outdoor experiments using regular PCs or even Laptops.
  2. **Simplifying** the system and its components
     - Reduces the cost of 3D-PTV systems
     - Increases modularity

The Solution

- **Reducing data rates dramatically !!**

1. **Mobility** – Throw away the huge storage unit

2. **Cost**
   - Standard high-speed cameras
   - Standard PC
   - No frame grabbers.
   - No multi-array of HDs
   - No data controllers

The Solution – How?!

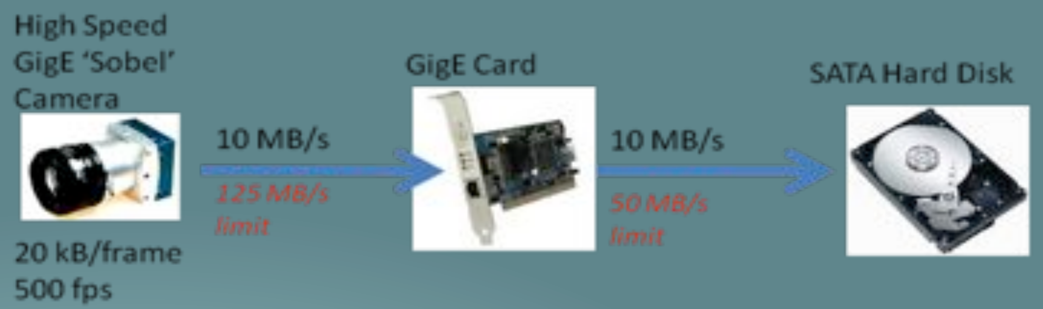
- Most of the information in a typical frame is redundant

- Particles' X-Y pixel position list holds up to 2% of the original full frame size:
  - for each camera, **10** MB/s (!!) instead of **650** MB/s.
  - 6 GB data of the whole experiment instead of 300 GB.

No storage unit is required anymore !!!

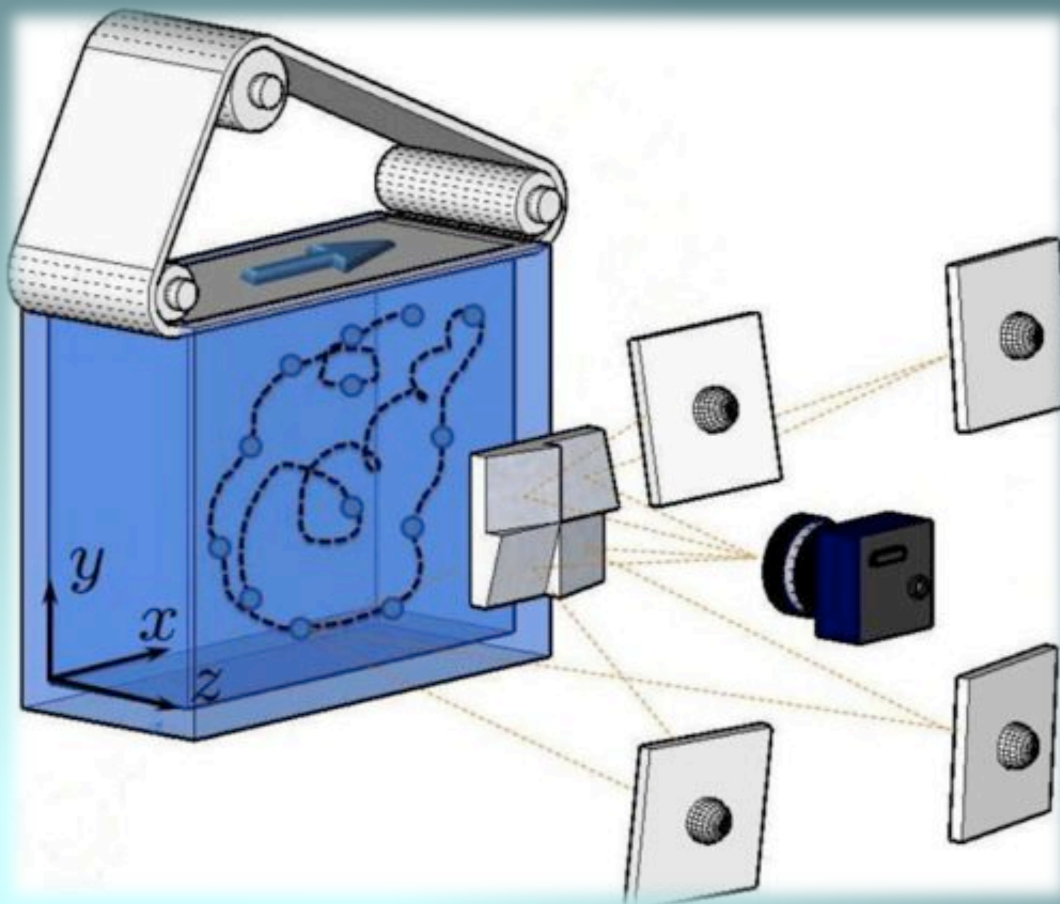
The Solution – Our implementation

- A single **camera** with **real-time processing** capability (onboard FPGA)



- View splitter mirrors array - Replacing 4 cameras with a single one.

- Laptop/PC with open-source PTV software



Camera

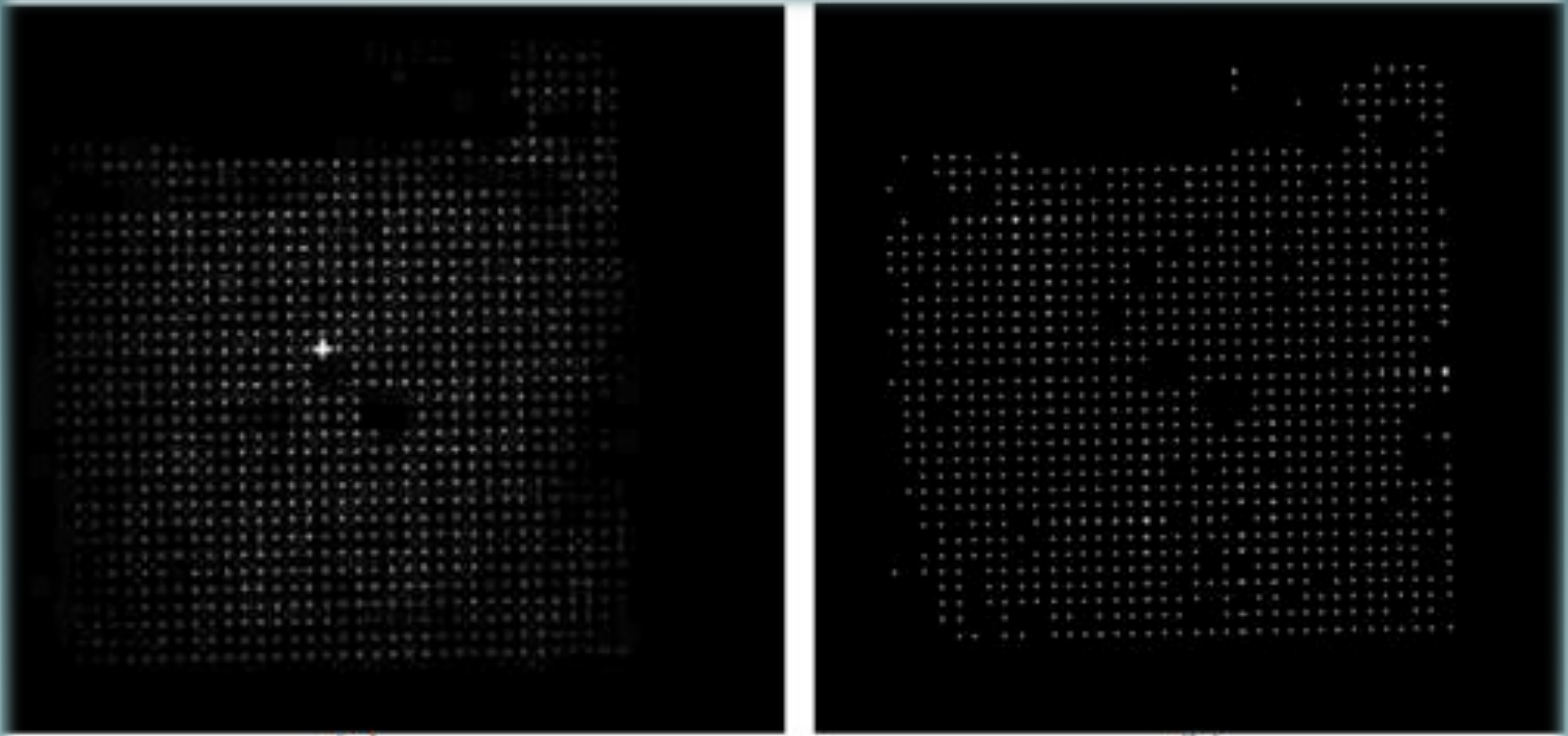
- 'Mikrotron MC1324' - GigE CMOS camera with onboard FPGA for real-time particles recognition (based on Sobel filter)
  - Up to 500 fps
  - 1280 x 1024 resolution
  - Output data of 20 kB raw file for each frame (instead of 1.3 MB). Contains x-y pixel position of each recognized particle.

00004185.raw		
00004186.raw		
00004187.raw		
00004188.raw		
00004189.raw	20 KB	RAW File
00004190.raw	20 KB	RAW File

Camera

Original View (**unprocessed**)

Visualization of **Processed** image



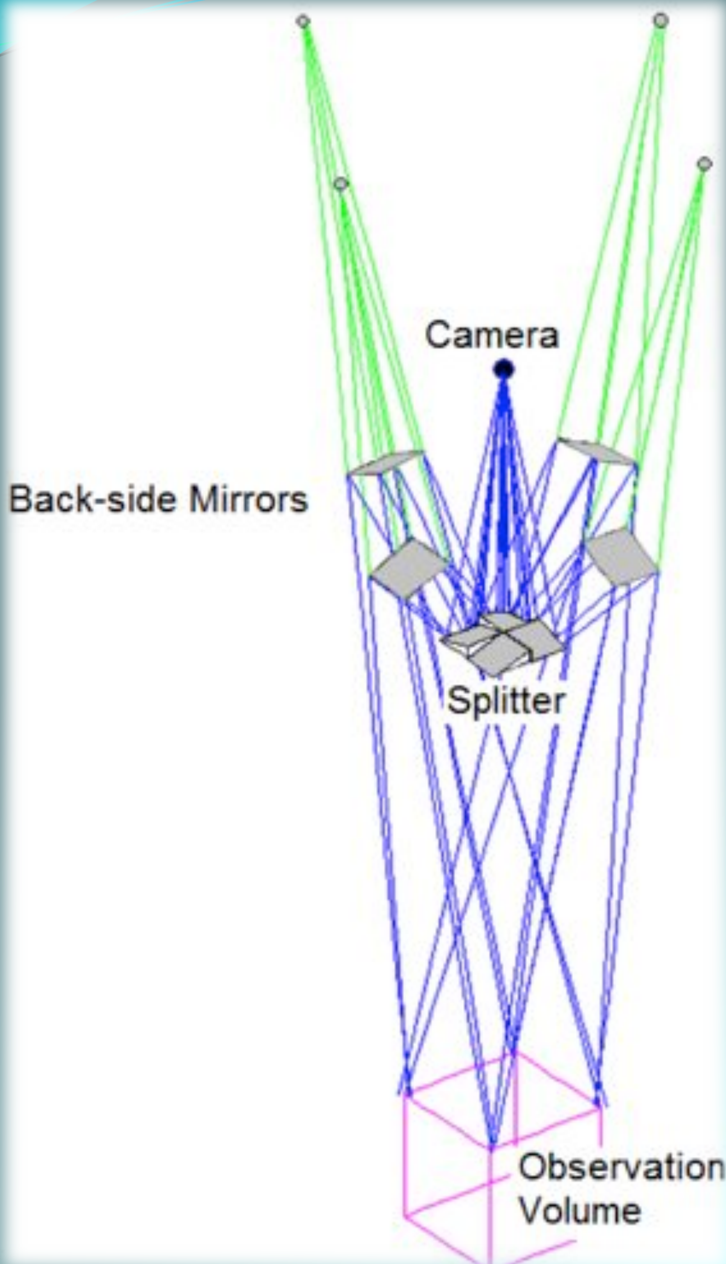
(a)

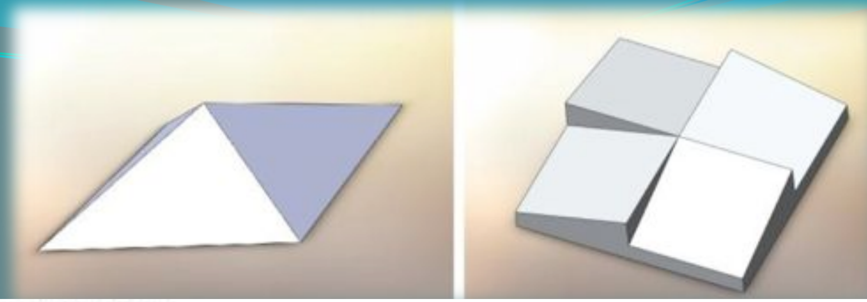
(b)

Camera

“**Sobel** Mode” visualization movie

View Splitter Array





Equation 1 – Vector's length equals one

$$\|Ray0_{bl}.c.vector\| = 1$$

Equation 2 - Vector's length equals one

$$\|Ray1_{bl}.c.vector\| = 1$$

Equation 3 – Back-side mirror normal vector has to be perpendicular to the mirror

$$\left[[Ray0_{bl}.b.head] - [Ray1_{bl}.b.head]\right] \cdot \left[[Ray0_{bl}.b.vector] - [Ray0_{bl}.c.vector]\right] = 0$$

Equation 4 – Back-side mirror normal vectors at both reflection points must be parallel with each other

$$\begin{vmatrix} 1 \quad 1 \quad 1 \\ \left[[Ray0_{bl}.b.vector] - [Ray0_{bl}.c.vector]\right] \\ \left[[Ray1_{bl}.b.vector] - [Ray1_{bl}.c.vector]\right] \end{vmatrix} = 0$$

Equation 5 – Back-side mirror reflection point product from section "b" and from section "c" must coincide

$$\begin{pmatrix} [Ray0_{bl}.b.tail] \\ [Ray1_{bl}.b.tail] \\ [Ray0_{bl}.c.tail] \\ [Ray1_{bl}.c.tail] \end{pmatrix} + \begin{pmatrix} r0 & 0 & 0 & 0 \\ 0 & r1 & 0 & 0 \\ 0 & 0 & t0 & 0 \\ 0 & 0 & 0 & t1 \end{pmatrix} \begin{pmatrix} [Ray0_{bl}.b.vector] \\ [Ray1_{bl}.b.vector] \\ [Ray0_{bl}.c.vector] \\ [Ray1_{bl}.c.vector] \end{pmatrix} = \begin{pmatrix} [Ray0_{bl}.b.head] \\ [Ray1_{bl}.b.head] \\ [Ray0_{bl}.c.head] \\ [Ray1_{bl}.c.head] \end{pmatrix}$$

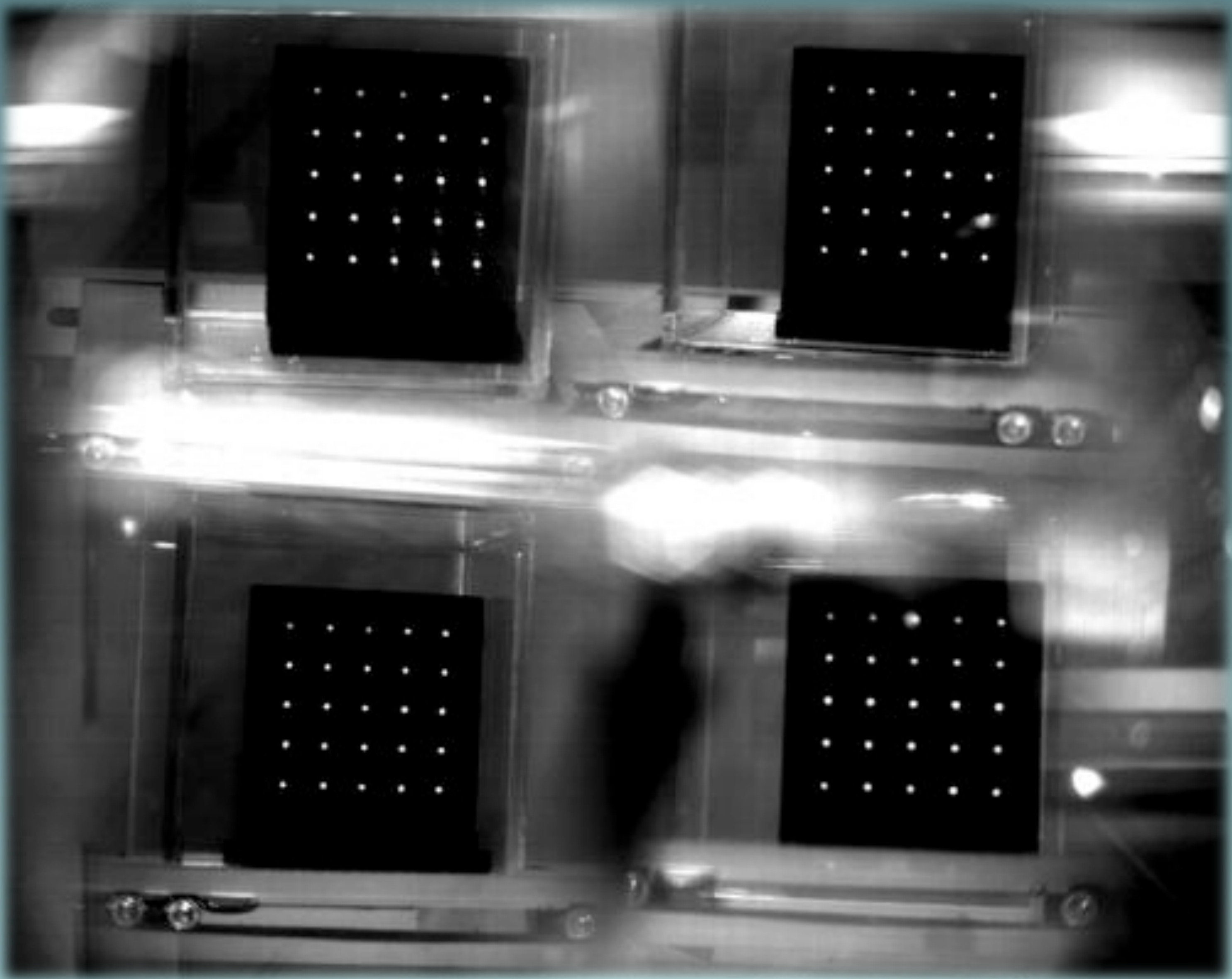
Equation 6 – "b.head" is "c.tail"

$$[Ray0_{bl}.c.tail] = [Ray0_{bl}.b.head]$$

Equation 7– "b.head" is "c.tail"

$$[Ray1_{bl}.c.tail] = [Ray1_{bl}.b.head]$$

View Splitter Array



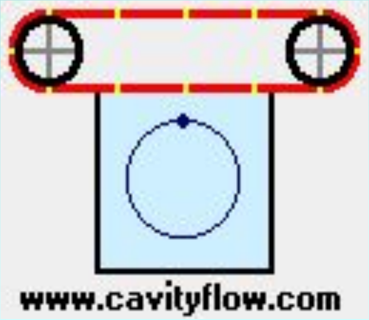
View Splitter Array

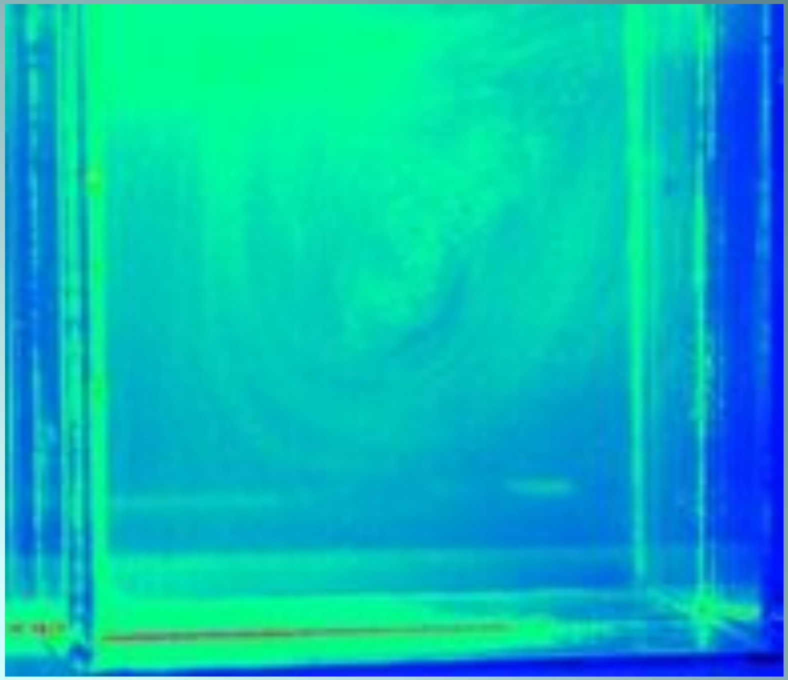
View-Splitter movie

Validation & Evaluation

1. **Detection rate** of the camera in “Sobel Mode”
2. **Our camera** (2D-PTV) in comparison with **PIV**
   - The 2D-PTV was performed using our camera, working in “Sobel Mode”, without the view-splitter.
3. **Our system** (3D-PTV) in comparison with **standard 3D-PTV**

- Experimental tests facility
  - Canonical flow in a cubic lid-driven cavity (1:1:1 aspect ratio)



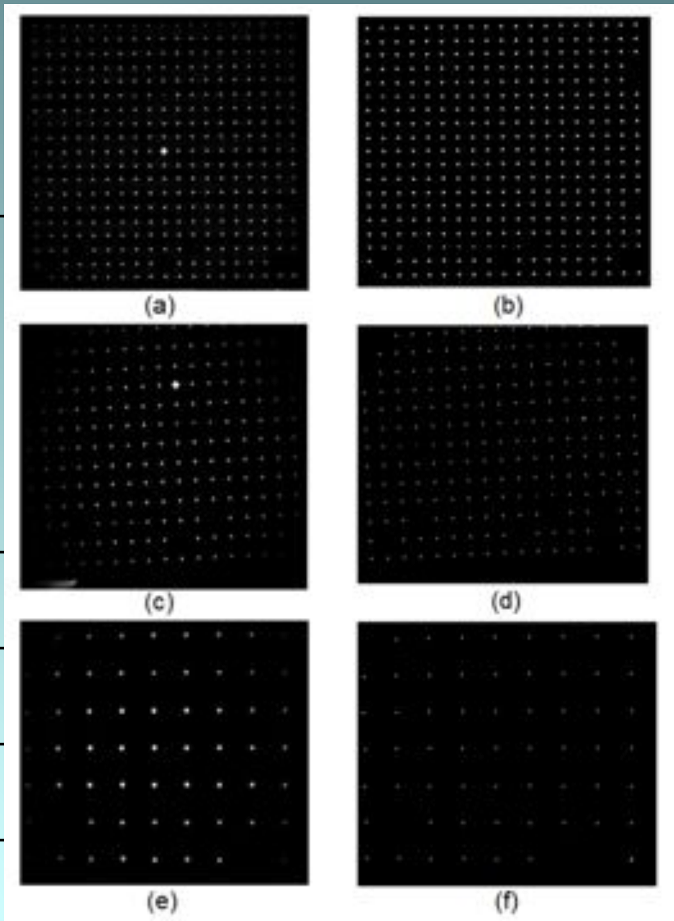


Validation & Evaluation – Detection Rate

1. **Detection rate** of the camera in "Sobel Mode"

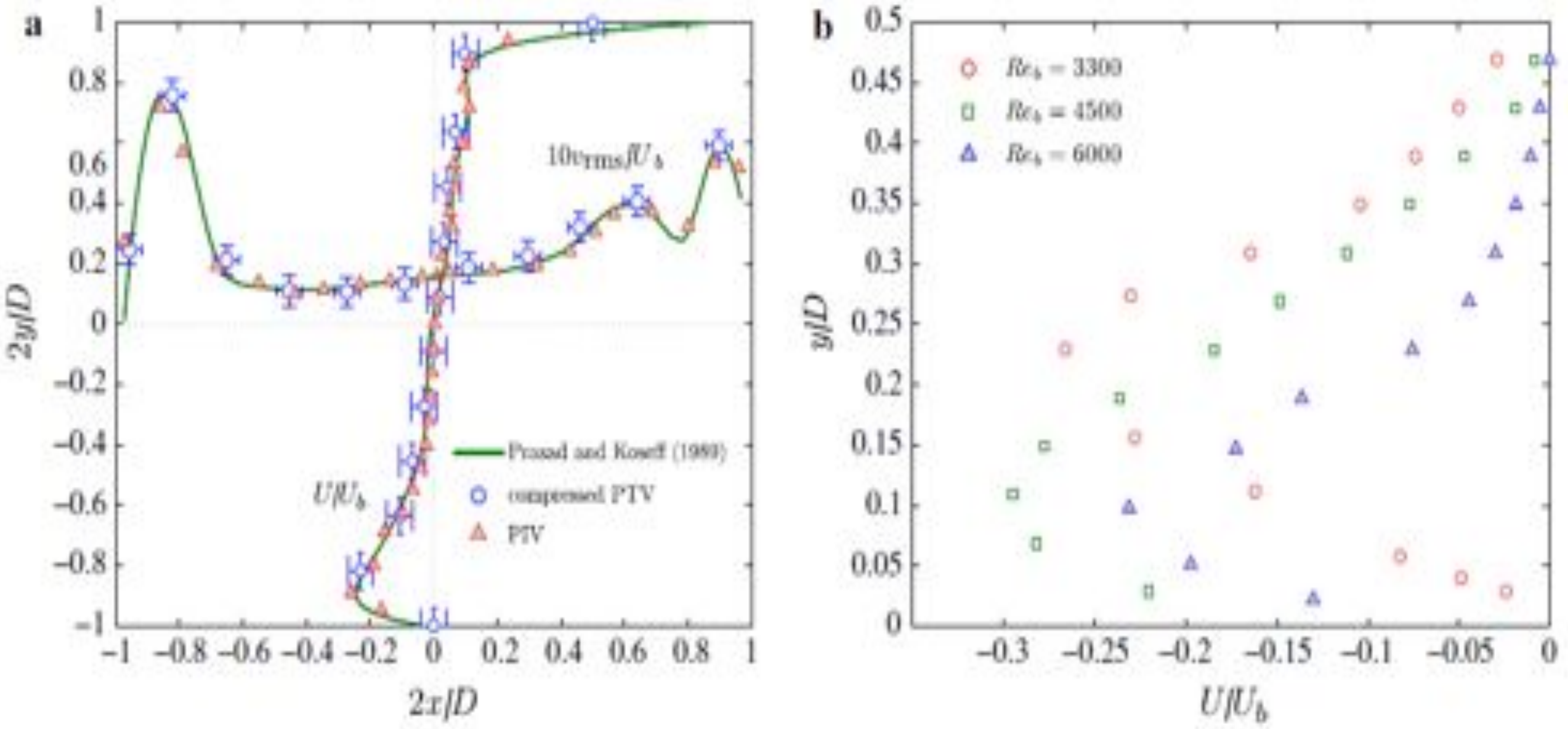
$$\text{Detection Rate} = \frac{\text{True Detections} - \text{False Detections}}{\text{Total number of particles}} \cdot 100$$

Series #	Distance between camera and calibration body [cm]	Total number of Particles on the Calibration body	Particles Size, [Pixels]	True detections [particles]	Number of false detections	**detection rate [%]**
1	120	980	2±1	967	3	**98.3**
2	80	354	5±2	352	0	**99.4**
3	60	210	15±3	207	1	**98.1**
4	30	60	40±5	59	1	**96.6**



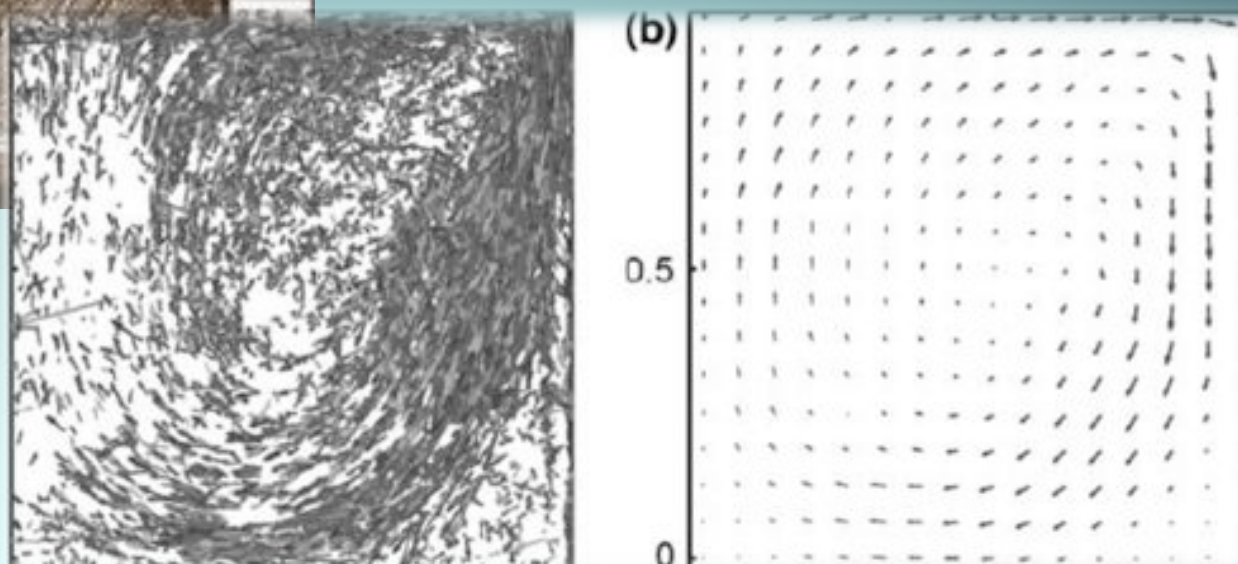
Validation & Evaluation – Results

Test #2 - Comparison: 2D-PTV vs. PIV



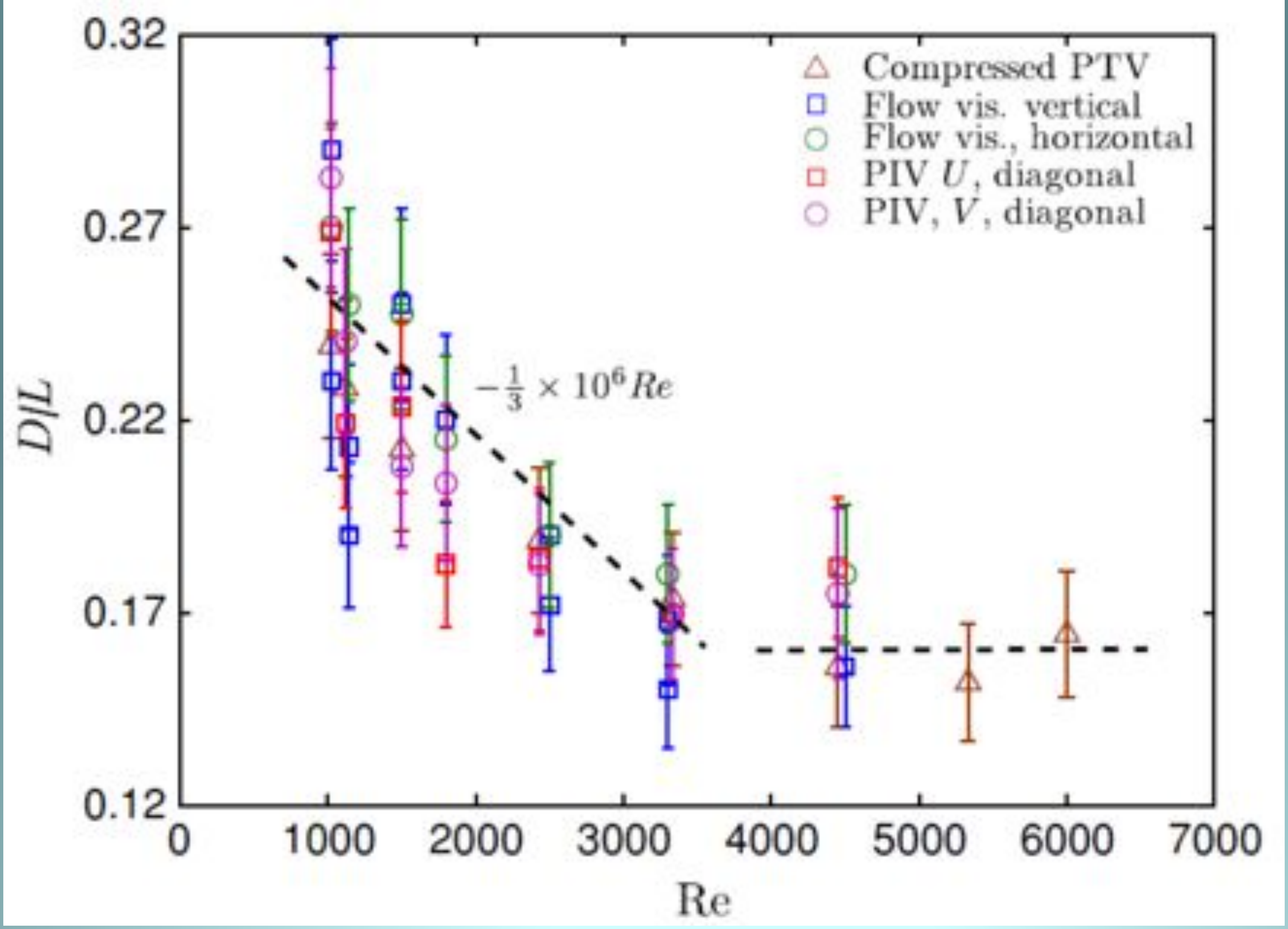
Validation & Evaluation – Results

Test #2 - Comparison: 2D-PTV vs. PIV



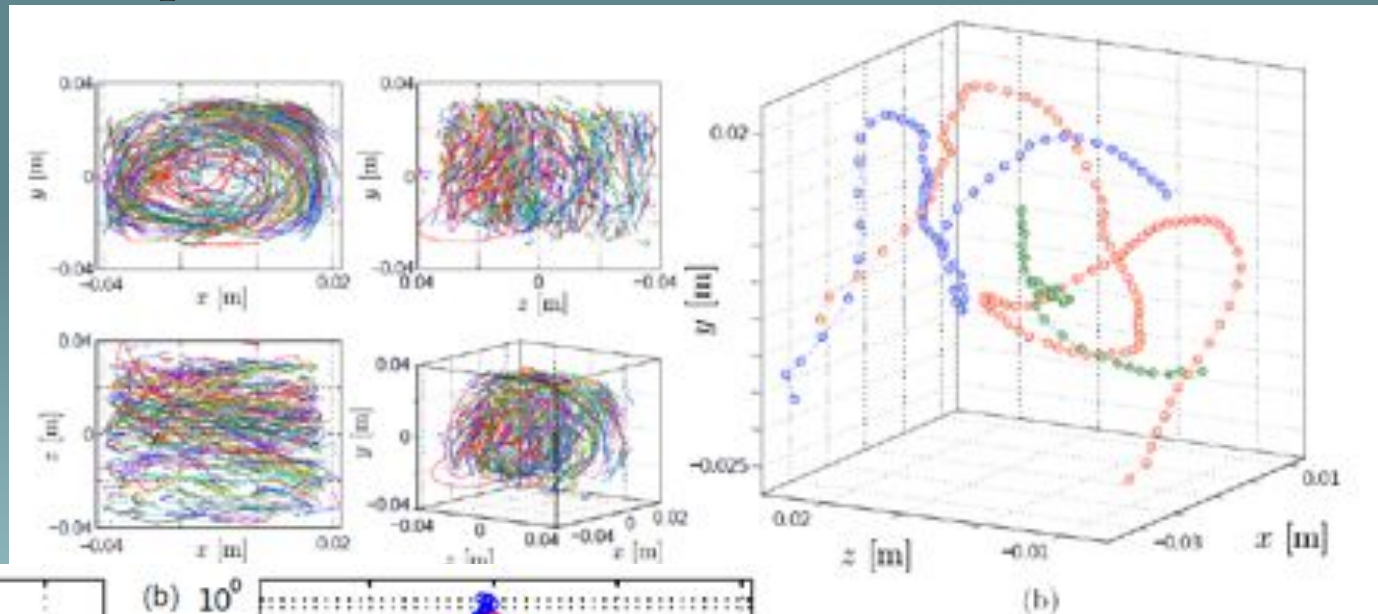
Validation & Evaluation – Results

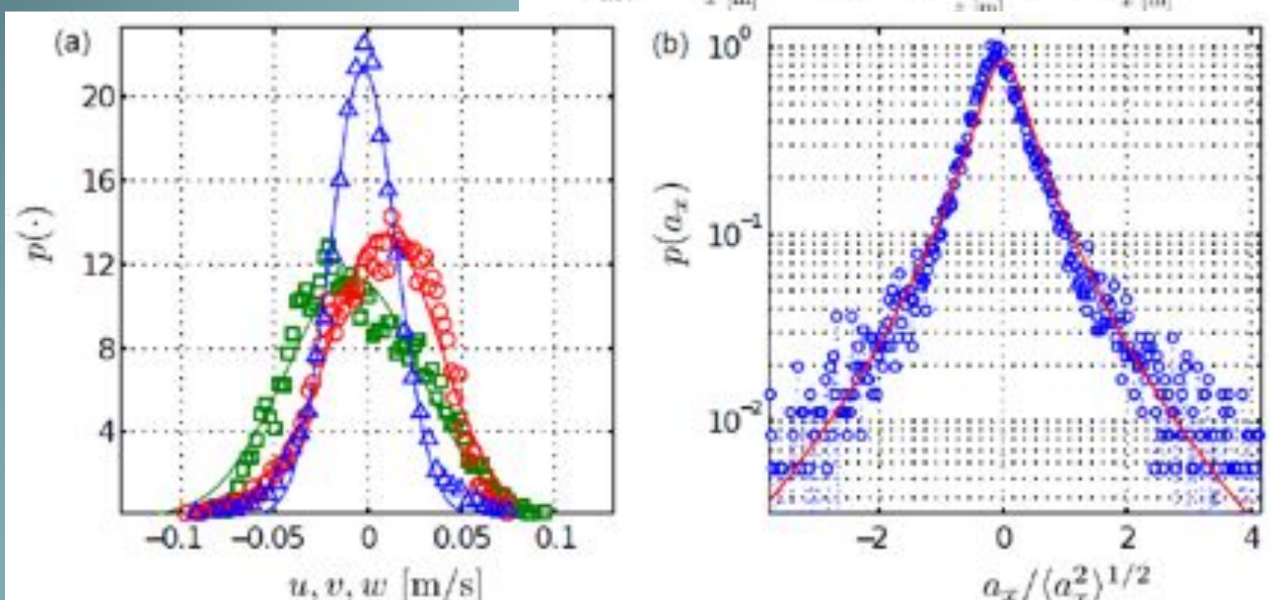
Test #2 - Comparison: 2D-PTV vs. PIV



Validation & Evaluation – Results

Test #3 - 3D-PTV Comparison – Our vs. Off-the-shelf

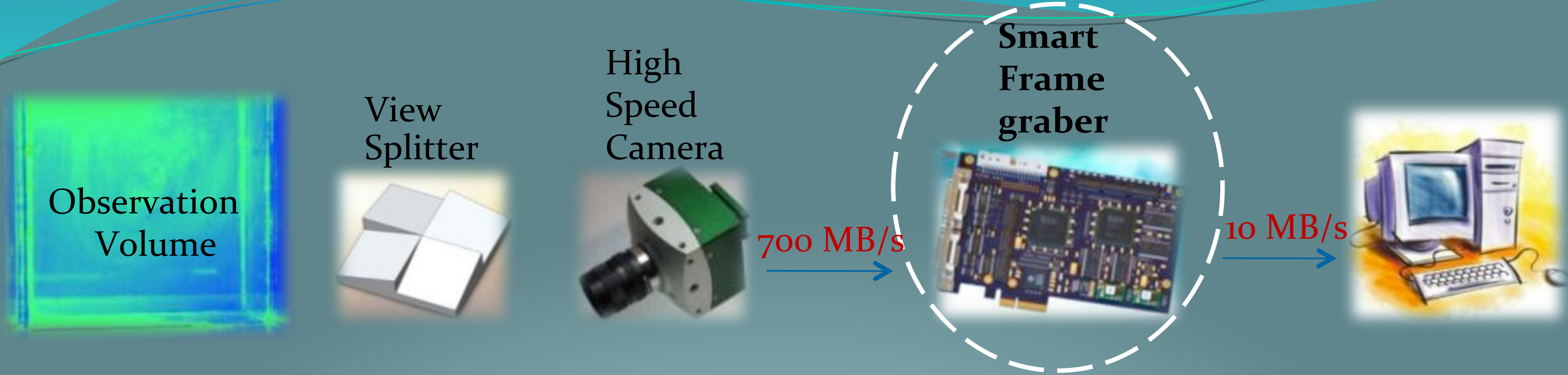




Future - Possible implementation

- The system can provide a solution for the remotely controlled tracking experiments:
  - Microgravity
  - Underwater
  - Harsh experimental conditions

Future - Possible Developments



- Frame-grabber with onboard **programmable FPGA via software**
- The frame-grabber performs **real-time image processing**, thus reduces data rates:
  - Split the image into four images
  - Enhance each of the four images, individually.
  - Apply Sobel filter, blob analysis for particles detection
  - Create a miniature text list file including data regarding the detected particles