3D-PTV real-time image processing

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

Supervision of: Dr. Alexander Liberzon
• Mechanical Engineering, Tel-Aviv University
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
3D-PTV is not that commonly used because of:

- Cost
- Immobility
- Cumbersome

“Could anything be done to solve these problems?”
Data rate for a typical experiment (2 minutes, 500 fps)

<table>
<thead>
<tr>
<th>Frame size</th>
<th>Frame rate</th>
<th>Data rate for a single camera</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.3 MB/\text{frame}</td>
<td>500 \text{frames/sec}</td>
<td>650 MB/sec</td>
</tr>
</tbody>
</table>

\[
\frac{1.3 \text{ MB}}{\text{frame}} \times 500 \frac{\text{frames}}{\text{sec}} = 650 \frac{\text{MB}}{\text{sec}}
\]

**Total Data Rate**

\[
650 \frac{\text{MB}}{\text{sec}} \times 4 \text{ cameras} = 2.55 \frac{\text{GB}}{\text{sec}}
\]

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

**Total Data Size**

\[
2.55 \frac{\text{GB}}{\text{sec}} \times 2 \text{ minutes} = 300 \text{ GB}
\]
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
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
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.
Original View (unprocessed)

Visualization of **Processed** image
“Sobel Mode” visualization movie
View Splitter Array

Equation 1 – Vector’s length equals one
\[ \| \text{Ray}_{0_{bl}.c.\text{vector}} \| = 1 \]

Equation 2 – Vector’s length equals one
\[ \| \text{Ray}_{1_{bl}.c.\text{vector}} \| = 1 \]

Equation 3 – Back-side mirror normal vector has to be perpendicular to the mirror
\[ ([\text{Ray}_{0_{bl}.b.\text{head}}] - [\text{Ray}_{1_{bl}.b.\text{head}}]) \cdot ([\text{Ray}_{0_{bl}.b.\text{vector}}] - [\text{Ray}_{0_{bl}.c.\text{vector}}]) = 0 \]

Equation 4 – Back-side mirror normal vectors at both reflection points must be parallel with each other
\[ \begin{vmatrix} 1 & 1 & 1 \\ ([\text{Ray}_{0_{bl}.b.\text{vector}}] - [\text{Ray}_{0_{bl}.c.\text{vector}}]) & ([\text{Ray}_{1_{bl}.b.\text{vector}}] - [\text{Ray}_{1_{bl}.c.\text{vector}}]) \end{vmatrix} = 0 \]

Equation 5 – Back-side mirror reflection point product from section “b” and from section “c” must coincide
\[ \begin{pmatrix} [\text{Ray}_{0_{bl}.b.\text{tail}}] \\ [\text{Ray}_{1_{bl}.b.\text{tail}}] \\ [\text{Ray}_{0_{bl}.c.\text{tail}}] \\ [\text{Ray}_{1_{bl}.c.\text{tail}}] \end{pmatrix} + \begin{pmatrix} r & 0 & 0 & 0 \\ 0 & r1 & 0 & 0 \\ 0 & 0 & t0 & 0 \\ 0 & 0 & 0 & t1 \end{pmatrix} \begin{pmatrix} [\text{Ray}_{0_{bl}.b.\text{vector}}] \\ [\text{Ray}_{1_{bl}.b.\text{vector}}] \\ [\text{Ray}_{0_{bl}.c.\text{vector}}] \\ [\text{Ray}_{1_{bl}.c.\text{vector}}] \end{pmatrix} = \begin{pmatrix} [\text{Ray}_{0_{bl}.b.\text{head}}] \\ [\text{Ray}_{1_{bl}.b.\text{head}}] \\ [\text{Ray}_{0_{bl}.c.\text{head}}] \\ [\text{Ray}_{1_{bl}.c.\text{head}}] \end{pmatrix} \]

Equation 6 – “b.head” is “c.tail”
\[ [\text{Ray}_{0_{bl}.c.\text{tail}}] = [\text{Ray}_{0_{bl}.b.\text{head}}] \]

Equation 7 – “b.head” is “c.tail”
\[ [\text{Ray}_{1_{bl}.c.\text{tail}}] = [\text{Ray}_{1_{bl}.b.\text{head}}] \]
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”**

   \[
   \text{Detection Rate} = \frac{\text{True Detections} - \text{False Detections}}{\text{Total number of particles}} \times 100
   \]

<table>
<thead>
<tr>
<th>Series #</th>
<th>Distance between camera and calibration body [cm]</th>
<th>Total number of Particles on the Calibration body</th>
<th>Particles Size, [Pixels]</th>
<th>True detections [particles]</th>
<th>Number of false detections</th>
<th>Detection rate [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>120</td>
<td>980</td>
<td>2±1</td>
<td>967</td>
<td>3</td>
<td>98.3</td>
</tr>
<tr>
<td>2</td>
<td>80</td>
<td>354</td>
<td>5±2</td>
<td>352</td>
<td>0</td>
<td>99.4</td>
</tr>
<tr>
<td>3</td>
<td>60</td>
<td>210</td>
<td>15±3</td>
<td>207</td>
<td>1</td>
<td>98.1</td>
</tr>
<tr>
<td>4</td>
<td>30</td>
<td>60</td>
<td>40±5</td>
<td>59</td>
<td>1</td>
<td>96.6</td>
</tr>
</tbody>
</table>
Validation & Evaluation – Results

Test #2 - Comparison: 2D-PTV vs. PIV
Test #2 - Comparison: 2D-PTV vs. PIV
Validation & Evaluation – Results

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

![Graph showing comparison between Compressed PTV, Flow vis. vertical, Flow vis. horizontal, PIV U, diagonal, and PIV V, diagonal methods. The graph plots D/L vs. Re with data points and error bars. There is a linear regression line with the equation $-\frac{1}{3} \times 10^6 Re$.](image-url)
Validation & Evaluation – Results

Test #3 - 3D-PTV Comparison – Our vs. Off-the-shelf
The system can provide a solution for the remotely controlled tracking experiments:

- Microgravity
- Underwater
- Harsh experimental conditions
- 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