The Effect of Parameterization on a Reconfigurable Implementation of PIV

Abderrahmane Bennis, Miriam Leeser, Gilead Tadmor, Russ Tedrake
abennis@ece.neu.edu  mel@ece.neu.edu  tadmor@ece.neu.edu  Russt@mit.edu

Overview of the Strategic Research Plan

Bio-Med Enviro-Civil

Overview of the Strategic Research Plan

Validating TestBEDs

Implementation of PIV

Abderrahmane Bennis, Miriam Leeser, Gilead Tadmor, Russ Tedrake
abennis@ece.neu.edu  mel@ece.neu.edu  tadmor@ece.neu.edu  Russt@mit.edu

Algorithm

PIV Parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
<th>Hardware latency (sec)</th>
<th>Software latency (sec)</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Img_width</td>
<td>The width in pixels of the images</td>
<td>0.063</td>
<td>0.17</td>
<td>56</td>
</tr>
<tr>
<td>Img_depth</td>
<td>The depth in pixels of the images</td>
<td>0.01</td>
<td>0.65</td>
<td>65</td>
</tr>
<tr>
<td>Area_width</td>
<td>The depth in pixels of the interrogation window</td>
<td>0.04</td>
<td>2.2</td>
<td>55</td>
</tr>
<tr>
<td>Area_depth</td>
<td>The depth in pixels of the sub-areas</td>
<td>0.07</td>
<td>2.49</td>
<td>42</td>
</tr>
<tr>
<td>Sub_area_width</td>
<td>Number of pixels by which a sub-area is moved</td>
<td>0.09</td>
<td>3.98</td>
<td>43</td>
</tr>
<tr>
<td>Sub_area_depth</td>
<td>Number of pixels by which a sub-area is moved</td>
<td>0.060</td>
<td>1.15</td>
<td>57</td>
</tr>
<tr>
<td>Displacement</td>
<td>Displacement of the interrogation window</td>
<td>0.136</td>
<td>2.25</td>
<td>42</td>
</tr>
<tr>
<td>Pixel_bits</td>
<td>Number of bits that represent a pixel</td>
<td>0.185</td>
<td>3.97</td>
<td>43</td>
</tr>
<tr>
<td>RAM_width</td>
<td>Number of bits in each memory address</td>
<td>0.219</td>
<td>4.54</td>
<td>21</td>
</tr>
<tr>
<td>Overlap</td>
<td>Number of interrogation windows that overlap</td>
<td>0.247</td>
<td>5.04</td>
<td>20</td>
</tr>
</tbody>
</table>

Our design outperforms [1], last advanced circuit, by processing 333 pairs of images per second instead of 204.

References:

Last Achievement:
Sub-Pixel interpolation is added to the implementation

Technology transfer:
Techniques used in this research can be transferred to implement other image processing applications on FPGA all in a parameterized way.

Publications:
A. Bennis, M. Leeser and G. Tadmor, “Implementing a highly parameterized particle image velocimetry system on reconfigurable hardware”, Application-Specific systems, Architectures and Processors (ASAP), Boston, MA, July 7-9 2009.

Contributions:
- A highly parameterized digital PIV system is designed using VHDL.
- A highly parameterized digital PIV is implemented on a FPGA board.
- A library of parameterized VHDL components has been built.
- C++ implementation of parameterized PIV.

This work was supported in part by Gordon-CenSSIS, The Bernard M. Gordon Center for Subsurface Sensing and Imaging Systems, under the Engineering Research Centers Program of the National Science Foundation (Award Number EEC-9986821).

Goal
Accelerate the performance of highly parameterized Particle Image Velocimetry (PIV) algorithm using Field Programmable Gate Arrays

Abstract
Particle image velocimetry (PIV) is used in computational fluid dynamics to obtain a detailed localized view of velocity vectors in an unsteady fluid flow. In a nutshell, PIV might be used only when real-time response is possible. Up to now, there is no PIV system that combines both, flexible parameterization and high computing performance. In our research we are creating such a system.

We have designed a highly parameterized PIV implementation based on reconfigurable hardware that adapts to various setups and application domains. The circuit is parameterized by the dimensions of the captured images as well as the dimensions of the interrogation windows and sub-areas, pixel representation, board memory width, displacement and overlap. Through this work a parameterized library of different VHDL components has been built. For a typical PIV configuration with images of 512×512 pixels, 40×40 pixel interrogation windows and 32×32 pixel sub-areas, we achieved about 65x speedup in hardware over a standard software implementation.

This research is supported by the National Science Foundation.