Integration of a Bio-Inspired Robotic Vision System on FPGA

Laurent Fiack, Thomas Lefebvre and Benoît Miramond
ETIS Lab UMR 8051 CNRS / ENSEA / UCP
6 Avenue du Ponceau
Cergy-Pontoise
Email: {firstName.lastName}@ensea.fr

I. INTRODUCTION

This paper describes the hardware implementation of a bio-inspired visual system, intended to feed the neural network of a mobile robot. The visual system is implanted in an FPGA which communicates through Ethernet with a PC running the neural network. We worked on the implementation of the Gaussian pyramid [CR03] vision system.

This vision system is well known in the Cognitive science world but requires heavy computing power. It can’t be embedded as software respecting real-time constraints. That’s why we realized a hardware implementation through an algorithm/architecture adequacy process [VMM+08].

In this embodiment approach [Cla99], it seems natural that the first steps of the visual scene processing be performed by the robot itself as it is done in the mammalian retina.

II. THE ROBOTIC VISION SYSTEM

In image processing, a pyramid is a technique for multi-resolution analysis. In our case, it consists in a succession of smoothing and subsampling. There are two output images between two subsampling. These outputs are a difference of two images smoothed by Gaussian kernels. Thus, the Gaussian pyramid (Figure 1) acts as multiple band-pass filters.

The first step of the algorithm is to compute the gradient magnitude of the image. It simply consists in the sum of the X and Y gradients. The resulting image is then smoothed several times by different sampled Gaussian kernels. The DoG (Difference of Gaussians) is then the difference of the output of a Gaussian filter by the output of the previous one.

The next processing steps are shown in Figure 2, and are present for each output of the pyramid. The local maxima of the DoG images are first extracted and sorted by magnitude, the N most significant per DoG are considered as keypoints.

Once a keypoint is detected, its neighbourhood can be extracted and remapped. The Log-polar mapping is inspired from the mammalian retina (Figure 3). It is used to make the local feature robust to rotation and scale variance. Furthermore, it reduces the information quantity to be processed by the neural network.

III. HARDWARE IMPLEMENTATION

Generally in hardware vision, the pixels of an image come as a flow, and they are processed with the same rhythm. A number of pixels must then be stored, depending on the kernel size and the image width. The output pixels are produced with a fixed latency, depending also on these two parameters.

In the first part of his project, Lefebvre [Lef] proposed an interface for his processing IPs. The input and output buses are composed of the pixel value and their coordinates. A standard interface
allows for modularity of these IPs. It is then easy to assemble the processing IPs (gradient, gauss, dog and subsampling) to build the hardware Gaussian pyramid.

Custom IPs have been designed for the keypoint search, the sorting of their coordinates by the magnitude of the relative keypoint, and the log-polar mapping. Each IP works at a rhythm given by the continuous flow of pixel.

IV. RESULTS

The first stage of the pyramid have been tested, and works in real-time. The architecture supports the flow of 320 × 240 images at 25 frames-per-second given by the image sensor.

The synthesis results are given in Figure 4 for the Altera Cyclone IV EP4CE115 FPGA of the Terasic DE2-115. The FPGA has 114480 LEs (Logic Elements) and 3981312 Memory bits. These results are given for the complete pyramid, for 320 × 240 images, with 8-bit grey-scale pixels, 10 pixels research radius in high-, 5 in mid-, and 3 in low-resolution, and \( N = 32 \) keypoints by scale. For IPs that appear multiple times in the design, the total occupation is given. The footprint of the SOPC used to send images and keypoint coordinates via Ethernet is also given. The SOPC (Figure 5) is composed of a NIOS II/f cpu and its SDRAM controller, an Ethernet controller, a dual-port BRAM for storing images and PIOs for generating index and reading coordinates.

<table>
<thead>
<tr>
<th>IP</th>
<th>LEs</th>
<th>%</th>
<th>Memory bits</th>
<th>Memory %</th>
</tr>
</thead>
<tbody>
<tr>
<td>delay</td>
<td>138</td>
<td>&lt;1</td>
<td>57344</td>
<td>1</td>
</tr>
<tr>
<td>dog</td>
<td>630</td>
<td>&lt;1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>gauss</td>
<td>14756</td>
<td>13</td>
<td>221184</td>
<td>6</td>
</tr>
<tr>
<td>gradient</td>
<td>365</td>
<td>&lt;1</td>
<td>8192</td>
<td>&lt;1</td>
</tr>
<tr>
<td>search</td>
<td>43073</td>
<td>38</td>
<td>573440</td>
<td>14</td>
</tr>
<tr>
<td>sort</td>
<td>11934</td>
<td>10</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SOPC</td>
<td>15826</td>
<td>14</td>
<td>1941353</td>
<td>49</td>
</tr>
<tr>
<td>Total</td>
<td>86722</td>
<td>76</td>
<td>2801513</td>
<td>70</td>
</tr>
</tbody>
</table>

V. CONCLUSION

With this work, the presented vision system, currently deported on external workstations, will be embedded in mobile robots. The implementation allowed to validate the image processing IPs.

The log-polar mapping IP, currently in development, has to be finalized for on-robot validation of the FPGA vision system. Tests are planned to make the architecture working with higher resolution images.

REFERENCES

[Lef] Thomas Lefèvre. Exploration architecturale pour la conception d’un système sur puce de vision robotique, adéquation algorithme-architecture d’un système embarqué temps réel. Université de Cergy-Pontoise.