

Low Power Multiple Object Tracking and Counting using a SCAMP Cellular Processor Array

David R. W. Barr, Stephen J. Carey and Piotr Dudek

School of Electrical & Electronic Engineering
The University of Manchester, Sackville Street Building, Manchester, M13 9PL, United Kingdom
e-mail: david.barr@manchester.ac.uk; s.carey@manchester.ac.uk; p.dudek@manchester.ac.uk

Abstract - A low-power demonstration system using a SCAMP-3 vision chip to track and count multiple objects with unpredictable trajectories is presented. The system can track as many discrete objects that can fit into its visual field. The compact, self-contained hardware consists of a battery, an ARM Cortex-M3 co-processor, and the sensor/processor array device. The tracking algorithm is performed entirely by the processor array and the complete system draws 7.3mA during operation.

Index Terms – Vision Chip, Processor Array, SIMD, Smart Sensors, Multiple Object Tracking

I. INTRODUCTION

Multiple object tracking is a challenging research topic with activity from the robotics, image processing [1], and cognitive research fields [2]. The difficulties of segmenting and tracking multiple objects in real-time are often due to the limitations of the image processing system employed. In a conventional image processing architecture, the scene is sampled via a photo-sensor, digitised, and then processed (mostly) sequentially by a Digital Signal Processor (DSP). This approach requires power consuming, high speed, data transfer throughout the system. To make matters worse, the actual processing needs of the algorithms used leave the processor under-utilised, as it is typically burdened with many simple, repetitive, atomic operations. Thus, this paper demonstrates an integrated sensing/tracking, low power solution developed using a focal plane cellular processor array – or ‘vision chip’ [3].

A vision chip can perform pixel-parallel processing of a scene, as each photo-sensor is tightly coupled to a complete processing element, replete with memory, arithmetic logic unit, and local neighbourhood communications architecture (*fig 1*). There is no need for a local program memory, as each pixel’s processing element (PE) will execute the same instruction at the same time, relying on local (to the individual PE) autonomy to govern whether or not the globally broadcast operation actually takes place. In this respect, a vision chip operates akin to a Single Instruction Multiple Data (SIMD) processing architecture. Conventional processing requires the programmer to think largely in sequence (temporally). A vision chip requires the programmer to additionally think of managing processor and memory resources spatially, as the

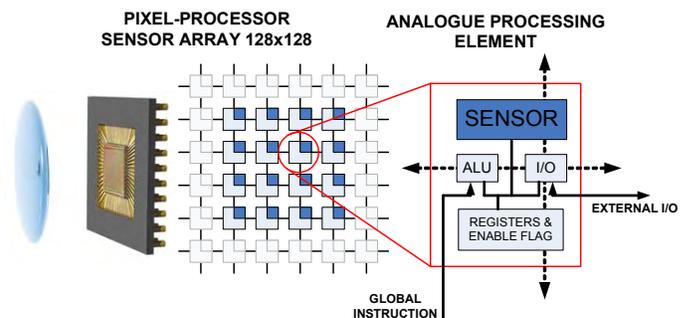


Figure 1 - Architecture of the SCAMP-3 vision chip

locations of the PEs within the processor array can be exploited to represent features.

This demonstration shows multiple objects are identified, tracked and counted by a vision chip. By storing the positions of objects and other describing features physically in ‘processor space’, rather than in a conventional ‘array of feature vectors’, further computational optimisations can be attained.

The vision chip used is a SCAMP-3 analogue processor array [4]. Power savings are made by performing the computation in the analogue domain without recourse to reading a frame to an external processor. Only the results of the image processing are read out from the vision chip, saving both power and bandwidth.

II. DEMONSTRATION

The new SCAMP system [5] being demonstrated is a self-contained, battery powered unit with a SCAMP-3 vision chip and an ARM Cortex-M3 control processor [6], designed with power consumption in mind. The ARM is responsible for issuing instructions globally to the vision chip, but the computation for tracking and counting objects is performed by the vision chip alone. When the vision chip senses a novel object has entered its field of view, it issues a signal to an external counting display box. The display box is self-powered, and is only a visual aid to the demonstration.

Attached to the SCAMP-3 system, is a current meter which displays the current consumed by the whole system. The system is given the task of counting how many whole toy ‘bugs’ enter the field of view of the sensor. The bugs move chaotically, changing direction randomly and collide into each

other. They also may only partially enter the field of view, in which case should not be counted.

III. TRACKING ALGORITHM

The objective of this algorithm is to count how many objects enter the “arena”, which is an area governed by the boundary of the visual field of the vision chip system. There are three additional constraints. 1) Only whole objects must be counted, i.e. an object partially entering and exiting the arena, or straddling the boundary should not be counted. 2) Objects must only be counted once. 3) An object is still tracked if it partially leaves the arena, or skirts the boundary. To satisfy these constraints it is necessary to identify the introduction of a whole novel object, count it, and then track its movements within the arena until it has fully exited it.

Figure 2 shows the state of the processor-array at various stages during the algorithm. (a) is the previous object history memory, i.e the result of the tracking from the previous iteration. (b) is the most recently acquired input frame from the photo-sensors. Although 5 objects can be seen, only 3 of them have history, thus two maybe novel objects. (c) shows how objects are associated with history. An object has history if the input image intersects the history image. (d) shows the intersections, which are the starting point of a flood-fill (repeated dilation and intersection tests) operation into the input image (e). (f) filled objects are removed from the input image, leaving just novel objects. A boundary is created (g)

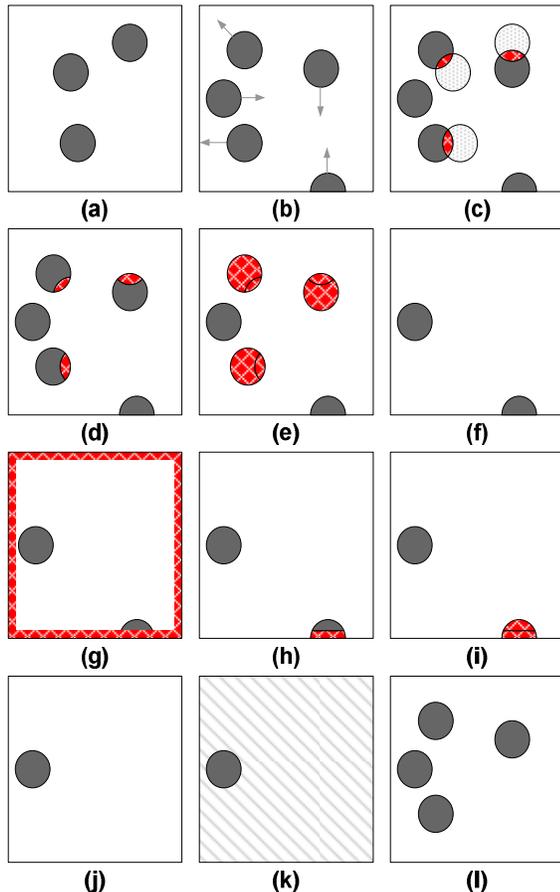


Figure 2 - The tracking algorithm. See text for details.

which is used to test for intersection with the remaining input image. (h) shows the intersection, thus the input object must be touching the boundary and is not guaranteed to be whole, thus must be filled (i), and removed (j). A global readout [7] test performed, to see if any input pixels remain (k). If there is, then a novel object has entered the visual field and must be whole, therefore counted. Finally (l) is the history register updated (with whole objects) for the next iteration.

However, the success of this algorithm is dependent upon the speed of the objects. If the objects move too quickly in-between frame acquisitions then input may not intersect with the history, causing erroneous novel objects. Also, if two objects enter the arena at the same time only a single object is counted. This could be corrected by performing area based read outs until you find a pixel that belongs to an object, then use this seed to “flood-fill” into the object, thus removing it. This sequence should be repeated in a quad-tree search fashion until there are no objects left, although the additional flood-filling is costly.

There are several interesting observations. There is no need for lists of feature vectors (coordinates, ids). This guarantees that the algorithm is deterministic in length regardless of how many objects are being tracked. The costly iterative flood-fill operation can be reduced to track objects of a known size. Our most recent vision chips [8] have asynchronous propagation flood-fill operations making the cost negligible, and pixel address extraction to quickly locate a pixel belonging to an object, both of which would significantly reduce the algorithm’s execution time, thus save even more power.

IV. RESULTS

To reduce the power consumption, a range of tactics were adopted. The ARM, slowly clocked at 25MHz issues instructions at 400KHz to the SCAMP-3. The frame capture rate is 8 fps (adequate for accurate bug-counting). Using a 3.96V supply, the average total current drawn by the entire system is 7.3mA. The processing time per frame was 12 ms, and the system slept for 113ms. During processing, the average current consumed is 16mA, and when sleeping 1.25mA.

REFERENCES

- [1] T. Lochmatter, P. Roduit, C. Cianci, N. Correll and J. Jacot et al. SwisTrack, “A Flexible Open Source Tracking Software for Multi-Agent Systems.” IEEE/RSJ 2008 (IROS 2008), Nice, France
- [2] Z.W. Pylyshyn and R.W. Storm. “Tracking multiple independent targets: Evidence for a parallel tracking mechanism”. Spatial Vision, 1998
- [3] L. Carranza, et al., “ACE 16k based stand-alone system for real-time pre-processing tasks”. VLSI Circuits and Systems II, May 9-11 2005
- [4] P. Dudek, “Implementation of SIMD vision chip with 128x128 array of analogue processing elements”, ISCAS 2005.
- [5] P.Dudek, D.R.W.Barr, A.Lopich and S.J. Carey, “Demonstration of real-time image processing on the SCAMP-3 vision system”, IEEE International Workshop on Cellular Neural Networks and their Applications, CNNA 2006, pp.13-13, Istanbul, August 2006
- [6] S.J.Carey, D.R.W.Barr and P.Dudek, “Low Power Image Processing System using a SCAMP3 Vision Chip”, ACM/IEEE International Conference on Distributed Smart Cameras, ICDS 2011, Ghent
- [7] P. Dudek, “A Flexible Global Readout Architecture for an Analogue Vision Chip”, IEEE International Symposium on Circuits and Systems, ISCAS 2003, Bangkok, Thailand, vol.III, pp.782-78
- [8] S.J.Carey, A.Lopich and P.Dudek, “A Processor Element for a Mixed Signal Cellular Processor Array Vision Chip”, ISCAS 2011, Rio de Janeiro, May 2011