Feature Extraction using a Portable Vision System

Jianing Chen, Stephen J. Carey and Piotr Dudek

***Abstract*— We demonstrate high-speed low-power feature extraction implemented on a portable vision system based on the SCAMP-5 vision chip. This embedded system executes a parallelized FAST16 corner detection algorithm on the vision chip's programmable on-focal-plane processor array to detect the corner points in the current image frame. The coordinates of these corner points are then extracted using the vision chip's event address readout infrastructure. These coordinates, as sparse data, can then be output through any of the IO buses within the micro-controller in the vision system at low latency. The USB-powered (400mA) system is capable of outputting 250 features at 2300 frames per second (FPS) in ideal lighting conditions, while 1000 FPS can be achieved in an indoor environment. The system can be applied to the real-time control of agile robots and miniature aerial vehicles.**

I. Introduction

This paper presents a feature extraction system implemented on a portable vision system (see Fig.1) based on the SCAMP-5 vision chip [1]. SCAMP-5 is an on-focal-plane SIMD processor array, in which each of the 256×256 pixels of the image sensor array has a processor. The processors can perform basic operations across its own registers, as well as accessing the registers of its 4 connected neighbors. The SIMD processor array provides a high-performance computational capability, and together with an ARM micro-controller is capable of executing vision algorithms in real-time, at high frame rates. A feature extraction algorithm has been developed for the vision system to output the coordinates of the features in view. Compared to an image, feature points require lower bandwidth to transfer and typically take much less computation power to utilize. Thus, feature based vision algorithms are well suited for embedded systems that are constrained by size, such as miniature aerial vehicles.

II. System Description

A. Feature Detection

The system uses corners in the image as features. We implemented the FAST corner detection algorithm [2]. In the algorithm, a corner pixel is determined by having a number of connected neighbor pixels with similar intensity. For FAST-16 in particular, 16 neighboring pixels in a circular pattern are examined for each of the pixels in the image. In this system, a fixed threshold is used to determine whether the intensity level of a neighbor pixel is similar to the center pixel. If a pixel has at least 10 connected neighbor pixels that are not similar, the pixel is identified as a corner. This principle is illustrated in Fig.2. Note, in this method, both

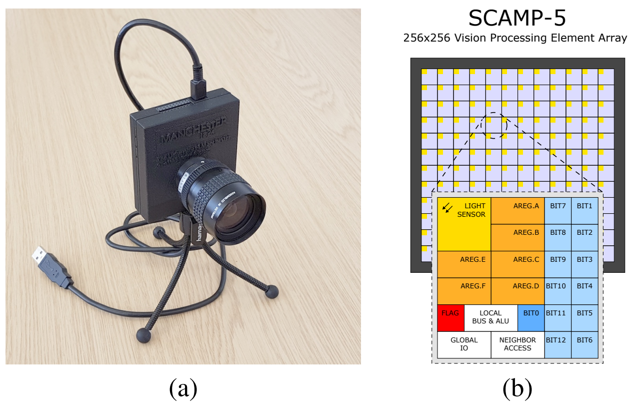

Fig. 1. (a) The 'Scamp5d' vision system. It weighs around 100 grams (excluding lens). Depending on the running program, the power consumption of the system is around 1.5 to 2.5 Watts. (b) The architecture diagram of the SCAMP-5 vision chip. The pixel-processors in the SIMD array contain local memory and have capability to perform arithmetic/logic operations.

dark corners on a bright background and the reverse can be detected.

This algorithm has been implemented in parallel for SCAMP-5's processor array. After the image frame is captured, each of the processing elements (PE) will save the light intensity and compare it with the values saved by the PE's 16 FAST neighbors one-by-one. If the result of this comparison remains the same as the previous comparison, a counter will be increased by one. Otherwise the counter will be reset to 0. When this counter is equal to 10, this pixel is flagged as a corner in a feature map. When this counter reaches 15, this flag is cleared and the counter is set to 0. Since FAST16 can produce a few corner pixels for each visual corner in an image, the feature map was post-processed to reduced the size of blobs to a single pixel using an erosion process.

B. Feature Extraction

In addition to the parallel processor array, the SCAMP-5 vision chip has an event-readout infrastructure, which can scan the coordinates of all '1's in a Boolean image (events). The time cost of the event-readout is proportional to the number of points. In contrast, if such a process is done in serial using a conventional CPU, the time cost would be proportional to the size of the image.

Event-readout was applied to extract the coordinates of the features in the feature map obtained after the feature detection process. Since the feature extraction is done by the vision chip, the data output is only 2 bytes per feature instead of a 256×256 image. Thus, very low communication bandwidth is necessary for transferring the feature data, even at high equivalent video frame rates.

This work has been supported by EPSRC Grant EP/M019284/1.

The authors are with the School of Electrical & Electronic Engineering, The University of Manchester, Manchester M13 9PL, United Kingdom

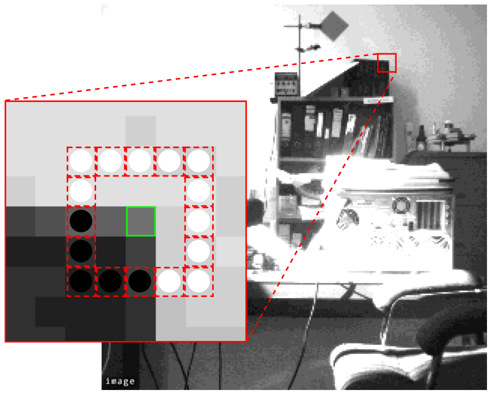

Fig. 2. FAST16 Corner Detection with a modified neighbour pattern. In the image, the centre pixel has 11 connected neighbors that have different intensity levels to itself, thus this pixel is identified as a corner pixel.

III. Results

A. Performance

The time cost of the algorithm was measured. The duration of the feature detection algorithm with the erosion process was 292 μs while the event-readout process took 140 μs when the system was configured to scan 250 points. Thus, the feature extraction system could run at 2.3 kHz (i.e. process 2300 frames per second). It has to be noted that operation at such high frame rates requires sufficient lighting conditions. In the current system, the 2.3 kHz operation is possible in a sun-lit outdoor environment. When the system was configured to operate at 1 kHz, the exposure time was sufficient enough for an indoor environment with daylight coming through windows.

When the algorithm is modified to detect only corners of dark objects against a bright background, the time cost of the feature detection algorithm is 190 μs and the system can run at 3 kHz. The time taken to scan the feature points increases linearly to the maximum number of points. When 100 points are scanned, the time-cost is 59 μs.

The high frame rate operation is highly advantageous. In particular feature tracking is simplified since the features do not change significantly from frame to frame, hence the correspondence problem can be reduced to finding a nearest neighbor, even in presence of agile motion.

In a real-time control application, latency is as important if not more important than frame-rate. Since our system provides as output the coordinates of points of interest (e.g. 500 bytes for 250 feature points), the time to transfer the data is much reduced, as compared with a full image transfer. This benefits both frame-rate and data latency. Experiments were done using a 10 Mbps SPI bus connecting the vision system to a single board computer (Odroid XU4). The latency was measured as the time between the end of exposure in the vision system to a LED signal output of the single board computer. For 250 points, this time interval was 1260 μs with a standard deviation of 35 μs.

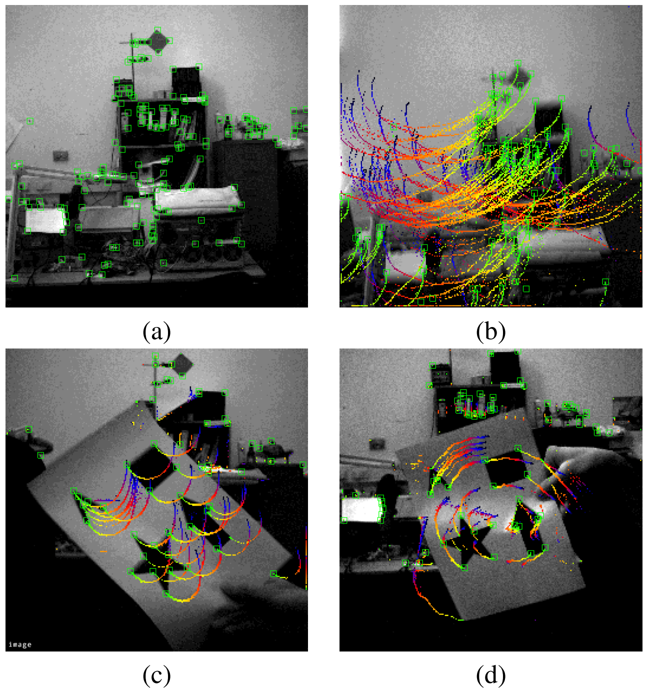

Fig. 3. Features in a view when the system was running at 480 Hz with a 30 FPS image display enabled. The gray-level images are quantized to 16-levels. The point trails in the images are produced by drawing all acquired events (feature points) in a time window of 0.5 s, using a color map to indicate the age of the event, most recent events are in green. (a) A static scene. (b) A scene when the system was being moved rapidly. (c) A moving object in front of the static background. (d) A spinning object in front of the static background.

B. Demonstration

In the demonstration, the vision system was connected to a PC running Windows 10 via a USB2.0 port. In the setup, the micro-controller in the vision system was capable of transferring all the feature data in synchrony with the algorithm loop, at 3 kHz. The power consumption of the entire vision system, running at 2.3 kHz was measured at 2 Watts (powered entirely via USB, with current of 400 mA at 5V).

The system normally returns only the coordinates of the feature points. Grayscale image output can be enabled for inspection and debug purposes while the algorithm loop is running at 480 Hz. This is achieved by outputting one binary image in each loop using a threshold changing periodically over 16 values. The binary images are then blended to create a 16-level grayscale image on the hosting PC. Thus the visual frame-rate of the display image is equivalent to 30 FPS, while all the features data are still output at 480 Hz frame rate. Fig.3 shows the features and their trails drawn on top of the captured image, as presented by software on the host PC.

References

[1] S. Carey, D. Barr, B. Wang, A. Lopich, and P. Dudek, "A 100,000 fps vision sensor with embedded 535 gops/w 256×256 simd processor array," in *Proc. 2013 IEEE Symp. VLSI Circuits*, Jun 2013, pp. 182–183.

[2] E. Rosten and T. Drummond, "Fusing points and lines for high performance tracking," in *Proc. Tenth IEEE Int. Conf. Computer Vision*, vol. 2, Oct 2005, pp. 1508–1515.