Keywords: real-time image processing applications, SIMD architectures, artificial vision, active contour, sensing and robot vision.

Abstract

An artificial high-level vision agent for the localisation and description of the movements of the SPIDER robot arm of the EUROPA/PAT systems during its operations is presented. The agent uses real-time image processing operations implemented on SIMPiL (SIMD Pixel Processor) to manage the large computational workloads and I/O throughput requirements. The vision agent, which includes a description agent, generates the perception grounded predicates obtained by image sequences and provides a 3D estimation of the arm movements. This provides the scientist user of SPIDER with meaningful feedback of his operations on the arm during a scientific experiment. This work evaluates the effectiveness of the SIMPiL design on this type of important image applications. Preliminary simulation results support the design choices and suggest that more complex, multistage applications can be implemented to execute at real-time frame rates.

1. Introduction

An artificial high-level vision agent for the localisation and description of the movements of the SPIDER robot arm of the EUROPA/PAT systems [7, 9] during its operations is presented. The agent uses some real-time image processing operations implemented on SIMPiL. The SIMD Pixel Processor (SIMPiL) [17], is a focal plane imaging system, being designed at Georgia Tech, which incorporates a specialized SIMD architecture with an integrated array of image sensors. Many SIMD systems [11, 12, 13] have been used for image processing before, but they achieve performance and generality at the expense of I/O coupling and physical size. Advances in current semiconductor technology have enabled the integration of CMOS image sensors with digital processing circuitry [15]. In a focal plane imaging system, images are optically focussed into the sensor array, or focal plane, and are made available to the underlying processing engine in a single operation. Because large amount of data parallelism is inherent in low level image processing applications, SIMD (Single Instruction Multiple Data) architectures provide an ideal programming model [16].

This work is an improved and powerful extension of the vision agent presented in [2, 3]. The described software module is related to the interpretation of sensory data in the framework of an ASI project. It aims at the application of AI techniques to the design and realisation of an effective and flexible system for the supervision of the SPIDER. The arm (a 7 d.o.f. robot arm) will work on board of the International Space Station (ISS) [4]. The framework project is a Italian three years research project sponsored by the Italian Space Agency (ASI) and involving AI researchers from the Universities of Rome, Turin, Genoa, Palermo, Parma, from the IP-CNR of Rome and from the IRST of Trento. The main aim of the vision agent is the advancement of the state of the art in the field of artificial vision for space robotics by introducing and integrating artificial vision techniques that offer opportunity for providing the SPIDER arm operations with effective great degrees of autonomy. In the following, Sect. 2 describes the SIMPiL architecture used to support the vision agent and its components. Sect. 3 describes the low-level processing operations, the utilisation of an active contour technique to individuate and segment the shape of the SPIDER arm, and performing the 3D reconstruction of the location of its links. Sect. 4 deals with the SIMPiL implementation details. Sect. 5 describes the experimental results and the performance of the system. Finally, Sect. 6 outlines some conclusions and future development.

2. The SIMPiL architecture

The SIMD Pixel Processor (SIMPiL) architecture consists of a mesh of SIMD processors on top of which an array of image sensors is integrated. A diagram for a 16-bit implementation is illustrated in Fig. 1. Each processing element includes a RISC load/store datapath plus an interface to a 4x4 sensor subarray. A 16-bit datapath has been implemented
which includes a 32-bit multiply-accumulator unit, a 16 word register file, and 64 words of local memory (the ISA allows for up to 256 words). The SIMD execution model allows the entire image projected on many PEs to be acquired in a single cycle. Large arrays of SIMPiP PEs can be simulated using the SIMPiL Simulator, an instruction level simulator. Early prototyping efforts have proved the feasibility of direct coupling of a simple processing core with a sensor device [17]. A 16 bit prototype of a SIMPiP PE was designed in 0.8 µm CMOS process and fabricated through MOSIS. A 4,096 PE target system has been used in the simulations. This system is capable to deliver a peak throughput of about 1.5 Tops/sec in a monolithic device, enabling image and video processing applications that are currently unapproachable using today’s portable DSP technology.

The SIMPiL architecture is designed for image and video processing applications. In general, this class of applications is very computational intensive and requires high throughput to handle the massive data flow in real-time. However, these applications are also characterized by a large degree of data parallelism, which is maximally exploited by focal plane processing. Image frames are available simultaneously at each PE in the system, while retaining their spatial correlation. Image streams can be therefore processed at frame rate, with only nominal amount of memory required at each PE [19].

![SIMPiL architecture](image)

Figure 1. The SIMPiL architecture.

3. Localisation and description of the movements of a robotic arm

The artificial vision agent generates the exact 2D description of the arm movements obtained by image sequences, providing the necessary information to perform 3D estimation of the arm movements, thus allowing the scientist user of SPIDER to receive meaningful feedback of his operations on the arm during a scientific experiment. This work is an improved and powerful extension of the vision agent presented in [2, 3].

The capabilities of the vision agent include the followings:
- individuate and segment the SPIDER arm also in contrasted and irregular backgrounds;
- perform a 3D estimation of the position of the arm by camera images;
- interpret complex movements of the arm acquired by a camera in terms of symbolic in terms descriptions.

The implemented computer vision agent is based on three main components:
- the perception component;
- the scene description component;
- the visualisation component.

The perception component of the agent processes the image data coming from a video camera that acquires the operation of the robotic arm and take advantage from SIMPiL system to process the images. The main task of this component is to estimate the position of the arm in the acquired image. It should be noted that the estimation, which is generated solely by the visual data, may be useful also for fault identifications of the position sensor placed on the joints of the arm.

3.1. Active contour for the extraction of the arm shape.

The images acquired by the camera are processed by the contour module that extracts the arm contours by a suitable algorithm based on active contours. This task is resolved by two step:
- contour initialisation and individuation of the set of points describing the robotic arm;
- tracking of the arm in the sequence.

In order to extract just the information related to the arm, some filtering operations are performed:
- noise reduction by 5x5 median filter;
- automatic tresholding;
- subtraction of image intensities between two sequentially frames.

The arm shape is extracted in a simple way and in short time using the evolution of the active contour using the GVF (Gradient Vector Flow) methodology [10].

3.2. GVF Snake model.

The snake is a deformable curve that moves in the image under the influence of forces related to the local distribution of the grey levels [8, 10]. When the snake reaches an object contour, it is adapted to its shape. In this way it is possible to extract the object shape of the image view. The snake as an open or closed contour is described in a parametric form by:

\[ v(s) = (x(s), y(s)) \]

where \( x(s), y(s) \) are \( x,y \) coordinates along the contour and \( s \) is the normalized arc length:

\[ s \in [0, 1] \]

The snake model defines the energy of a contour, named the snake energy, \( E_{snake} \) to be:
\[ E_{\text{snake}}(v(s)) = \int_0^1 (E_{\text{edn}}(v(s)) + E_{\text{image}}(v(s)))ds \]

The snake used in this framework has only edge functional which attracts the snake point at high gradient:

\[ E_{\text{image}} = E_{\text{edge}} = -(G_\sigma \ast \nabla^2 I(x,y))^2 \]

Figure 2: Gradient Vector Flow technique applied to capture SPIDER arm shape.

This is the image functional proposed by Kass [8]. It is a scale based edge operator that increase the locus of attraction of energy minimum. \( G_\sigma \) is a Gaussian of standard deviation sigma which controls the smoothing process prior to edge operator. Minima of \( E_{\text{edge}} \) lies on zero-crossing of \( G_\sigma \ast \nabla^2 I(x,y) \) which defines edges in Marr-Hildreth [5, 6] theory.

Scale space filtering is employed, which allows the snake to come into equilibrium on heavily filtered image, and then the level of filtering is reduced, increasing the locus of attraction of a minimum. The gradient vector flow (GVF) field is defined to be the vector field \( v(x,y) = (u(x,y), v(x,y)) \) that minimizes the energy functional:

\[ E = \int \int \mu (u_x^2 + u_y^2 + v_x^2 + v_y^2 + |\nabla f|^2 |\nabla f|^2) \, dx \, dy \]

Figure 3: Evolution and final result of the GVF contour module.

Particular advantages of the GVF snake over a traditional snake are its insensitivity to initialisation and ability to move into concave boundary regions (for example see figure 1). Another useful computational feature is that the external force is defined statically, unchanged in time and independent from position of the contour itself. In this way the calculation of the external force is done at the beginning and is unchanged during contour evolution. The external force moving the contour is based on an edge extraction function with uniform absolute value. This choice in fact addresses the problem of weakly vector field near concavities of the followed contour. Starting from arbitrary location of a set of points, the contour is captured after some iterations, when the equilibrium of the forces is reached [8].

3.3. Localisation and segmentation of the arm shape.

The implemented snake allows to extract the arm shape in a simple way and in short time. Fig. 3 shows the evolution and result of the contour module. From extracted arm snake it is possible to estimate the position of the links of the arm in the image plane, i.e., without the depth information (see figure 3.a).

The arm links of the arm are located in two steps:

- individuation of lines that characterise the different segments of the arm and grouping based on parallelism;
- localisation of links by intersection of the lines.

3.4. Reconstruction of 3D position of the arm links

Let us consider a generic link \( i \) of the arm at time \( t \); the link is characterized by its 3D coordinates \( [x_i(t), y_i(t), z_i(t)] \). A generic posture of the SPIDER arm at time \( t \) is characterized the vector \( x(t) \) which individuates the seven links of the arm:
The snake information allows us to estimate the first coordinates of each link, i.e., their projection in the image plane:

\[
x(t) = \begin{bmatrix}
(x_1(t), y_1(t), z_1(t)) \\
(x_2(t), y_2(t), z_2(t)) \\
\vdots \\
(x_7(t), y_7(t), z_7(t))
\end{bmatrix}
\]

The third coordinate is obtained using standard computer vision algorithms [5, 6]. A generic 3D point X generates the point w on image:

\[
\hat{X} = P\hat{X} = K[R | t]\hat{X}
\]

where P is the 3x4 projection matrix, decomposable on calibration matrix K, rotation matrix R and translation vector t. X and w are indicated using homogeneous coordinates. The seven link points are used to estimate the projection matrix and to perform triangulation on the images of the sequence. The process of reconstruction supposes that the reference coordinate system is placed on link which is fixed (label 8 in figure 3), and the distances between links are known. In this way we obtain the position in the reference system of the links (see fig. 4a).

The implemented system also works if the image sequence does not show all the seven links (depending from arm position respect to camera): the recovery the position of the hidden links it is simply performed using the data reconstructed of the visible ones. In this way the following scene description component can aim only on the classification of the movements. Moreover, it is generated a graphic 3D representation of the arm movements receiving as input the data coming from the perception-reconstruction agent [1, 2, 3] (see fig. 4b). This representation provides the system with an immediate, visual feedback of the arm operations that complements the symbolic description coming from the previous component.

4. SIMPil implementation detail

The described artificial vision agent uses a series of low-level image processing applications that have been developed and simulated on the SIMPil
Simulator. The system uses images with size 256x256 pixels and 256 grey levels. The low-level image processing is performed in parallel by every PE. Each PE is associated to a portion of the whole image (with PPE = 16) and associated to the two directions x and y.

A reference frame is chosen from the video sequence every ten images in order to subtract static components of the scene (background removal). Every frame is preliminarily processed using 3x3 median filter, thresholding and sharpening. In this way the images have only the data related to the moving robotic arm (see fig. 5).

The implementation of the snake also required the computation of the gradient of the image. The magnitude of the gradient can be approximated by the expression:

\[ \nabla f \approx \left| (x_1 + 2x_2 + x_3) - (x_1 + 2x_2 + x_3) \right| + \left| (x_1 + 2x_2 + x_3) - (x_1 + 2x_2 + x_3) \right| \]

and it is calculated by the two following masks (Sobel operators):

\[
\begin{array}{ccc}
1 & -2 & -1 \\
0 & 0 & 0 \\
1 & 2 & 1 \\
\end{array}
\]

\[
\begin{array}{ccc}
-1 & 0 & 1 \\
-2 & 0 & 2 \\
-1 & 0 & 1 \\
\end{array}
\]

4.1. The snake evolution.

The movement of the snake is implemented on SIMPil in order to complete the whole automatic process of 2D automatic localization of arm contour. The snake is initialized on the border of image after the background removal phase. Two gradient masks (horizontal and vertical directions) are calculated and, where no information is available are added components in order to move the active contour in direction of the centre of the image. The SIMPil implementation of the movements of the snakes is based in some tokens that are transferred from a PE to another following the dominant direction of the image gradient. After few iterations, the tokens representing the points of the snake are placed on the edge of the object. Naturally, after the first capture of object contour, the tokens in the new frame start from the last calculated position.

5. Results

The applications described in the previous section have been developed and simulated on the SIMPil Simulator. Two facts can be considered in order to evaluate the goodness of the system:

- performance: at 500 MHz the time requested to process two images is 63\mu sec allowing for frame-rate processing;
- high degree of parallelism: as it is showed in figure 6 the number of communication between PEs is low, while memory operations and arithmetic operations are prevalent (see fig. 6).

6. Conclusion

The research demonstrated how the implemented artificial high-level vision agent may be an effective tool that helps the user scientist of the Spider arm to monitor his own operations by providing high-level feedback descriptions of the arm movements during scientific experiment. The SIMPil system architecture has been used to offer the required computational power for real time image processing operations. The application set of the vision agent is described, and performance analysis is given. Simulation results support the design choices made and suggest that more complex, multistage applications can be implemented on SIMPil to execute at real-time frame rates. The product of the described activity may be employed in all areas where ich the interactive autonomy of a space robotic system is a mandatory requirement, as the exploration of the Moon and Mars.

5

Figure 6: Table of the performances for the agent implemented.
References