



➤ Algorithm for Detecting a Bright Spot in an Image

Corrections for background intensity and dark current are included.

NASA's Jet Propulsion Laboratory, Pasadena, California

An algorithm processes the pixel intensities of a digitized image to detect and locate a circular bright spot, the approximate size of which is known in advance. The algorithm is used to find images of the Sun in cameras aboard the Mars Exploration Rovers. (The images are used in estimating orientations of the Rovers relative to the direction to the Sun.) The algorithm can also be adapted to tracking of circular shaped bright targets in other diverse applications.

The first step in the algorithm is to calculate a dark-current ramp — a correction necessitated by the scheme that governs the readout of pixel charges in the charge-coupled-device camera in the original Mars Exploration Rover application. In this scheme, the fraction of each frame period during which dark current is accumulated in a given pixel (and, hence, the dark-current contribution to the pixel image-intensity reading) is proportional to the pixel row number. For the purpose of the algorithm, the dark-current contribution to the intensity reading from each pixel is assumed to equal the average of intensity readings from all pixels in the same row, and the factor of proportionality is estimated on the basis of this assumption. Then the product of the row number and the factor of proportionality is subtracted from the read-

ing from each pixel to obtain a dark-current-corrected intensity reading.

The next step in the algorithm is to determine the best location, within the overall image, for a window of $N \times N$ pixels (where N is an odd number) large enough to contain the bright spot of interest plus a small margin. (In the original application, the overall image contains 1,024 by 1,024 pixels, the image of the Sun is about 22 pixels in diameter, and N is chosen to be 29.)

The window is placed at a given position within the overall image. A weighted average of the intensities of the $4N - 4$ outer pixels of the window is taken as an estimate of background intensity and subtracted from a weighted average of the intensities of the remaining inner $(N - 2) \times (N - 2)$ pixels of the window to obtain a background-corrected weighted sum of pixel intensities for the window. The weighted averages are simply pixel-intensity averages multiplied by common denominators so as to obviate floating-point arithmetic operations and thereby accelerate computations. The window is then moved to an adjacent column position, and weighted averages for the new position are calculated from the previous weighted averages by adding the appropriate values for the new outer and inner pixels and subtracting the corre-

sponding values for the pixels that have been left behind or changed in status between the inner and the outer. This process is repeated until the computations have been performed for all possible window positions. The position that yields the highest background-corrected weighted sum of pixel intensities is assumed to contain the bright spot of interest (the image of the Sun in the original application), and the window is then used to locate the bright spot more precisely as described next.

Within the inner $(N - 2) \times (N - 2)$ portion of the window, the position of the bright spot is determined by means of a simple centroid calculation, using the background-corrected pixel intensities. Because the window position selected as described above may not necessarily be the optimum one, the centroid calculation is performed twice in an iterative process: For the second centroid calculation, the window is re-centered on the centroid determined by the first centroid calculation.

This work was done by Carl Christian Liebe of Caltech for NASA's Jet Propulsion Laboratory.

The software used in this innovation is available for commercial licensing. Please contact Karina Edmonds of the California Institute of Technology at (626) 395-2322. Refer to NPO-41801.

➤ Extreme Programming: Maestro Style

Modifications have been made to suit a specific development environment.

NASA's Jet Propulsion Laboratory, Pasadena, California

“Extreme Programming: Maestro Style” is the name of a computer-programming methodology that has evolved as a custom version of a methodology, called “extreme programming” that has been practiced in the software industry since the late 1990s. The name of this version reflects its origin in the work of the Maestro team at NASA's Jet Propulsion Laboratory that develops software

for Mars exploration missions.

Extreme programming is oriented toward agile development of software resting on values of simplicity, communication, testing, and aggressiveness. Extreme programming involves use of methods of rapidly building and disseminating institutional knowledge among members of a computer-programming team to give all the members a shared

view that matches the view of the customers for whom the software system is to be developed. Extreme programming includes frequent planning by programmers in collaboration with customers, continually examining and rewriting code in striving for the simplest workable software designs, a system metaphor (basically, an abstraction of the system that provides easy-to-remem-