

## MEDIAN CALCULATION USING SIMD OPERATIONS

### TECHNICAL FIELD

This invention relates to methods and systems of finding median values from sets of values, using computers and other processing systems that are capable of single-instruction/multiple-data (SIMD) operations.

### BACKGROUND OF THE INVENTION

Computers almost invariably utilize raster-type display devices for presenting information to users. A cathode ray tube (CRT) is an example of a raster-type display device. In such display devices, images are composed of a plurality of visible picture elements or dots, commonly referred to as pixels. The pixels are arranged in a two-dimensional array having rows and columns. Each pixel has a single color, specified from a large palette of colors. When spaced very closely, the individual pixels are indiscernible to a human viewer, and the image appears to have been painted in continuous tones over the surface of the display device.

The discrete nature of the pixels is useful when representing images in digital formats—each pixel can be conveniently specified as a number that corresponds to a specific color. However, in the process of capturing an image into digital form, various sources of noise may degrade the image. This noise causes the pixel values to deviate from their “true” values. Computers often employ some form of filtering to remove noise such as this from viewed images. One method of filtering is to replace each pixel with the average of its adjacent pixels. Another filtering method involves replacing each pixel with the median of its adjacent pixels. A median filter is typically used to mitigate the effects of “impulse” noise or “shot” noise.

To illustrate this process, FIG. 1 shows a matrix or array of individual pixels, wherein each pixel is represented as a grid square. Five of the pixels are arbitrarily labeled as pixels a, b, c, d, and e.

Assume that in the example of FIG. 1, it is desired to filter pixel d. In this example, the pixel d will be replaced by the average of it and its orthogonally-adjacent pixels a, b, c, and e. Thus, filtering pixel d involves finding the average pixel value from among pixels a, b, c, d, and e. This task is repeated for every single pixel of the image. One way to accomplish this, in conventional microprocessors, is to assign the five pixel values to five different registers, add the registers together, and divide the result by five. However, this can consume significant processor resources, since it needs to be done for every pixel. Many images contain over one million pixels. This can create severe processing bottlenecks.

Modern microprocessors have special instructions that are intended to reduce or eliminate bottlenecks such as this. Such instructions are generally referred to as “single-instruction/multiple-data” (SIMD) instructions. In microprocessors manufactured by Intel Corporation, such instructions are referred to as multimedia extensions (MMX). In microprocessors manufactured by Digital Corporation, such instructions are referred to as motion video instructions (MVI).

SIMD instructions and operations are very useful in many signal processing operations. Generally, they allow registers to be grouped as an array, so that an operation can be carried out in parallel on each of the registers. For example, the individual registers of one group or array can be added to the

corresponding registers of another group or array using only a single instruction and using parallel arithmetic processing units of a microprocessor (a set of grouped registers is alternatively referred to as a “wide” register or an “MMX” register, having constituent bytes, words, or double words). This is a great advantage in graphics operations, where similar operations must be performed repetitively on all the pixels of an image.

FIG. 2 shows an example of an SIMD operation. FIG. 2 shows a first SIMD array 20. This is a grouping of three separate pixel value registers R1, R2, and R3. Each of these registers contains a single pixel value. A second SIMD register 22 is a grouping of three additional pixel registers R4, R5, and R6, each containing further pixel values. Result registers R7, R8, and R9 are contained in a third SIMD array 24.

In this example, it is desired to calculate R7, R8, and R9 such that  $R7=R1+R4$ ;  $R8=R2+R5$ ; and  $R9=R3+R6$ . Rather than conducting three different addition operations, a single SIMD instruction is used to accomplish this result. When executing such an instruction, a microprocessor performs each of the three discrete operations in parallel, resulting in a significant gain in speed. Popular processors are capable of operating on as many as eight different values in parallel rather than the three illustrated in FIG. 2.

It is not difficult to see how SIMD operations can be used to speed graphics operations such as an averaging operation. Instead of averaging each pixel individually, the averaging process is carried out in parallel for five different pixels, using the five parallel registers of each SIMD array. The first registers of the SIMD arrays are used for the first set of values that are to be averaged, the second registers of the SIMD arrays are used for the second set of values that are to be averaged, and so on. Thus, five pixels can be filtered in little more than the time that would otherwise have been required to filter only a single pixel.

SIMD operations work well for simple algorithms where operations can be conducted in parallel on different sets of values—where defined operations are applied non-conditionally to each set of values. Finding a median value, however, is not as simple. When finding a median value, it is generally necessary to compare different pixel values and to sort them as a result of such a comparison—certain mathematical operations will be applied in one case, but not in another.

This is illustrated in FIG. 3, which shows a comparison of pixels a and b. One action 31 is performed if a is greater than b. Another action 32 (or possibly no action) is performed if a is less than b. FIG. 3 illustrates that two divergent processing branches are required to perform this logic.

SIMD instructions do not provide this type of conditional logic. If one SIMD action or operation is applied to one value in an SIMD array, the same action or operation is necessarily applied to all values in the array. Thus, it has previously not been possible to effectively utilize SIMD instructions when calculating median values.

As a result, calculating median values remains a significant processing bottleneck in spite of the availability of SIMD operations.

### SUMMARY OF THE INVENTION

The invention is a system and method for finding median values in a way that can be optimized by the use of SIMD instructions. In accordance with the invention, a Boolean table or matrix is constructed using a plurality of bit registers. Each bit register has bit positions forming columns that