US 20150147005A1

(19) United States

(12) Patent Application Publication

(10) Pub. No.: US 2015/0147005 A1

(43) Pub. Date: May 28, 2015

Lerner et al.

(54) METHODS AND APPARATUS FOR IMAGE PROCESSING AT PIXEL RATE

(71) Applicant: **ANALOG DEVICES GLOBAL**, Hamilton (BM)

(72) Inventors: **Boris Lerner**, Sharon, MA (US); **Michael Meyer-Pundsack**, Egenburg (DE); **Gopal Gudhdur Karanam**, Bangalore (IN); **Pradip Thaker**, Gujarat (IN)

(73) Assignee: **ANALOG DEVICES GLOBAL**, Hamilton (BM)

(21) Appl. No.: **14/611,735**

(22) Filed: **Feb. 2, 2015**

Related U.S. Application Data

(63) Continuation of application No. 13/892,531, filed on May 13, 2013, now Pat. No. 8,947,446, which is a continuation of application No. 13/892,508, filed on May 13, 2013, now Pat. No. 8,766,992, which is a continuation of application No. 13/354,819, filed on Jan. 20, 2012, now Pat. No. 8,441,492, which is a continuation of application No. 12/619,825, filed on Nov. 17, 2009, now Pat. No. 8,130,229.

Publication Classification

(51) **Int. Cl.**
G06T 1/20 (2006.01)

(52) **U.S. Cl.**
CPC ***G06T 1/20*** (2013.01); *G06T 2200/28* (2013.01)

(57) **ABSTRACT**

Embodiments of the present invention provide for improved timing control in 2-D image processing to maintain a constant rate of fetches and pixel outputs even when the processing operations transition to a new line or frame of pixels. A one-to-one relationship between incoming pixel rate and outgoing pixel rate is maintained without additional clock cycles or memory bandwidth as an improved timing control according to the present invention takes advantage of idle memory bandwidth by pre-fetching a new column of pixel data in a first pixel block of a next line or frame while a new column of an edge pixel block on a current line is duplicated or zeroed out. As the edge pixel block(s) on the current line are processed, the data in the first pixel block of the next line or frame become ready for computation without extra clock cycles or extra memory bandwidth.

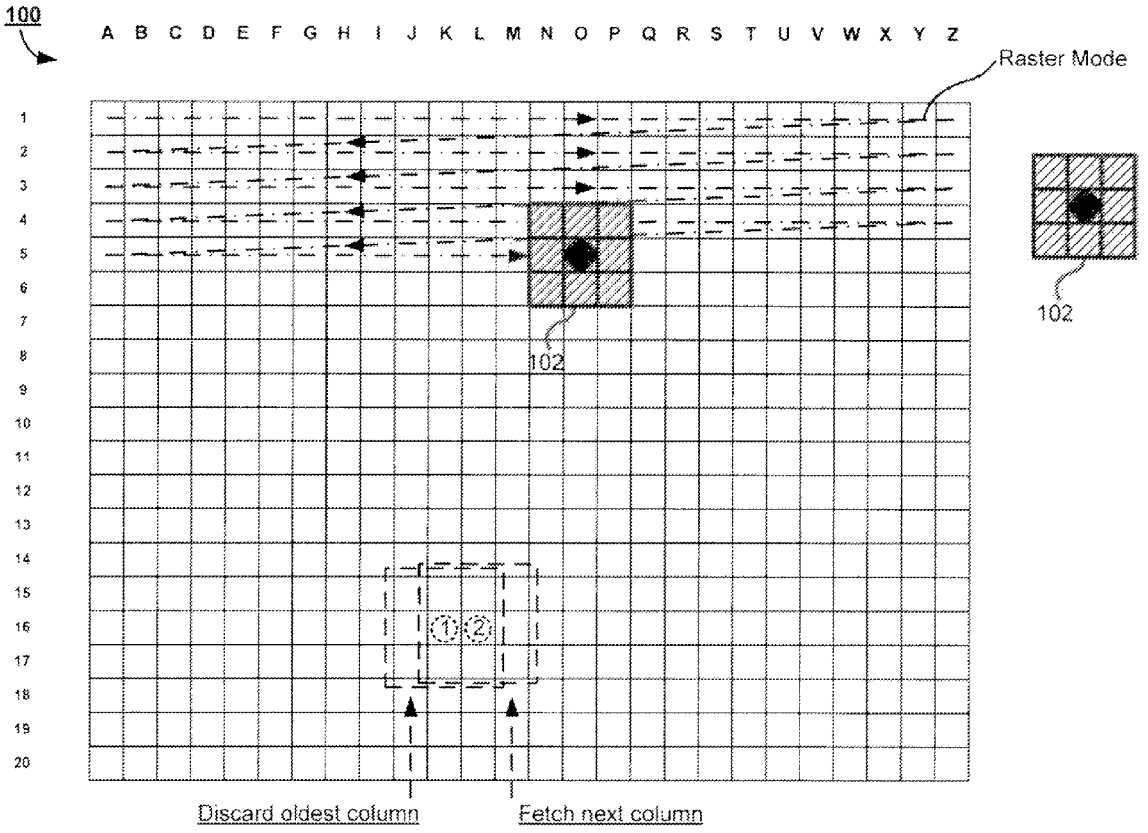


Figure 1

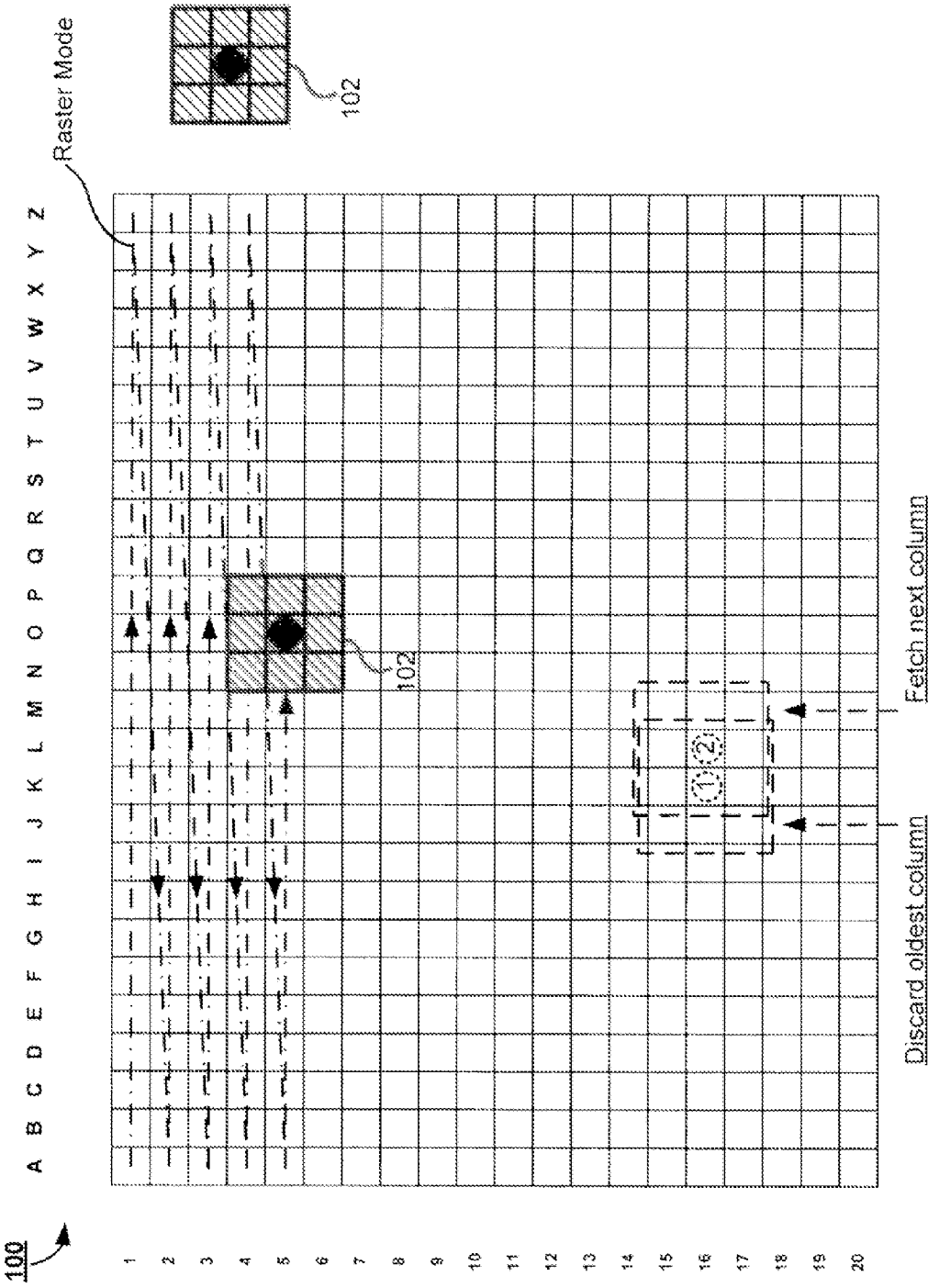

100
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Raster Mode
102
102
1
2
Fetch next column
Discard oldest column

Figure 2

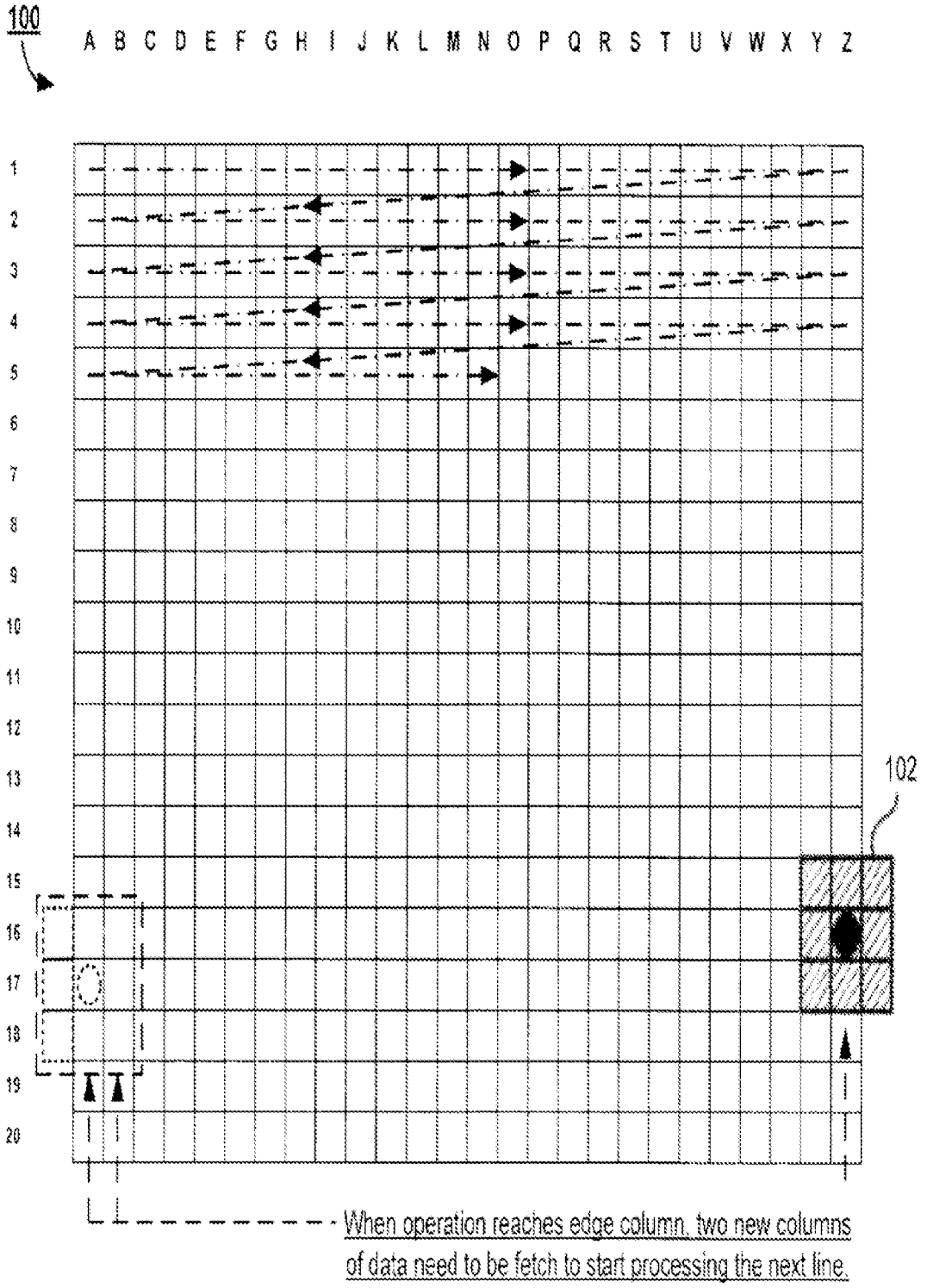

100
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
102
When operation reaches edge column, two new columns of data need to be fetch to start processing the next line.

Figure 3

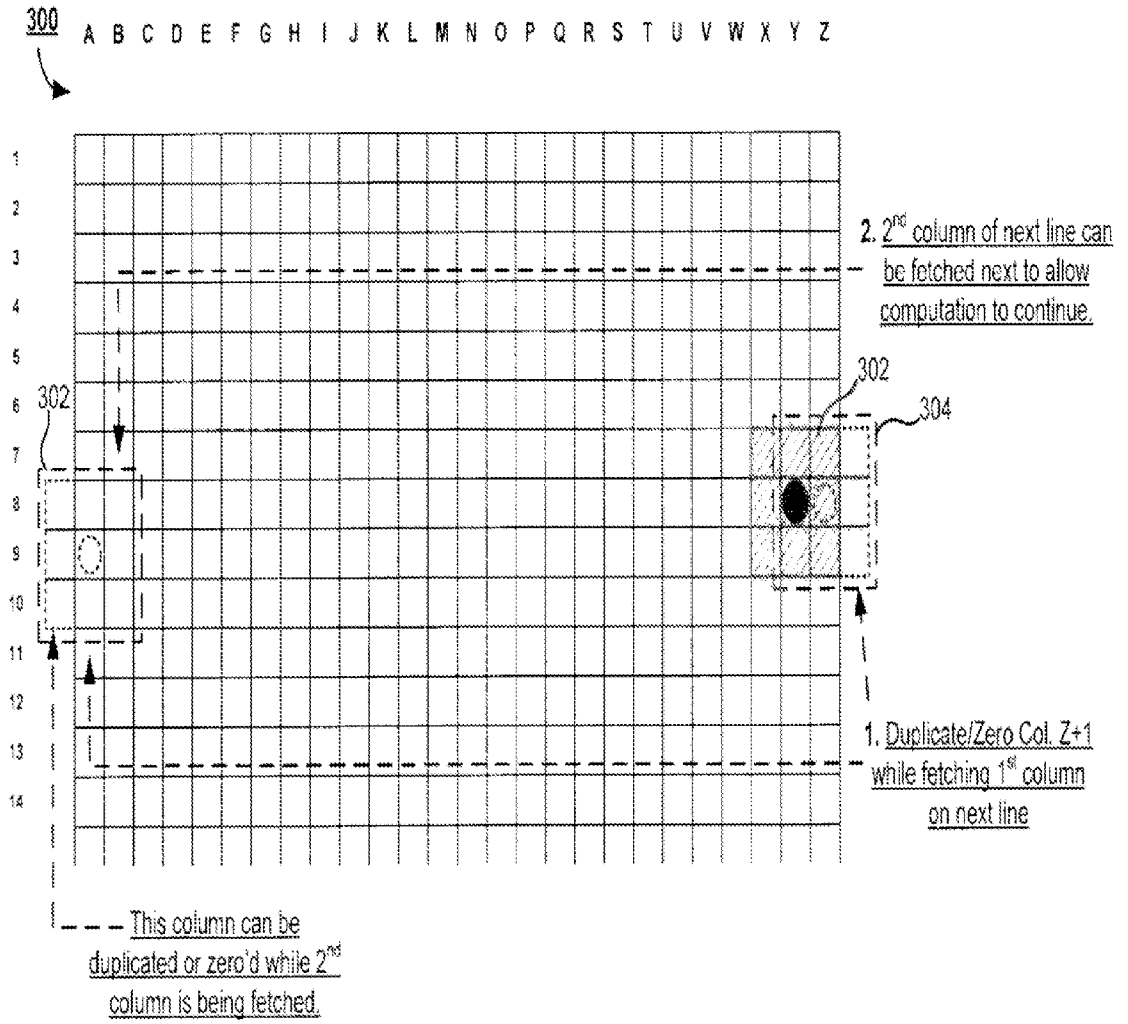

300
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
1
2
3
4
5
6
7
8
9
10
11
12
13
14
302
302
304
2. 2nd column of next line can be fetched next to allow computation to continue.
1. Duplicate/Zero Col. Z+1 while fetching 1st column on next line
This column can be duplicated or zero'd while 2nd column is being fetched.

Figure 4

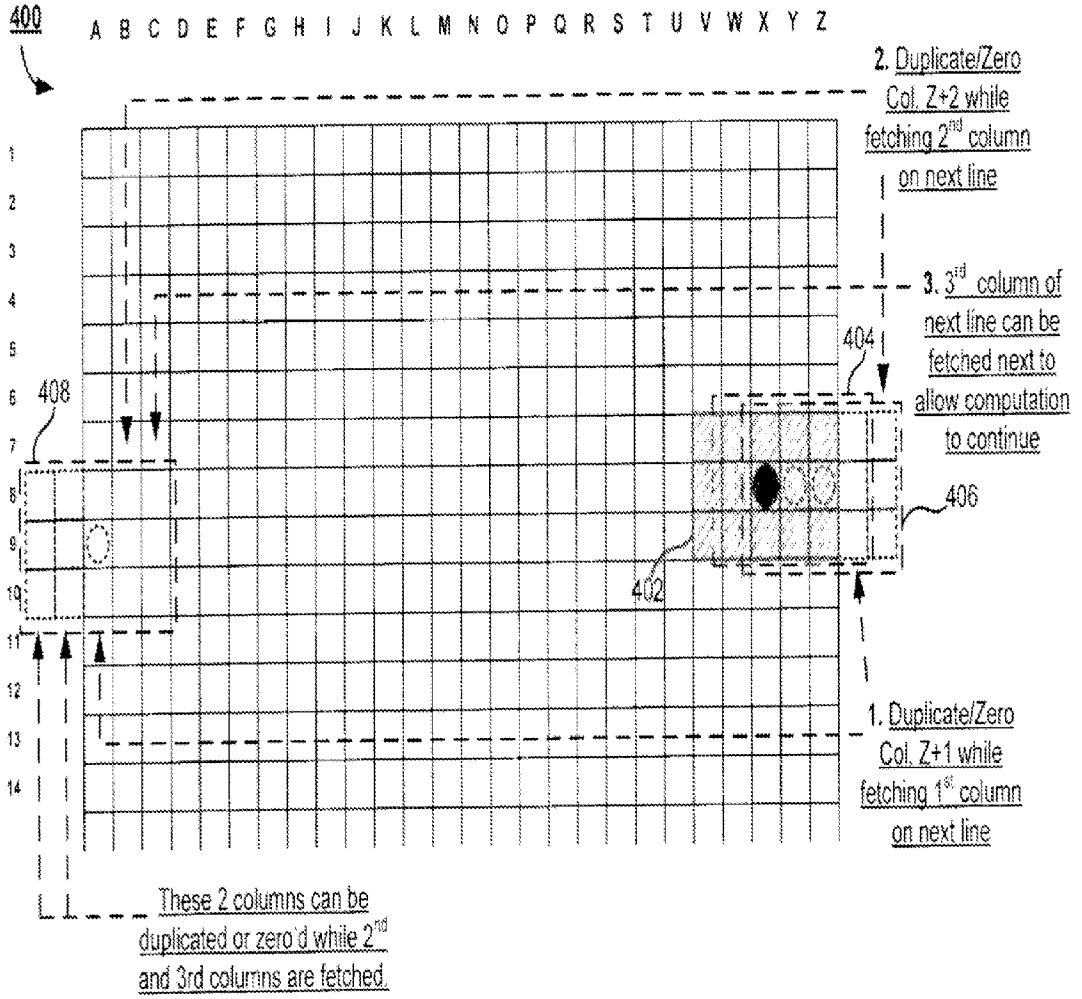

400
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
1
2
3
4
5
6
7
8
9
10
11
12
13
14
2. Duplicate/Zero Col. Z+2 while fetching 2nd column on next line
3. 3rd column of next line can be fetched next to allow computation to continue
404
408
406
402
1. Duplicate/Zero Col. Z+1 while fetching 1st column on next line
These 2 columns can be duplicated or zero'd while 2nd and 3rd columns are fetched.

Figure 5

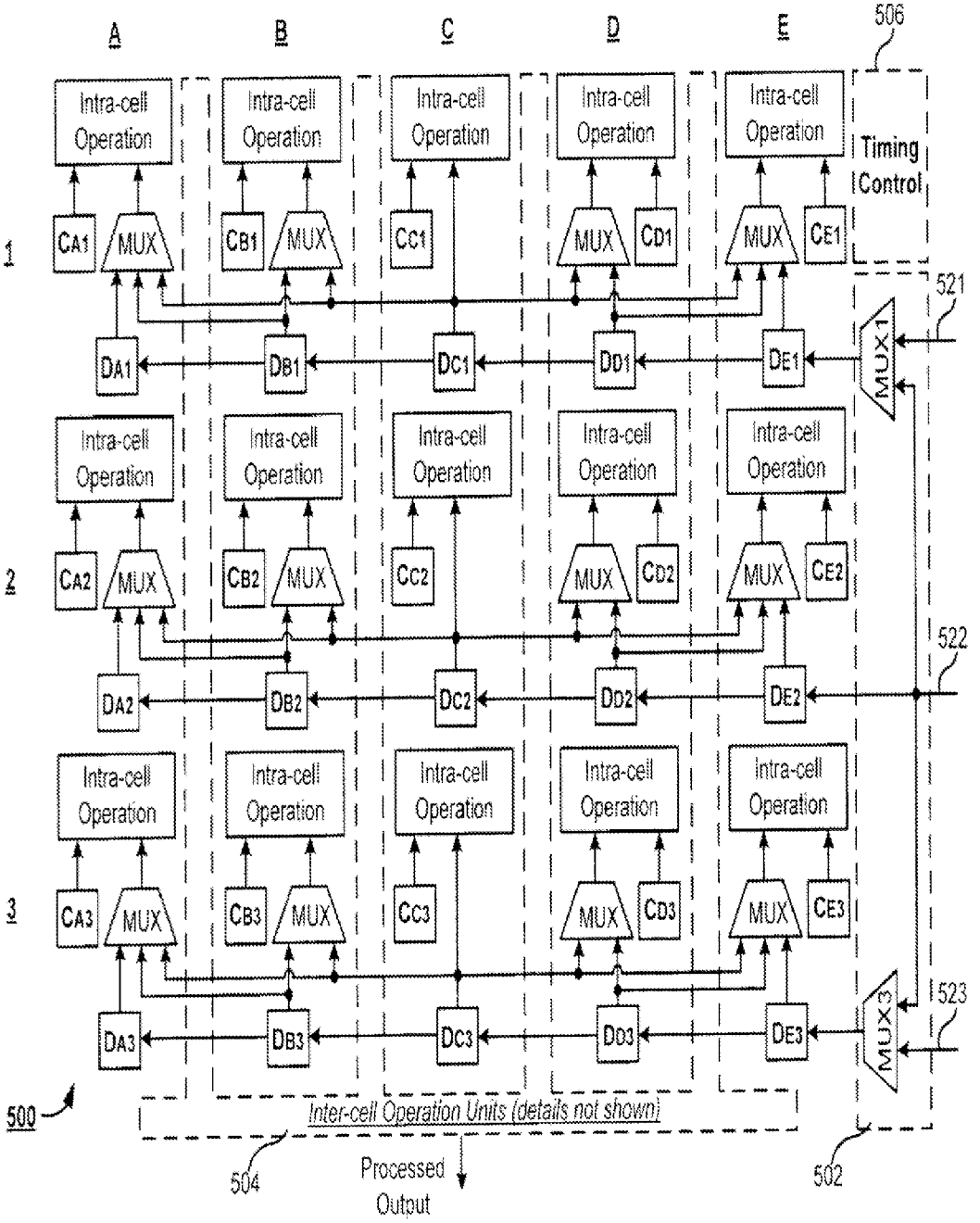
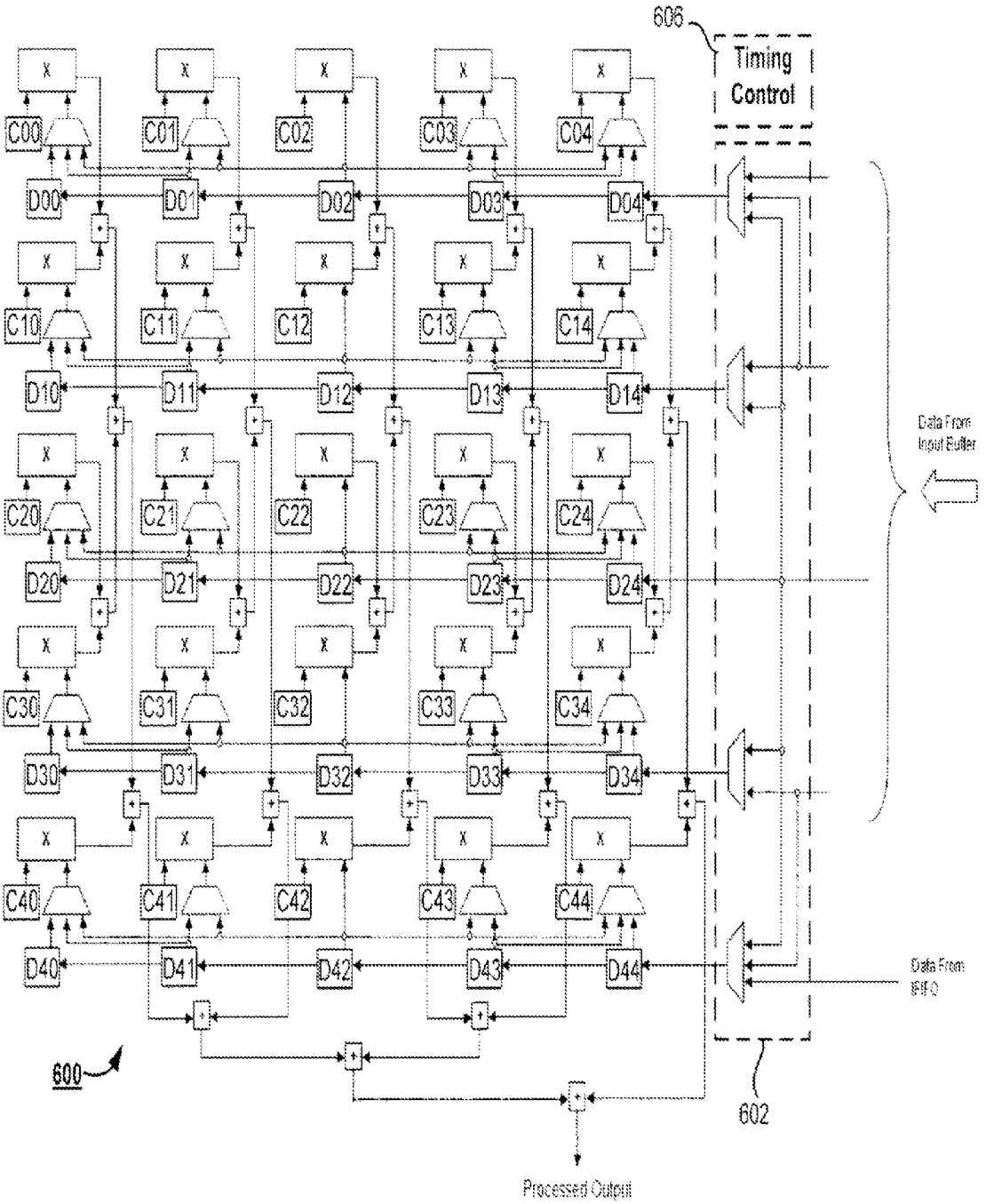

A
B
C
D
E
506
Intra-cell Operation
Timing Control
1
2
3
CA1
CB1
CC1
CD1
CE1
CA2
CB2
CC2
CD2
CE2
CA3
CB3
CC3
CD3
CE3
MUX
DA1
DB1
DC1
DD1
DE1
DA2
DB2
DC2
DD2
DE2
DA3
DB3
DC3
DD3
DE3
521
522
523
MUX1
MUX3
500
Inter-cell Operation Units (details not shown)
504
502
Processed Output

Figure 6


606
Timing Control
C00
C01
C02
C03
C04
D00
D01
D02
D03
D04
C10
C11
C12
C13
C14
D10
D11
D12
D13
D14
C20
C21
C22
C23
C24
D20
D21
D22
D23
D24
C30
C31
C32
C33
C34
D30
D31
D32
D33
D34
C40
C41
C42
C43
C44
D40
D41
D42
D43
D44
Data From Input Buffer
Data From IFIFO
600
602
Processed Output

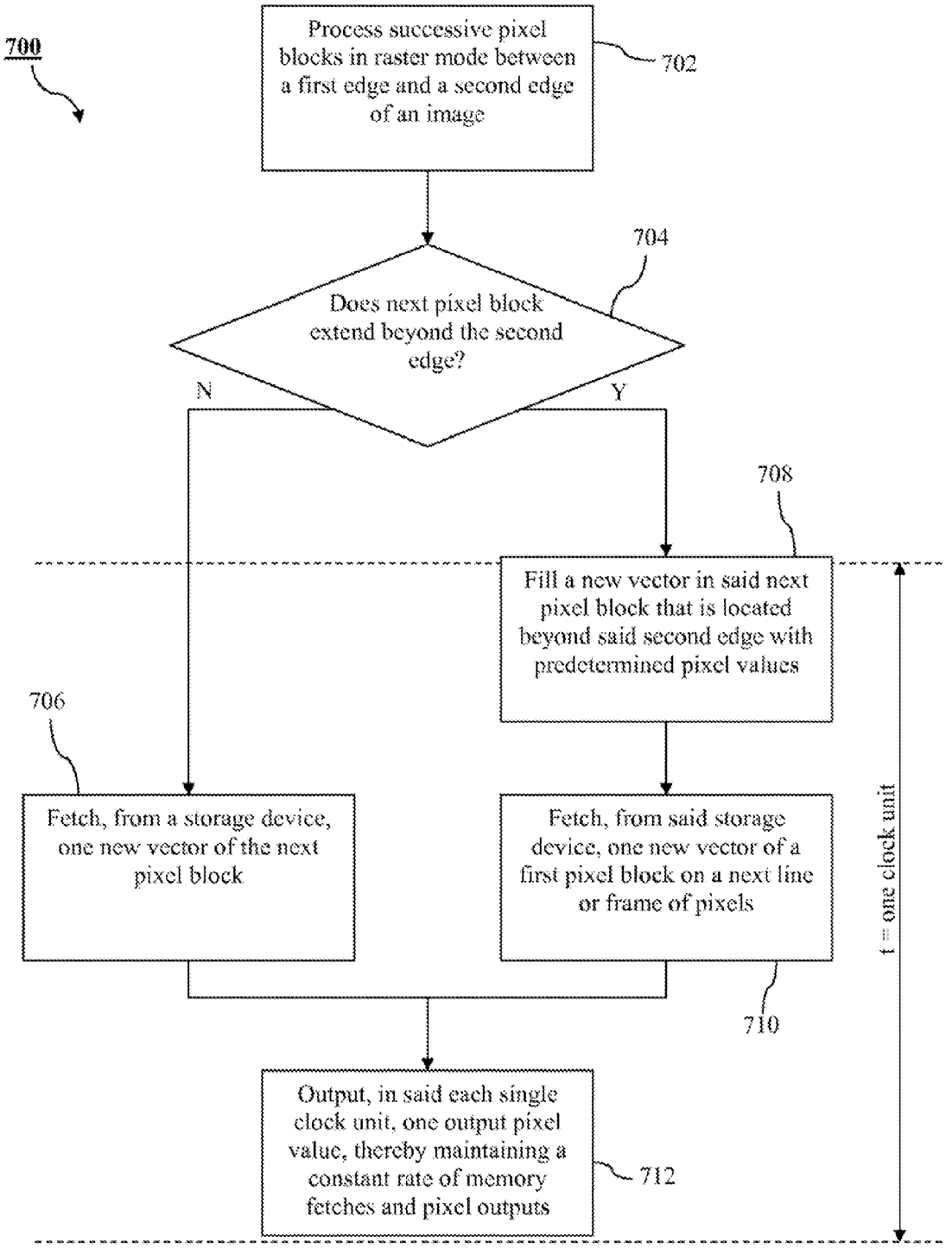

700
Process successive pixel blocks in raster mode between a first edge and a second edge of an image
702
704
Does next pixel block extend beyond the second edge?
N
Y
708
Fill a new vector in said next pixel block that is located beyond said second edge with predetermined pixel values
706
Fetch, from a storage device, one new vector of the next pixel block
Fetch, from said storage device, one new vector of a first pixel block on a next line or frame of pixels
710
Output, in said each single clock unit, one output pixel value, thereby maintaining a constant rate of memory fetches and pixel outputs
712
t = one clock unit

Figure 7

METHODS AND APPARATUS FOR IMAGE PROCESSING AT PIXEL RATE

CROSS-REFERENCE TO RELATED APPLICATION

[0001] This application is a continuation (and claims the benefit of priority under 35 U.S.C. §120) of U.S. patent application Ser. No. 13/892,531, filed May 13, 2013, and entitled "METHODS AND APPARATUS FOR IMAGE PROCESSING AT PIXEL RATE", which is a continuation of U.S. patent application Ser. No. 13/892,508, filed May 13, 2013, and entitled "METHODS AND APPARATUS FOR IMAGE PROCESSING AT PIXEL RATE", being issued on Jul. 1, 2014 as U.S. Pat. No. 8,766,992, which is a continuation of U.S. patent application Ser. No. 13/354,819, filed Jan. 20, 2012, and entitled "METHODS AND APPARATUS FOR IMAGE PROCESSING AT PIXEL RATE" being issued on May 14, 2013 as U.S. Pat. No. 8,441,492, which is a continuation of Ser. No. 12/619,825, filed Nov. 17, 2009, and entitled "METHODS AND APPARATUS FOR IMAGE PROCESSING AT PIXEL RATE," now issued as U.S. Pat. No. 8,130,229. The disclosures of the prior applications are considered part of (and are incorporated herein by reference of this application.

FIELD OF THE INVENTION

[0002] This invention relates generally to digital signal processing and computer graphics, and more particularly, to image processing at pixel rate.

BACKGROUND OF THE INVENTION

[0003] In processing two-dimensional (2-D) images, it is known to apply a small matrix or kernel to successive pixel blocks to generate output pixels. For example, in 2-D convolution operations, an n^xrn rnatrix ("convolution mask") is typically applied to an image following a raster pattern and, for each pixel in the image, the convolution mask is centered on that pixel and convolved with the corresponding n^xm pixels in the image to compute an output pixel value.

[0004] The output pixels so generated then collectively form a new (processed) digital image. Depending on the convolution mask used, 2-D convolution operations can filter out noise, enhance object edges, or achieve other desired effects on a digital image. Similarly, in 2-D correlation operations, a matrix is applied to an image in raster mode and computed with each pixel and its neighboring pixels to generate a corresponding output pixel. These and other kernel-based 2-D processing operations can be implemented in software or hardware and applied to still images or frames in video sequences.

[0005] FIG. **1** shows an exemplary image **100** having a 26×20 array of pixels, each small square representing one pixel. A 3×3 kernel **102** may be applied to the image **100**, starting from the top left corner (i.e., Pixel A1) and passing from edge to edge, line by line in raster mode. For each pixel in the image **100**, the digital values of a corresponding pixel block —that pixel and its eight neighboring pixels—need to be retrieved from a memory device or an input buffer before those pixel data are computed with the values of the kernel matrix. The steps of retrieving the pixel data and computing the pixel data with the kernel are typically pipelined, driven by a clock at pixel rate. As the kernel is advanced to the next pixel, for example, from Pixel K16 (marked "1") to Pixel L16 (marked "2") in the image **100**, only one new column of pixel data (i.e., those of Pixels M15-17) needs to be retrieved. That is, during each clock cycle in the pipelined process, only one new column of pixel data is retrieved from the memory or input buffer, and one output pixel value is usually generated at the same time. Thus, when those pixel blocks located completely within the boundaries of the image **100** ("internal pixel blocks") are processed, there can be a one-to-one relationship between the rate of retrieving pixel data columns and the outgoing pixel rate.

[0006] However, such a fetch-one-column-and-output-one-pixel timing pattern cannot be maintained when the kernel reaches an edge of the image and is about to start scanning and operating on a new line or frame. FIG. **2** illustrates this problem, again with the image **100** and the kernel **102**. The pixel block covered by the kernel **102** in FIG. **2**, which is centered on Pixel Z16, may be referred to as an "edge pixel block." As the kernel **102** moves from the last pixel on line **16** (Pixel ZI6) to the first pixel on line **17** (Pixel A17), two new columns of pixel data (i.e., those of Pixels A16-18 and B16-18) need to be fetched before the output pixel value corresponding to Pixel A17 can be computed. A similar problem exists with kernels of other sizes and when the kernel transitions to a new frame in a video sequence, In order to maintain the outgoing pixel rate during the transition, conventional 2-D image processing approaches would require either extra clock cycles for the extra column(s) to be fetched or a memory bandwidth much larger than what is used for non-edge pixel blocks. Neither of these solutions is desirable for lack of efficiency,

SUMMARY OF THE INVENTION

[0007] Embodiments of the present invention provide for improved timing control in 2-D image processing to maintain a constant rate of memory fetches and pixel outputs even when the processing operations transition to a new line or frame of pixels.

[0008] In one aspect of the invention, a method for processing one or more images having a plurality of pixels includes processing successive pixel blocks in raster mode between a first edge and a second edge of an image. During each single clock unit when a next pixel block to be processed does not extend beyond said second edge, one new vector (i.e., a row or a column) of the next pixel block is fetched from a storage device. During each single clock unit when said next pixel block to be processed extends beyond said second edge, a new vector in said next pixel block that is located beyond said second edge is filled with predetermined pixel values while one new vector of a first pixel block on a next line or frame of pixels is fetched from said storage device. In said each single clock unit, one output pixel value is generated, thereby maintaining a constant rate of memory fetches and pixel outputs.

[0009] In another aspect of the invention, an apparatus for processing one or more images having a plurality of pixels includes: (i) means for processing successive pixel blocks in raster mode between a first edge and a second edge of an image; (ii) means for, during each single clock unit when a next pixel block to be processed does not extend beyond said second edge, fetching one new vector of the next pixel block from a storage device; (iii) means for, during each single clock unit when said next pixel block to be processed extends beyond said second edge, filling a new vector in said next pixel block that is located beyond said second edge with predetermined pixel values and simultaneously fetching,

from said storage device, one new vector of a first pixel block on a next line or frame of pixels; and (iv) means for outputting, in said each single clock unit, one output pixel value, thereby maintaining a constant rate of memory fetches and pixel outputs.

[0010] In still another aspect, an apparatus for processing successive pixel blocks in raster mode between a first edge and a second edge of an image may comprise: an input data interface coupled with one or more storage devices to receive at least one input stream of pixel values of said image; an array of pixel processing cells, each cell including a coefficient storage unit that stores a coefficient value of a kernel to be applied to said image during a two-dimensional image processing, a pixel storage unit that stores a pixel value of said image, said pixel storage unit being coupled either to another pixel storage unit of an adjacent cell or to said input data interface, thereby allowing said pixel value to be initially received via said input data interface and subsequently propagated from one vector to an adjacent vector across said array, a multiplexer unit, if said each cell is not in a center vector of said array, having inputs coupled to said pixel storage unit, a center-vector pixel storage unit on the same line of said array, and any pixel storage unit located between said center vector and said each cell, and one or more intra-cell operation units; inter-cell operation units that process outputs from said array of pixel processing cells to generate an output pixel value; and a timing control module that coordinates operations of said input data interface, said army of pixel processing cells, and said inter-cell operation units. During each single clock unit when a next pixel block to be processed does not extend beyond said second edge, said timing control module causes one new vector of said next pixel block to be fetched from said one or more storage devices. During each single clock unit when said next pixel block to be processed extends beyond said second edge, said timing control module causes a new vector in said next pixel block that is located beyond said second edge to be filled with predetermined pixel values while causing one new vector of a first pixel block on a next line or frame of pixels to be fetched from said one or more storage devices, thereby maintaining a constant rate of memory fetches and pixel outputs.

[0011] The present invention will now be described in more detail with reference to exemplary embodiments thereof as shown in the accompanying drawings. While the present invention is described below with reference to exemplary embodiments, it should be understood that the present invention is not limited thereto. Those of ordinary skill in the art having access to the teachings herein will recognize additional implementations, modifications, and embodiments, as well as other fields of use, which are within the scope of the present invention as described herein, and with respect to which the present invention may be of significant utility.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] Other objects, features and advantages will occur to those skilled in the art from the following description of a preferred embodiment and the accompanying drawings, in which:

[0013] FIG. **1** shows an exemplary image with an array of pixels;

[0014] FIG. **2** illustrates a disruption of constant pixel rate encountered in conventional 2-D image processing approaches;

[0015] FIG. **3** illustrates an exemplary method of 2-D image processing with a 3×3 kernel in accordance with an embodiment of the present invention;

[0016] FIG. **4** illustrates an exemplary method of 2-D image processing with a 5×3 kernel in accordance with an embodiment of the present invention;

[0017] FIG. **5** shows an exemplary 2-D image processing engine in accordance with an embodiment of the present invention

[0018] FIG. **6** shows an exemplary 2-D convolution engine for image processing in accordance with an embodiment of the present invention; and

[0019] FIG. **7** illustrates one embodiment of a method for processing an image at pixel rate in accordance with an embodiment of the invention.

DETAILED DESCRIPTION OF THE INVENTION

[0020] Embodiments of the present invention improve image processing and can maintain a one-to-one relationship between incoming pixel rate and outgoing pixel rate without additional clock cycles or memory bandwidth even when the processing operations transition to a new line or a new frame of pixels. An improved timing control according to the present invention takes advantage of idle memory bandwidth by pre-fetching a new column of pixel data in a first pixel block of a next line or frame while a new column of an edge pixel block on a current line is duplicated or zeroed out. As the edge pixel block(s) on the current line arc processed, the data in the first pixel block of the next line or frame become ready for computation without extra clock cycles or extra memory bandwidth. More details, features, and advantages of the present invention can be appreciated with reference to the accompanying drawings and the detailed explanation below.

[0021] Referring to FIG. **3**, there is illustrated an exemplary method of 2-D image processing with a 3×3 kernel **302** in accordance with an embodiment of the present invention. FIG. **3** shows part of a digital image **300** over which the kernel **302** is scanned in raster mode.

[0022] As used herein, the tem1 "raster mode" refers to a line-by-line or line-after-line pattern of' scanning a kernel over an image during 2-D image processing operations such as convolution or correlation. Typically, the kernel is advanced one pixel at a time and from one edge of the image to an opposite edge, thereby covering successive pixel blocks where each new block includes only one new column of pixels not in its adjacent block. However, the raster mode does not necessarily require the scanning to cover every single line of the image or consistently moving from one line to its next adjacent line. For example, the raster pattern may cover odd lines of an image and then cover even lines. Nor does the raster mode necessarily require a specific scanning direction on each line of the image. Furthermore, those skilled in the art will appreciate that the term "column" and "line" have relative significance when one is used in the context of the other. Therefore, a 2-D image processing implementation where a kernel is scanned column-wise and new lines of pixel data arc pre-fetched also falls within the scope of the present invention herein disclosed.

[0023] In FIG. **3**, the kernel **302** is scanned over the image **300** one line after another in raster mode. As the kernel **302** is advanced on Line **8** to the right edge of the image **300**, the last complete pixel block covered by the kernel **302** is centered around Pixel Y8, and the last column of new data fetched on that line correspond to Pixels Z 7-9. The values of Pixels Z7-9

may be fetched from a storage device (e.g., a memory, input buffer, or first-in-first-out (FIFO) device) during one clock unit. As used herein, the term "clock unit" refers to a unit of time, such as one full clock cycle or one half of a clock cycle, that provides timing reference to or synchronizes logic gates or arithmetic devices during pipelined image processing operations.

[0024] As the kernel is advanced to the last pixel on Line **8** (i.e., Pixel Z8), the next pixel block to be processed, block **304**, extends beyond the right edge of the image **300**. A new column of pixel data (on Col. Z+I and in the pixel block **304**) that would otherwise need to be fetched from the storage device will instead receive values duplicated from pixels in a corresponding portion of the right edge of the image **300**. That is, in one embodiment the new column (Col. Z+I) will be a copy of the values of the last column (Pixels Z7.-9) which have already been fetched in the previous clock unit. in another embodiment, the image processor may be configured to fill the new column with zeros or other numeric values, That is, in general, the "new" column (Col. Z+1) is to be filled with pixel values that do not need to be fetched from the storage device. While Col. Z+1 is being filled with predetermined values (e.g., either duplicated values or zeros), memory bandwidth is available for fresh data to be fetched from the storage device. Accordingly, embodiments of the present invention take advantage of this idle memory bandwidth and may cause the first column of pixel data on the next line or frame of the image to be pre-fetched. In this particular example shown in FIG. **3**, during the single clock unit while Col. Z+1 on Line **8** is being filled with pixel values duplicated from Col. Z or simply filled with zeros, the pixel values of the first column on Line **9**, Pixels A8-10, may be pre-fetched from the storage device.

[0025] Then, in the next dock unit, the second column of pixel data on Line **9** may be pre-fetched from the storage device. The portion of the first pixel block on Line **9** that extends beyond the left edge of the image **300** may be filled with either pixel values of the first column (Pixels A8-10, already fetched in the previous clock unit) or zeros. As a result of the timing arrangement heretofore described, the image processor has been able to fetch one and only one new column of pixel data during each single clock unit, even when the 2-D image processing operations transition from Line **8** to Line **9** of the image **300**. Such memory fetches perfom1ed at the constant input pixel rate, pipelined with the other computation steps, can therefore ensure a constant output rate of one output pixel value per clock unit.

[0026] FIG. **4** illustrates an exemplary method of 2-D image processing with a 5×3 kernel **402** in accordance with an embodiment of the present invention. FIG. **4** shows part of a digital image **400** over which the kernel **402** is scanned in raster mode.

[0027] When the center of the kernel **402** advances to Pixel X8, in clock unit T, the next pixel block to be processed, block **404**, will be centered on PixelY8 and extend beyond the right edge of the image **400**. During the single clock unit T while Col. Z+1 is being filled with zeros or values duplicated from Pixels Z7-9, the first column on the next line (Line **9**), Pixels A8-I 0, may be pre-fetched from a storage device.

[0028] In the next clock unit T+1, the next pixel block to be processed, **406**, will be centered on Pixel Z8 and extend even further beyond the right edge of the image **400**. During the single clock unit T+I while Col. Z+2 is being filled with zeros or values duplicated from Pixels Z7-9, the second column on Line **9** may be pre-fetched from the storage device.

[0029] In the next clock unit T+2, the next pixel block to be processed, **408**, will be centered on Pixel A9. During the single clock unit T+2, the third column on Line **9** may be fetched from the storage device. Columns A-1 and A 2 on Line **9** may be filled with predetermined pixel values such as zeros or values of Pixels A8-10.

[0030] At the conclusion of the clock unit T+2, all elements of the first pixel block (**408**) on Line **9** are ready for computation with the kernel **402**.

[0031] Embodiments of the present invention may be implemented in software, firmware, and/or hardware. Most preferably, embodiments of the present invention are implemented in a 2-D convolution or correlation compute engine as part of a digital image processing system or graphics acceleration apparatus.

[0032] FIG. **5** shows an exemplary 2-D image processing engine **500** in accordance with an embodiment of the present invention.

[0033] The 2-D image processing engine **500** includes an array of pixel processing cells (in this embodiment, a 5×3 array), an input data interface **502**, inter-cell operation units **504**, and a timing control module **506**.

[0034] Each of the pixel processing cells in the array includes a coefficient storage unit, such as a data register (e.g., C_A1 and C_{B2}), that stores a coefficient value of a kernel to be applied to a pixel block during a 2-D image processing (e.g., convolution or correlation operations). Each pixel processing ceil also includes a corresponding pixel storage unit, such as a data register (e.g., D_{A1} and D_{B2} that stores a pixel value of the pixel block. Each pixel storage unit is coupled with pixel storage units of adjacent pixel processing cells on the same line. If the pixel processing cell is in the edge column on the input side (here, Col, E), the pixel storage unit is also coupled to the data input interface **502** to receive fresh pixel data. As a result, a new column of pixel values may be obtained via the data input interface **502** and temporarily stored in the pixel storage units in the input side edge column, and the previously stored pixel values in each column of pixel storage units can be "pushed" or duplicated to the next column on the left. This arrangement forms a data path whereby a column of pixel values can be propagated from the data input interface **502** to the Jett, hopping one column per clock unit. By the end of each clock unit, the pixel values previously stored in the pixel storage units of the leftmost column, here Col. A, are overwritten and therefore discarded.

[0035] Each pixel processing cell, if it is not in the center column, here Col. C, also includes a multiplexer unit ("MUX"), The multiplexer unit has one input coupled to the pixel storage unit in the same cell. In addition, the multiplexer unit has inputs coupled to a center-column pixel storage unit on the same line as well as any pixel storage unit that is located between the center column and the current cell. As a result, data paths are created for each column not located on the left or right edge of the array ("non-edge column") to duplicate its content outwardly to other column(s) to the extent the multiplexer units introduce additional delays in the non-center-column cells, delay elements may be included in the center-column cells to balance out the delays among the cells, as can be appreciated by those skilled in the art.

[0036] Each pixel processing cell further includes one or more intra-cell operation units (e.g., arithmetic or logic devices such as a multiplier) that receive an input from the

corresponding coefficient storage unit in the same cell. The intra-cell operation unit(s) also receive an input from the corresponding pixel storage unit directly (here the center column, Col. C) or from the output of the multiplexer unit (as in non-center-columns). Results of the intra-cell operations may be further processed with inter-cell operation units **504** (details not shown) to generate a processed output such as an output pixel value.

[0037] The input data interface **502** of the 2-D image processing engine **500** couples the array with one or more storage devices (not shown), such as an input buffer and/or an input first- in-first-out (IFIFO) device, that provide fresh pixel values for the three rows of pixel processing cells. More specifically, the fresh pixel values arc inputted to the pixel storage units in the edge column on the input side (here Col. E) via input lines **521**, **522**, and **523** before being propagated to the rest of the columns. Two multiplexers, MUX1 and MUX3, are provided to allow the center row (here Row 2) pixel values to be duplicated to the non-center-rows, for example, when processing a top or bottom line of an image. Typically, the input image pixel data are all temporarily stored in an input buffer, and therefore all three input lines **521**, **522**, and **523** are coupled to the input buffer. According to one embodiment of the present invention, one of the input lines (e.g., line **523** which feeds the bottom row input) may be coupled directly to an IFIFO receiving a live feed of one row' of the pixel values, therefore saving one third of the buffer space and memory bandwidth that would othe1wisc be required.

[0038] The timing control module **506** coordinates a pipelined process of computing successive pixel blocks of an image against the kernel whose coefficient values are stored in the coefficient storage units. In each clock unit, the timing control module **506** causes one column of fresh pixel values to be fetched into pixel storage units D_{E1}, D_{E2}, and D_{E3} (here in Col. E) via the data input interface **502**. After pixel values of an edge column of the image on a current line are fetched into the edge column pixel storage units (here Col. E), in the next clock unit (T+1), the timing control module **506** may cause a first column on a next line of the image (or a new line of a next frame) to be pre-fetched into an edge column (here Col. E) via the data input interface **502**. The pixel values of this edge column are now duplicated lo the pixel storage units in the left adjacent column (here Col. D). Then, in the next clock unit (T+1), such pixel values of the edge column can be selected by the multiplexer units in the edge column (here Col. E) for intra-cell operations. During clock unit (T+1), the timing control module **506** may cause a second column on the next line of the image to be pre-fetched into the edge column (here Col. E) via the data input interface **502** while pushing pixel values of the first column on the next line into the left adjacent column (here Col. D) and pushing the pixel values of the edge column on the current line into the next left adjacent column (here Col. C). In the next clock unit (T+2), the pixel values of the edge column can again be selected by the multiplexer units in Col. D and Col. E for intra-cell operations. During clock unit (T+2), the timing control module **506** may cause a third column on the next line of the image to be pre-fetched into the edge column (here Col. E) while pushing pixel values of the first and second columns on the next line into the two left adjacent columns (here Col. C and Col. D respectively). The next clock unit (T+3) will see the 2-D image processing engine **500** ready to proceed with computation on the first pixel on the next line of the image.

[0039] The architecture of the 2-D image processing engine **500** shown in FIG. **5** is scalable in both dimensions to accommodate arbitrary kernel sizes. FIG. **6** shows an exemplary 2-D convolution engine **600** with a 5×5 array of pixel processing cells for image processing in accordance with an embodiment of the present invention. For 2-D convolution operations, the intra-cell operation unit in each pixel processing cell is a multiplier, and the inter- cell operation units include adders cascaded to sum up the output products of all the multipliers to finally generate a processed output With more rows than the 2-D image processing engine **500**, the 2-D convolution engine **600** includes additional multiplexers in its input data interface **602** to route center-row pixel values to the non-center rows. A timing control module **606** coordinates the operations of the various elements in a pipelined process wherein the input pixel rate and output pixel rate maintain a one-to-one relationship in accordance with the above-described memory fetch and data routing techniques.

[0040] From the examples illustrated in FIGS. **3**-**6**, it can be appreciated that embodiments of the present invention can be implemented in 2-D image process mg operations applying any $n^x m$ kernel matrix to successive pixel blocks where n and m are typically both odd integers. According to some embodiments, the methods and apparatus disclosed herein may be adapted to accommodate a $n^x m$ kernel matrix where n and/or m are even integers. In addition, such 2-D image processing is not limited to convolution operations and can be applied to 2-D correlation or other operations.

[0041] FIG. **7** illustrates one embodiment of a method **700** for processing an image at pixel rate in accordance with an embodiment of the invention. Successive pixel blocks are processed in raster mode between a first edge and a second edge of an image (step **702**). If a next pixel block to be processed does not extend beyond the second edge (step **704**), one new vector of the next pixel block is fetched from a storage device (step **706**). If, on the other hand, the next pixel block to be processed extends beyond the second edge (step **704**), a new vector is filled in the next pixel block (that is located beyond said second edge) with predetermined pixel values (step **708**), and one new vector of a first pixel block on a next line or frame of pixels is fetched from the storage device (step **710**). One output pixel value is output, in each single clock unit, thereby maintaining a constant rate of memory fetches and pixel outputs (step **712**).

[0042] While the foregoing description includes many details and specificities, it is to be understood that these have been included for purposes of explanation only, and are not to be interpreted as limitations of the present invention. It will be apparent to those skilled in the art that other modifications to the embodiments described above can be made without departing within the scope of the invention as intended to be encompassed by the following claims and their legal equivalents.

What is claimed:

1. A method used in scanning an image in raster mode, the method comprising:

filling a vector in a pixel block with predetermined values, in response to a determination that the vector of the pixel block is located beyond an edge of the image;

performing an image processing on the pixel block with the predetermined values; and

fetching a vector of a next pixel block of the image, in response to the determination that the vector of the pixel block is located beyond the edge of the image.

2. The method of claim 1, wherein the pixel block is centered around a pixel on a line, and the next pixel block is centered around a pixel on an adjacent line to the line on an opposing edge of the edge of the image.

3. The method of claim 1, wherein the pixel block is centered around a pixel on a line, and the next pixel block is centered around a pixel two lines away from the line.

4. The method of claim 1, wherein the predetermined values are duplicated from other values in the pixel block.

5. The method of claim 1, wherein the predetermined values are zeroes.

6. The method of claim 1, wherein the image processing includes a convolution operation applied to the pixel block or a correlation operation applied to the pixel block.

7. The method of claim 1, wherein the fetching is performed in a same clock unit as the filling is performed, thereby maintaining a constant rate of pixel outputs.

8. A non-transitory, computer-readable medium encoded with instructions that, when executed by a processor, cause the processor to perform a method that scans an image in raster mode, the method comprising:

filling a vector in a pixel block with predetermined values, in response to a determination that the vector of the pixel block is located beyond an edge of the image;

performing an image processing on the pixel block with the predetermined values; and

fetching a vector of a next pixel block of the image, in response to the determination that the vector of the pixel block is located beyond the edge of the image.

9. The medium of claim 8, wherein the pixel block is centered around a pixel on a line, and the next pixel block is centered around a pixel on an adjacent line to the line on an opposing edge of the edge of the image.

10. The medium of claim 8, wherein the pixel block is centered around a pixel on a line, and the next pixel block is centered around a pixel two lines away from the line.

11. The medium of claim 8, wherein the predetermined values are duplicated from other values in the pixel block.

12. The medium of claim 8, wherein the predetermined values are zeroes.

13. The medium of claim 8, wherein the image processing includes a convolution operation applied to the pixel block or a correlation operation applied to the pixel block.

14. The medium of claim 8, wherein the fetching is performed in a same clock unit as the filling is performed, thereby maintaining a constant rate of pixel outputs.

15. An apparatus that scans an image in raster mode, the apparatus comprising:

a processor configured to fill a vector in a pixel block with predetermined values, in response to a determination that the vector of the pixel block is located beyond an edge of the image, wherein the processor performs an image processing on the pixel block with the predetermined value, and

the processor is further configured to fetch a vector of a next pixel block of the image, in response to the determination that the vector of the pixel block is located beyond the edge of the image.

16. The apparatus of claim 15, wherein the pixel block is centered around a pixel on a line, and the next pixel block is centered around a pixel on an adjacent line to the line on an opposing edge of the edge of the image.

17. The apparatus of claim 15, wherein the pixel block is centered around a pixel on a line, and the next pixel block is centered around a pixel two lines away from the line.

18. The apparatus of claim 15, wherein the predetermined values are duplicated from other values in the pixel block.

19. The apparatus of claim 15, wherein the image processing includes a convolution operation applied to the pixel block or a correlation operation applied to the pixel block.

20. The apparatus of claim 15, wherein the processor is configured to fetch the vector of the next pixel block in a same clock unit as the processor fills the vector in the pixel block, thereby maintaining a constant rate of pixel outputs.

* * * * *