



(51) International Patent Classification:

G06F 17/21 (2006.01) H03M 7/30 (2006.01)

(21) International Application Number:

PCT/US2017/062463

(22) International Filing Date:

20 November 2017 (20.11.2017)

(25) Filing Language:

English

(26) Publication Language:

English

(71) Applicant: **HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P.** [US/US]; 10300 Energy Drive, Spring, Texas 77389 (US).

(72) Inventors: **CRAMPTON, Bryan**; 1070 NE Circle Blvd., Corvallis, Oregon 97330-4239 (US). **GILG, Thomas J.**; 1070 NE Circle Blvd., Corvallis, Oregon 97330-4239 (US).

BALAKRISHNAN, Mahesh; 1070 NE Circle Blvd., Corvallis, Oregon 97330-4239 (US). **SAHU, Balam**; 1070 NE Circle Blvd., Corvallis, Oregon 97330-4239 (US). **GOPINATHAN, Prasanth**; 1070 NE Circle Blvd., Corvallis, Oregon 97330-4239 (US).

(74) Agent: **PERRY, Garry A.** et al.; HP Inc., Mail Stop 35 3390 E. Harmony Road, Fort Collins, Colorado 80528 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA,

(54) Title: BLENDING PIXEL CELLS

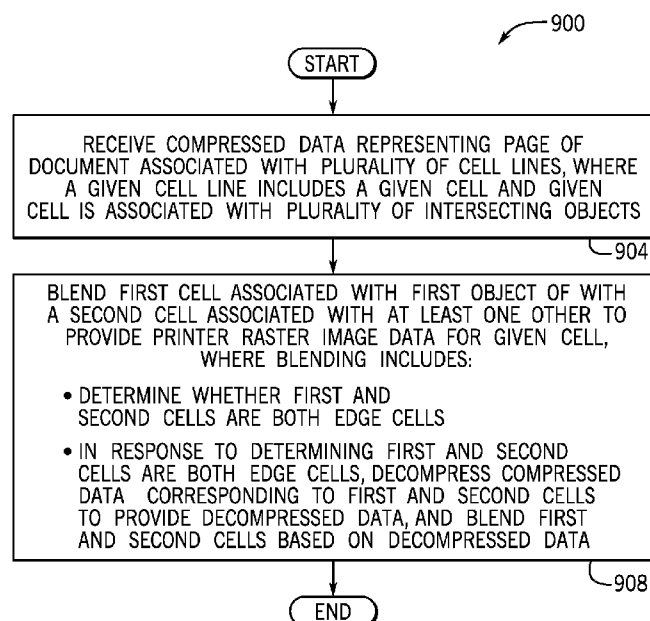


FIG. 9

(57) Abstract: A method includes receiving compressed data representing a page of a document that is associated with a plurality of cell lines. A given cell line of the plurality of cell lines includes a given cell, and the given cell is associated with a plurality of intersecting objects. The technique includes blending a first cell associated with a first object of the plurality of objects with a second cell that is associated with at least one other object of the plurality of objects to provide printer raster image data for the given cell. The blending includes determining whether the first and second cells are both edge cells; and in response to determining that the first and second cells are both edge cells, decompressing the compressed data corresponding to the first and second cells to provide decompressed data, and blending the first and second cells based on the decompressed data.

SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN,
TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

- *as to the identity of the inventor (Rule 4.17(i))*
- *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

Published:

- *with international search report (Art. 21(3))*

BLENDING PIXEL CELLS

Background

[0001] For purposes of printing a document on a digital printing press, a document description file (a portable document file (PDF), for example) may be processed to generate raster image data for the press. The raster image data represents raster images for pages of the document. A raster image is a bit map, which defines a grid of pixels or pixel cells of a document page and defines colors or continuous tones for the pixels/pixel cells.

Brief Description Of The Drawings

[0001] Fig. 1 is a schematic diagram of a system that includes a digital printing press according to an example implementation.

[0002] Fig. 2A is an illustration of a cell line table generated and used by a recomposition engine of the system of Fig. 1 according to an example implementation.

[0003] Fig. 2B depicts an example target cell line according to an example implementation.

[0004] Figs. 3 and 6 are illustrations of cell blending according to example implementations.

[0005] Fig. 4 is an illustration of an object intersection table generated and used by the recomposition engine of Fig. 1 according to an example implementation.

[0006] Fig. 5 is a flow diagram depicting a technique to generate raster image data according to an example implementation.

[0007] Fig. 7A is an illustration of the blending of an edge cell and a solid color cell according to an example implementation.

[0008] Fig. 7B is an illustration of the blending of edge cells according to an example implementation.

[0009] Fig. 7C is an illustration of the reduction of the number of colors of the blended cell of Fig. 7C to produce a blended cell having a reduced number of colors according to an example implementation.

[0010] Fig. 7D is an illustration of the blending of edge cells to produce a single color cell according to an example implementation.

[0011] Fig. 8 is an illustration of source cell blending according to an example implementation.

[0012] Fig. 9 is a flow diagram depicting a technique to blend pixel cells according to an example implementation.

[0013] Fig. 10 is an illustration of machine executable instructions to blend pixel cells according to example implementations.

[0014] Fig. 11 is a schematic diagram of an apparatus to blend pixel cells according to an example implementation.

Detailed Description

[0015] Variable Data Printing (VDP) printing refers to a form of digital printing in which some variable objects, such as texts, graphics and images, may change from one printing of a document to the next; and other, reoccurring, or static, objects of the document do not change. As examples, VDP printing may be used for purposes of printing brochures, advertisements, announcements, and so forth, with information (mailing addresses, for example) that changes among the copies that are produced by the printing press. VDP may present challenges for print shops and their content creators due the changing content.

[0016] In general, to print a VDP document, a print server may analyze a document file that describes the VDP document, such as a portable document format (PDF) file to analyze the document file to identify the static and variable objects of the document. Once identified, the static objects may be processed to derive the corresponding raster image data, and the raster image data for the static objects may then be reused until no longer needed. Through the reuse of the raster image

data for static image objects, computing intensive operations that may otherwise be used to produce raster images for the reoccurring static may be reduced.

[0017] The printer server may include a raster image processor that generates the raster image data for the pages of a document to be printed. One way for the raster image processor to generate the raster image data for a given document page is for the processor to allocate a region of memory for the entire page and use the memory region as a canvas. For this technique, for each object of the document page, the raster image data may write data to the region of memory to effectively form the object on the canvas and blend the object with any other objects that partially or wholly share the same space on the canvas.

[0018] Processing a given document page to generate raster data may, however, may be relatively inefficient, however, as such operations may consume a significant amount of memory and may be associated with relatively intensive computing operations. In accordance with example implementations that are described herein, a raster image processor, or recomposition engine, generates raster image data for a document page for one pixel cell row, or line, at a time; communicates the raster image data to a digital printing press; and then repeats the process until raster data for all of the pixel lines are communicated to the digital printing press. Generating raster image data in this manner reduces the load on memory and computing resources, as further described herein.

[0019] In the context of this application, a "pixel cell," or "cell" is associated with an atomic spatial unit of an image, such as a raster image of a document page. For example, the raster image of the document page may be viewed as being formed from a rectangular grid of pixel cells. As an example, in accordance with some implementations, a "cell" may be associated with a single pixel may be associated with a particular color value. As another example, in accordance with some implementations, a "cell" may be a collection, or group, of spatially adjacent pixels, and the pixels of the cell may be associated with the same color. For example, the cell may be a block of 4x4 pixels that is associated with a particular color (i.e., the color is homogenous for the cell).

[0020] In accordance with example implementations, a given pixel cell may be a solid color cell, i.e., a pixel cell in which all pixel(s) of the have the same color.

Moreover, in accordance with example implementations, a solid color cell may be by definition opaque, meaning that the solid color pixel cell dominates any pixel cells that are disposed beneath the solid color pixel cell.

[0021] A given pixel cell may be a block of pixels that form an "edge cell." In general, an edge is a multiple pixel, pixel cell (a 4X4 block of pixel cells or other size, as examples) that has more than one color. In accordance with some implementations, one or multiple pixel cells of an edge cell may have some degree of transparency. In this manner, in accordance with some implementations, a pixel of a pixel cell may have a "shading," or transparency level, which defines the degree of transparency (or opacity) of the cell. The pixel cell may have an associated "color shading," which refers to a combination of a color and a degree of transparency for the cell.

[0022] In accordance with some implementations, a pixel may have two levels of shading: a first level (L1); and a second level (L2). In this manner, the L2 level may be relatively more opaque than the first level L1, so that when an L1 pixel cell is disposed beneath an L2 pixel cell, the L2 pixel cell dominates.

[0023] The edge cell may be a continuous tone cell, in accordance with some implementations, such as, for example, an Indigo compressed format (ICF) continuous tone, or "contone."

[0024] In general, the image of an object may be partitioned into pixel cells called "object source cells," or "source cells." The raster image of a document page may be partitioned into pixel cells called "target cells." Moreover, a document page may be associated with "cell lines," which may be viewed as the raster image of the document page being horizontally partitioned into rows. In accordance with example implementations, a cell line extends across the width of the raster image, and the cell line has a height of one cell. As such, in accordance with example implementations, the number of cell lines is equal to the height of the raster image in pixels divided by

the pixel height of the pixel cell, and the number of cells per cell line is equal to the width of the raster image in pixels divided by the pixel width of the pixel cell.

[0025] In general, the objects (text or graphics, which are defined by a PDF file, for example) that are part of a given document page may be associated with different layers. The "layer" associated with an object refers to a plane in which the object lies and which is parallel to the document page. The layer number, or order, represents a depth dimension, or order, of the layer, and in accordance with example implementations, the layer number increases with distance from the plane in which the background of the document page lies.

[0026] Objects of a document page may partially or entirely intersect, or overlap; and whether or not object portions that are overlapped are visible in the raster image of the document page depends on the degrees of transparency (or opaqueness) of the overlapping pixel cells or pixels within the pixel cells (when edge cells are involved). For example, for a given document page, a pixel cell A of object A that is associated with layer number 3 may overlap a pixel cell B of object B that is associated with layer number 2. For this example, object B is located behind object A, and the pixel B may or may not be visible, depending on the degree of opaqueness of the pixel cell A. In this manner, in accordance with example implementations, a given pixel cell may be opaque, nearly opaque, nearly transparent or transparent. In accordance with example implementations, an opaque or nearly opaque pixel cell means that the cell blocks enough light to prevent the viewing of a pixel cell that is disposed at the same position and associated with a lower order layer. Moreover, in accordance with example implementations, a transparent pixel cell means that an underlying pixel cell is fully viewable; and a nearly transparent pixel cell means that values (contones or colors, depending on the particular implementation) for the cell and an underlying cell are combined, or blended. The process of determining a pixel cell value for overlapping, or intersecting, pixel cells is called "blending" herein.

[0027] As an example of the blending, for the example of pixel cell A (the layer 3 pixel cell) and pixel cell B (the layer 2 pixel cell) given above, if pixel cell A is a solid

color pixel cell, then pixel cell B may not be visible (such as in implementations in which a solid color pixel is opaque by definition). If pixel cells A and B are both solid color pixel cells, then pixel cell B is not visible due to pixel cell A being the upper cell. If pixel cell B is a solid color pixel cell and pixel cell A is an edge cell, then pixel cell A may not be visible (such as in implementations in which a solid color pixel is opaque and dominates by definition). If pixel cells A and B are both edge cells, then parts of both pixel cells A and B may be visible, depending on the relative degrees of transparency, or shading, as further described herein.

[0028] In accordance with example implementations, the recomposition engine processes a document description file for purposes of generating a cell line table, which identifies, for each cell line associated with the document page, which objects are associated with the cell line. In other words, for each cell line, the cell line table identifies objects that are partially or fully contained in the cell line and the positions of the contained objects.

[0029] The recomposition engine, in accordance with example implementations, constructs an object intersection table from the cell line table for purposes of identifying intersections of objects (if any) for each cell line. Using the object intersection table, the recomposition engine may then process the cell lines (called "target cell lines" herein) one at a time and communicate raster image data to the digital printing press in corresponding units of data. In this manner, in accordance with example implementations, the recomposition engine may, for a given target cell line, determine whether objects overlap, or intersect, in the given target cell line, and based on a result of this determination, perform a blending of the intersecting source object cells (if any) for purposes of generating the raster image data for the target cell line. Moreover, as described herein, in accordance with example implementations, the recomposition engine may use the cell line intersection table to, for a given target cell line, optimize the generation of raster image data for the target cell line to accommodate the cases in which one or no objects are contained in the cell line or the case in which multiple objects exist for the cell line but do not intersect.

[0030] As described further herein, in accordance with example implementations, the recomposition engine may blend pixel cells for overlapping objects in a manner that conserves memory and computing resources. More specifically, in accordance with example implementations, the recomposition engine may, in general, blend pixel cells that are not edge cells in a compressed domain, in that the blending may process compressed data representing overlapping pixel cells to blend these pixel cells and produce corresponding raster image data without decompressing the data. In accordance with example implementations, the recomposition engine may determine if a pair of pixel cells to be blended are edge pixel cells, and if so, the recomposition engine may first decompress the corresponding data representing these cells and then perform the blending using decompressed data, as further described herein.

[0031] As a more specific example, Fig. 1 depicts a system 100 in accordance with some implementations. In general, the system 100 includes a recomposition engine 114 that receives page description data 116, which indicates, or represents, a description of one or multiple pages of a document to be printed. As an example, in accordance with some implementations, the page description data 116 may be data contained in a portable document file (PDF), which may describe one or multiple document pages to be printed. Moreover, in accordance with some implementations, the page description data 116 may describe static objects and variable objects associated with VDP. Regardless of its particular form, the page description data 116 describes a document page containing one or multiple objects, which the recomposition engine 114 processes to produce raster image data 130 for a digital printing press 160.

[0032] In accordance with example implementations, the raster image data 130 represents a single target cell line, of a document page. As described herein, the recomposition engine 114 constructs a cell line table 118 based on the page description data 116. The cell line table 118 identifies, per cell line, which object are objects are partially or entirely contained in the cell line. Based on the cell line table 118, the recomposition engine 114 generates an object intersection table 120, which, per cell line, identifies the positions of any object(s) contained in the cell line and

whether objects overlap in the cell line. Based on the object intersection table 120, the recomposition engine 114 may then, generate the raster data 130 for each target cell line, as described herein.

[0033] Among the other features of the system 100, in accordance with some implementations, the print server 110 may include one or multiple processors 140 (one or multiple central processing units (CPUs), one or multiple processing cores, and so forth) and a memory 144. In general, the memory 144 is a non-transitory memory that may store data representing machine executable instructions (or software), which are executed by one or multiple processors 140 for purposes of performing techniques that are described herein. For example, in accordance with some implementations, the memory 144 may store machine executable instructions that when executed by the processor(s) 140 may cause the processor(s) 140 to perform functions of the recomposition engine 114 as described herein. The memory 144 may further store data representing initial, intermediate and final versions of the raster image data 130, as well as other data, in accordance with example implementations.

[0034] In accordance with some implementations, the memory 144 may be formed from semiconductor storage devices, memristors, phase change memory devices, non-volatile memory devices, volatile memory devices, a combination of one or more of the foregoing memory storage technologies and so forth.

[0035] In accordance with some implementations, the recomposition engine 114 may be partially or wholly based in software (i.e., formed by one or more of the processors 140, executing machine executable instructions). However, in accordance with further example implementations, the recomposition engine 114 may be formed partially or in whole from one or multiple hardware circuits such as one or multiple field programmable gate arrays (FPGAs) or application specific integrated circuits (ASICs).

[0036] As depicted in Fig. 1, in accordance with some implementations, the raster image data 130 may be communicated to the digital printing press 160 over network fabric 150. In accordance with example implementations, the network fabric 150

may be formed from components and use protocols that are associated with any type of communication network such as (as examples) Fiber Channel Networks, iSCSI networks, ATA over Ethernet (AoE) networks, HyperSCSI networks, local area networks (LANs), wide area networks (WANs), global networks (the Internet, for example), or any combination thereof.

[0037] Fig. 2A is an illustration 200 depicting an example cell line table 118 that is generated by the recomposition engine 114 (Fig. 1) from an example page 204 of a document to be printed. The document page 204 contains, for this example, three objects 205, 206 and 207. Moreover, the object 206 is associated with a lower layer than the object 207, and the object 207, where the objects 206 and 207 overlap, is opaque or partially opaque (i.e., the portion of the object 206 where the objects 206 and 207 overlap cannot be seen).

[0038] The corresponding cell line table 118 may contain rows 220, where each row 220 describes the object or objects that may be contained in a cell line that is associated with the row 220. In this manner, in accordance with example implementations, the cell line table 118 includes a column 208, which identifies the particular target page 204. The cell line table 118 includes a column 210 identifying the cell lines associated with the target page 204. The number of cell lines may be equal or less than the number of vertical pixels on the page, depending on the cell size. For example, if the page 204 contains 8000 pixels in height and the compression ratio is 4:1 (i.e., 4 associated pixels per cell), then there are 2000 cell lines and 2000 corresponding rows 220 of the table 118. Each cell line, in turn, may contain, or hold, zero, one or more objects. For example, the row 220-1 corresponds to example cell line number one, and example cell line 220-4 contains information for cell line number 4559.

[0039] In accordance with example implementations, the table 118 further includes a column 212 identifying an object count for the number of implicated objects in the target cell line. For the example page 204, the object 205 does not coincide with any other object. Therefore, for the example cell line table 118, the object 205 is represented in the table 118 (as Object ID=1) with corresponding rows 220 having

an object count of "1." To the contrary, the objects 206 and 207 (represented by Object ID=2 and Object ID=3, respectively overlap, and correspondingly, rows 220 of the cell line table 118 have an object count of "2" where the objects 206 and 206 overlap. As more specific examples, row 220-1 of the cell line table 118 contains an object count of "0" representing that no objects are contained in the first cell line of the page 204. As another example, row 220-3 of the table 118, is associated with cell line number 12, contains the objects 206 and 207, and has an object count of "2."

[0040] In accordance with example implementations, the cell line table 118 includes an object identification and cell line column 214, which contains an identifier for each object that is associated with the cell line and the particular cell line number of the object. For example, row 220-3 of the cell line table 118, which corresponds to cell line number 12, has the following entries: "2(3)," which identifies object number "2" (the object 206), and the "(3)" represents that the cell line number 12 contains row number three of the object 206; and the column 214 entry for the row 220-3 further represents that cell line number 12 contains object number 3 (i.e., the object 210) and contains cell line number two for the object.

[0041] Fig. 2B depicts an example target cell line 250 in its final state, accordance with example implementations. In particular, the target cell line 250 corresponds to row line 220-4 of the cell line table 118 and thus, corresponds to the intersection of objects 206 and 207 in cell line number 4559. The target cell line 250 contains a group 254 of cells in cell positions numbers 1 to 9, which contain background fill color as there are no objects for these cells. Due to the grouping of contiguous cells that have the same cell value, in accordance with example implementations, the group of cells 254 may be represented in the raster data 130 (Fig. 1) by compressed data, such as, for example, data that represents run length encoding or Indigo compression format (ICF) encoding, as examples. The target cell line 250 for the example of Fig. 2B also contains cells 258 in cell positions 10 to 19, which correspond to the intersection of the objects 206 and 207 in cell line number 4559. The cells 258, which have the same associated values, may also be encoded, or compressed. The target cell line 250 further includes a group of cells 262 in cell

positions 20 to 28, which corresponds to the object 207, where the objects 206 and 207 do not overlap. In a similar manner, due to the cells 262 corresponding to the same value, the cells 262 may be compressed, or encoded. Finally, for example target cell line 250, the cell line 250 contains another group 266 of cells in positions 29 and 30, which are associated with the background color, or no fill.

[0042] In accordance with example implementations, the recomposition engine 114 (Fig. 1) blends overlapping object source cells using reverse z-order blending, where "z" represents the page depth dimension. In particular, Fig. 3 is an illustration 300 of z-order blending for three target cells 310, 314, 318 and 320, in accordance with example implementations. Moreover, Fig. 3 depicts source cells associated with four non-background layers 302, 303, 304 and 305. In particular, for this example, the layer 305 is the lowermost layer (i.e., layer number 1), the layer 304 is the next highest layer (i.e., layer number 2), the layer 303 is the third highest layer (i.e., layer number 3), and the layer 302 is the uppermost layer (i.e., layer number 4). The reverse z-order blending, as its name implies, proceeds in a direction opposite to the z axis (i.e., proceeds in a direction into the page).

[0043] The recomposition engine 114 performs the reverse z-order blending by beginning with the uppermost layer 302 and stopping when an opaque or nearly opaque source cell is encountered. For example, for the target cell 310, the reverse z-order blending views the cells along a reverse z direction 330. In this direction, the blending first encounters a nearly transparent source cell 332 that is associated with the uppermost layer 302. Because the source cell 332 is neither nearly opaque nor opaque, the processing continues along the direction 330, and as shown, source cells 334 and 336, which are associated with the next two layers 303 and 304 are transparent. Therefore, processing along the direction 330 continues to the lowest layer source cell 338, which, for this example, is opaque or nearly opaque. Accordingly, the recomposition engine 114 assigns the value of the source cell 338 to the target cell 310.

[0044] The value for the target cell 318 is derived by processing in a reverse z-order direction, as indicated at reference numeral 350. As shown, source cells 352

and 354, which are associated with the uppermost 302 and next uppermost 303 layers, are transparent. However, a source cell 356 of the next layer 304 is opaque or nearly opaque. Therefore, the reverse z-order processing stops at the second layer 304, as the value of the source cell 356 sets the value for the target cell 318. It is noted that the reverse z-order processing ends at the layer 304, as due to the opacity of the source cell 356, the values of any source cells below the cell 356 do not contribute to or affect the value of the target cell 318. In a similar manner, for purposes of determining the color value for the target cell 320, the recomposition engine 114 proceeds in a reverse z-order direction, as indicated by reference numeral 370. The processing ends at the layer 303, as the corresponding source cell 374 is opaque or nearly opaque, thereby providing the value for the target cell 320.

[0045] Fig. 4 depicts an illustration 400 of an example object intersection table 120 for an example page 410. For the page 410, a cell is formed by a block of pixels, and the cells that correspond to the page 410 are represented by a grid. The page 410 contains a source object 420, which is overlapped by another object 422. Moreover, page 410 contains a third object 424, which does not overlap either object 420 or 422. In the corresponding object intersection table 120 that is depicted in Fig. 4, rows 460 of the table 120 correspond to the pixel cell lines associated with the page 410.

[0046] In accordance with example implementations, each row 460 of the object intersection table 120 identifies the object intersection(s), if any, for an associated cell line. In accordance with example implementations, the object intersection table 120 includes a column 450 that contains a cell line identifier (1, 2, 3, and so forth) identifying the cell line for the associated row 460. Moreover, the object intersection table 120 includes a column 452 that identifies information pertaining to the objects that are contained in the associated cell line.

[0047] For example, row 460-2 contains information pertaining to cell line number "15," which is highlighted and assigned reference numeral 430 on the page 410. For the row 460-2, the column 452 contains three entries: an entry for each object of the

cell line. Each entry, in turn, describes an identifier for the object, the cell line on which the object begins, the horizontal cell offset for the object, and the horizontal length of the object. For example, for the first entry in column 452 for the row 460-2, the entry is "1:11:5:12," which means object number 1 (i.e., object 420) begins on cell line number "11," begins on cell "5" of the cell line, and has a length of "12" contiguous cells.

[0048] As also depicted in Fig. 4, in accordance with some implementations, the object intersection table 120 contains a column 454, which indicates whether the objects of the associated cell line intersect, or overlap. In this manner, in accordance with some implementations, the column 454 contains a Boolean value that is "True" to indicate object overlap and is "False" to indicate that no objects overlap in the associated cell line. For example, for example row 460-1, the object 420 (i.e., object "1") appears in the associated cell line and no other objects appear in the cell line; and as such, the corresponding Boolean value in column 454 is "False." However, for example row 460-2, three objects overlap, thereby corresponding to a Boolean value of "True" in the column 454.

[0049] Referring to Fig. 5, in accordance with example implementations, the recomposition engine 114 uses the object intersection table 120 pursuant to a technique 500 for purposes of generating raster image data for a document page. In particular, pursuant to the technique 500, the recomposition engine 114 first generates (block 504) an object intersection table for the page and then reads (block 508) source cell data for the next target cell line to be processed. From this data, the recomposition engine 114 determines (decision block 512) whether any source object is implicated for the target cell line. If not, then, in accordance with example implementations, the recomposition engine 114, pursuant to block 516, sets the raster image data equal to the encoded background data and communicates (block 520) the raster image data for the target cell line to the printing press 160 (Fig. 1). Moreover, if a determination is made (decision block 524) that another target cell line is to be processed, control returns to block 508.

[0050] If at least one source object is implicated for the target cell line (per decision block 512), the recomposition engine 114 determines (decision block 530) whether a single source object is implicated for the target cell line; and if so, the recomposition engine 114 sets the raster image data equal to the encoded source cell data, pursuant to block 534, and communicates the raster image data to the printing press pursuant, to block 520.

[0051] In accordance with example implementations, if two or more source objects are implicated for the target cell line, then, pursuant to block 538, the recomposition engine 114 initializes a transparent target cell line, pursuant to block 538. If the recomposition engine 114 then determines (decision block 542) that an intersection or overlap between source objects occur for the cell line, the recomposition engine 114 decodes (block 546) the source cell line data (e.g., decodes the run length encoding) and blends (block 554) the source pixel cells for the target cell line based on the decoded data. Otherwise, if multiple objects do not overlap for the target cell line (as determined in decision block 542), the recomposition engine 114 combines (block 558) the source cell lines without decoding to form the raster image data.

[0052] Fig. 6 depicts a technique 600 to blend a target pixel cell and a source pixel cell according to an example implementation. Referring to Fig. 6, the technique 600 includes determining (decision block 604) whether the target cell is an edge cell. If not, in accordance with example implementations, the target cell is a single color cell, which dominates due to the reverse z order blending. In other words, a single color cell, in accordance with example implementations, is opaque, and when a single color cell appears in the target cell line, then no other pixel cell is visible below it. As such, if the target cell is a single color cell (and not an edge cell), the blending terminates.

[0053] If, however, a determination is made pursuant to decision block 604 that the target cell is an edge cell (i.e., the target cell is not a single color cell), then it is possible that pixels of the source cell will be visible beneath the target cell. The technique 600 includes determining (decision block 608) whether the source cell is a single color cell. If so, then, accordance with example implementations, the ICF code

for the blended pixel cell may be derived without decompressing either cell. In this manner, an ICF code for the blended cell may be derived by looking up an ICF code (from a lookup table, for example) that represents transparent pixels of the target cells being replaced with solid color pixel cells of the source pixel cell. This is illustrated in Fig. 7A for an example target pixel cell 704 (an edge cell) being blended with a solid color source pixel cell 716, which contains opaque, color pixels 718. The target pixel cell 704 includes transparent pixels 712, and as such, after the blending, the color pixels 718 of the source pixel 716 are visible, as depicted by blended pixel cell 720. For purposes of deriving the blended cell 720, the technique 600 may include lookup an ICF code from a lookup table, based on the ICF code for the target pixel cell 704, an edge cell, and the ICF code for the single color source cell 716. In other words, in accordance with example implementations, the blending occurs in the ICF compressed domain without decompressing the target 704 or source 716 pixel cells.

[0054] Referring to Fig. 6, If, pursuant to decision block 608, a determination is made that the source cell is an edge cell (the "no" prong of decision block 608), then, according to example implementations, the technique 600 includes decompressing (block 620) the ICF codes of the target and source pixel cells to generate data representing the pixel cells as corresponding pixel grids. In other words, the blending is performed "on a canvas" without using the relatively compact ICF codes for the pixel cells. Therefore, pursuant to block 624, the pixel grids are merged on a pixel by pixel basis to determine a blended pixel cell.

[0055] As a more specific example Fig. 7B is an illustration 730 in which a target cell 732, an edge cell, is blended with a source cell 738, which is also an edge cell, to produce a blended pixel cell 750. The target pixel cell 732 contains relatively opaque pixels 734 and relatively transparent pixels 736; and the source pixel cell 738 contains relatively transparent pixels 742 and relatively opaque pixels 740. In the blended pixel cell 750, the opaque pixels 734 of the source pixel cell 732 dominant; the opaque pixels 740 of the source pixel cell 738 appear where these pixels 740 are overlapped by the transparent pixels 736 of the target pixel cell 732; and the transparent pixels 736 of the target pixel cell 732 remain where the pixels

736 overlap the transparent pixels 742 of the source pixel cell 738. Thus, as can be seen in Fig. 7B, the blended pixel cell 750 for this example contains three color shades.

[0056] In accordance with some implementations, the edge cell may be limited to a predetermined number of color shades. For example, in accordance with some implementations, this number limit may be two. However, as depicted by example blended pixel cell 740 of Fig. 7B, this limit may be exceeded when edge cells are blended pixel by pixel in the uncompressed domain.

[0057] Referring to Fig. 6, in accordance with example implementations, the blending may include, pursuant to decision block 628, whether the number of color shades of the blended cell (called the "first blended cell" in block 628) is greater than a threshold (for this example, greater than two). If so, pursuant to block 632 of the technique 600, one or multiple color shades of the first blended cell may be replaced to decrease the number of color shades so that the number of shades complies with the threshold (the number of shades is two or less, for this example).

[0058] More specifically, Fig. 7C is an illustration 760 of the reduction of the number of color shades of the blended pixel cell 750 (Fig. 7B) to produce a second blended cell 766 that has two color shades. In accordance with some implementations, the blending may determine distances between the colors and combined colors that are closest together. For the example of Fig. 7C, the first blended cell 760 has three color shades, with, for this example, the pixels 740 having a color shade that is closer in color distance to the pixels 736 than to the pixel 740. In accordance with some implementations, the pixels having the lowest transparency, here the pixels 736, may be replaced with the pixels that are the closest in color distance, here the pixels 740. Other techniques may be used to combine or replace color shades for purposes of reducing the number of color shades of the final blended cell, in accordance with further example implementations.

[0059] It is possible for the blending of two pixel cells to produce a single color pixel cell. In this manner, Fig. 7D is an illustration 780 of the blending of a target edge

782 and a source edge cell 788, which results in a single color, blended cell 792. Thus, as depicted in Fig. 7D, the target pixel cell 782 may contain pixels 786 associated with a particular color and transparent pixels 784. The source pixel cell 788 contains single color pixels 787, which have the same color as the single color pixels 786 of the target pixel cell 782; and the source pixel cell 788 contains transparent pixels 785. The single color pixels 786 dominant in the blended pixel cell 792, and for this example, all of the transparent pixels 784 of the target pixel cell 782 are replaced in the blended pixel cell 792 with the solid pixels 787 of the source pixel cell 788.

[0060] Fig. 8 depicts an illustration 800 of the blending of object source cells to create a target cell line in accordance with example implementations. In this manner, the recomposition engine 114 first initializes a transparent target cell line (i.e., the target cell line in its initial state) and then adds the source pixel cell(s) from an uppermost layer 810 (i.e., layer number 4 for this example). For this example, layer number 4 contains object source pixel cells 814 and 815. The source pixel cell 815 is an edge cell. The source pixel cell 814 is a single color cell, which dominates the corresponding target cell location (nothing appears from below the pixel cell 814). Therefore, no blending occurs for the pixel cell location. Because the initial target cell line is transparent, the source pixel cells 814 and 815 are added to the target cell line to create an intermediate target cell line 818.

[0061] For this example, no objects exist in the next lower layer (here, layer number 3). However, the next lower layer (here, layer number 2) contains object source pixel cells 820, 824 and 826: the source pixel cell 820 is an edge cell; and the source pixel cells 824 and 826 are single color pixel cells. Because the target pixel cell line 818 contains transparent target pixel cells that corresponds to the positions of the source pixel cells 820 and 824, the recomposition engine 114 copies the source pixel cells 820 and 824 into to the next intermediate target cell line 814. Because the source pixel cell 826 is a single color pixel cell and the overlapping target pixel cell 815 is an edge cell, the recomposition engine 114 may blend the pixel cells 815 and 826 by looking up the ICF code of the corresponding pixel cell

828 (that appears in the target cell line 814) without decompressing the ICF codes for the pixel cells 815 and 826.

[0062] The lowest layer for the sources (layer number 1 for this example) contains three object source pixel cells 840, 844 and 848. The source pixel cell 840 is a single color cell, and the target cell line 830 has a corresponding transparent pixel cell. Therefore, the recomposition engine 114 copies the source pixel cell 840 into the next intermediate target cell line 860. The source pixel cell 848 is dominated by the single color pixel cell 824. Accordingly, the recomposition engine 114 copies the single color pixel cell 824 into the intermediate target cell line 860.

[0063] The source pixel cell 844 is an edge cell and is overlapped by the pixel cell 820 of the intermediate target cell line 830. Because the pixel cell 820 is also an edge cell, the recomposition engine 114 decompresses the ICF codes of the pixel cells 820 and 844 and performs the blending pixel by pixel in the uncompressed domain to determine a corresponding blended pixel cell 850 in the intermediate target cell line 860.

[0064] In accordance with example implementations, the recomposition engine 114 may blend the target cell line 860 with a background color (i.e., blend the portions of the pixels that are not opaque or nearly opaque with the background color) to form the final target cell line; and then, the recomposition engine 114 may communicate raster image data representing this target cell line, in its final state, to the digital printing press 160.

[0065] Referring to Fig. 9, thus, in accordance with example implementations, a technique 900 includes receiving (block 904) compressed data that represents a page of a document that is associated with a plurality of cell lines. A given cell line of the plurality of cell lines includes a given cell, and the given cell is associated with a plurality of intersecting objects. The technique 900 includes blending a first cell associated with a first object with a second cell associated with at least one other object to provide printer raster image data for the given cell, pursuant to block 908. The blending may include determining whether the first and second cells are both edge cells; and in response to determining that the first and second cells are both

edge cells, decompressing the compressed data corresponding to the first and second cells to provide decompressed data, and blending the first and second cells based on the decompressed data.

[0066] Referring to Fig. 10, in accordance with example implementations, a non-transitory machine readable storage medium 1000 may store machine executable instructions 1010 that, when executed by a machine, cause the machine to provide a target cell line for a document page, where the target cell line includes a target cell; and in response to the target cell having a plurality of color shades and a source cell that corresponds to an object of the page and corresponds to a target cell having a plurality of color shades, decompressed data representing the source cell, blend the source cell and the target cell based on the decompressed data to provide a blended cell, and replace the target cell with the blended cell. The instructions, when executed by the machine, may cause the machine to, in response to one of the target or source cells having a single color, blend the source cell and the target cell based on compressed data representing the source cell.

[0067] Referring to Fig. 11, in accordance with example implementations, an apparatus 1100 includes a processor 1110 and a memory 1114. The memory 1114 stores instructions 1118 that, when executed by the processor 1110, cause the processor 1110 to provide a target cell line for a document page, where the target cell line includes a target cell and access compressed data representing a page of a document that is associated with a plurality of cell lines and a plurality of objects. A given cell line is associated with a first cell of a first object and associated with the target cell. The instructions 1011, when executed by the processor 1110, cause the processor 1110 to, in response to a determination that the first cell and the target cell are each associated with a plurality of color shades, decompress the compressed data corresponding to the first cell to provide decompressed data corresponding to the first cell, and merge the first cell and the target cell based on the decompressed data. The instructions 1118, when executed by the processor 1110, cause the processor 1110 to communicate data representing the target cell line to a printer.

[0068] While the present disclosure has been described with respect to a limited number of implementations, those skilled in the art, having the benefit of this disclosure, will appreciate numerous modifications and variations therefrom. It is intended that the appended claims cover all such modifications and variations

What is claimed is:

1. A method comprising:

receiving compressed data representing a page of a document associated with a plurality of cell lines, wherein a given cell line of the plurality of cell lines comprises a given cell, and the given cell is associated with a plurality of intersecting objects; and

blending a first cell associated with a first object of the plurality of objects with a second cell associated with at least one other object of the plurality of objects to provide printer raster image data for the given cell, wherein the blending comprises:

determining whether the first and second cells are both edge cells; and
in response to determining the first and second cells are both edge cells, decompressing the compressed data corresponding to the first and second cells to provide decompressed data, and blending the first and second cells based on the decompressed data.

2. The method of claim 1, further comprising:

in response to determining that at least one of the first and second cells is not an edge cell, blending the first and second cells without decompressing the compressed data corresponding to the first and second cells.

3. The method of claim 1, wherein:

the first and second cells are both edge cells; and
blending the first and second based on the decompressed data comprises
blending the first and second cells based on pixel by pixel comparisons of color shades of the first and second cells.

4. The method of claim 1, wherein:

the second cell is part of a target cell line; and
the blending of the first and second cells further comprises replacing the second cell in the target cell line with a blended cell formed from the first and second cells.

- 1 5. The method of claim 4, further comprising:
2 storing raster data representing the target cell line in a memory;
3 communicating the raster data representing the target cell line to a printer;
4 and
5 reusing the memory to generate raster data representing another target cell
6 line of the document page.
- 1 6. The method of claim 1, wherein:
2 the first cell comprises an edge cell comprising a transparent portion;
3 the second cell comprises a homogenous color cell associated with a given
4 color;
5 blending the first and second source cells comprises determining a code
6 representing the first source cell with the transparent portion replaced with the given
7 color.
- 1 7. The method of claim 1, further comprising:
2 creating a target cell line for the given cell line, wherein the blending
3 comprises determining data representing a first cell of the target cell line based on
4 the first and second cells.
- 1 8. An article comprising a non-transitory machine readable storage medium to
2 store instructions that, when executed by a machine, cause the machine to:
3 provide a target cell line for a document page, the target cell line comprising a
4 target cell;
5 in response to the target cell having a plurality of color shades and a source
6 cell corresponding to an object of the page and corresponding to the target cell
7 having a plurality of color shades, decompress data representing the source cell,
8 blend the source cell and the target cell based on the decompressed data to provide
9 a blended cell, and replace the target cell with the blended cell; and

10 in response to one of the target cell or the source cell having a single color,
11 blend the source cell and the target cell based on compressed data representing the
12 source cell.

1 9. The article of claim 8, wherein the storage medium to store instructions that,
2 when executed by the machine, causes the machine to:
3 store raster data representing the target cell line in a memory of the machine;
4 communicate the raster data representing the target cell line to a printer; and
5 reuse the memory to generate raster data representing another target cell line
6 of the document page.

1 10. The article of claim 8, wherein:
2 the source cell has a plurality of color shades;
3 the target cell has a plurality of color shades; and
4 the storage medium to store instructions that, when executed by the machine,
5 causes the machine to:
6 determine a first blended cell, wherein the first blended cell has a
7 number of colors; and
8 process the first blended cell to provide a second blended cell, wherein
9 the second blended cell has a number of colors less than the number of colors of the
10 first blended cell.

1 11. The article of claim 10, wherein:
2 the first blended cell has a first color, a second color and a third color; and
3 the storage medium to store instructions that, when executed by the machine,
4 causes the machine to:
5 determine a first distance between the first color and the second color;
6 determine a second distance between the first color and the third color;
7 compare the first and second distances; and
8 replace the first color with either the second color or the third color
9 based on a result of the comparison.

1 12. An apparatus comprising:
2 a processor; and
3 a memory to store instructions that, when executed by the processor, cause
4 the processor to:
5 access compressed data representing a page of a document
6 associated with a plurality of cell lines and a plurality of objects, wherein a given cell
7 line of the plurality of cell lines is associated with a first cell of a first object of the
8 plurality of objects and associated with a target cell of a target cell line;
9 in response to a determination that the first cell and the target cell are
10 each associated with a plurality of color shades:
11 decompress the compressed data corresponding to the first cell
12 to provide decompressed data corresponding to the first cell;
13 merge the first cell and the target cell based on the
14 decompressed data; and
15 communicate data representing the target cell line to a printer.

1 13. The apparatus of claim 12, wherein the compressed data comprises data
2 representing the first cell in an Indigio Compressed Format (ICF).

1 14. The apparatus of claim 12, wherein:
2 the first and second cells are both edge cells; and
3 the processor merges the first cell and the target cell based on pixel by pixel
4 comparisons of color shades of the first cell and the target cell.

1 15. The apparatus of claim 12, wherein:
2 the processor, in response to a determination that one of the first cell or the
3 target cell is not associated with a plurality of color shades, merges the first cell and
4 the target cell to provide compressed data representing a merged cell without
5 decompressing the compressed data.

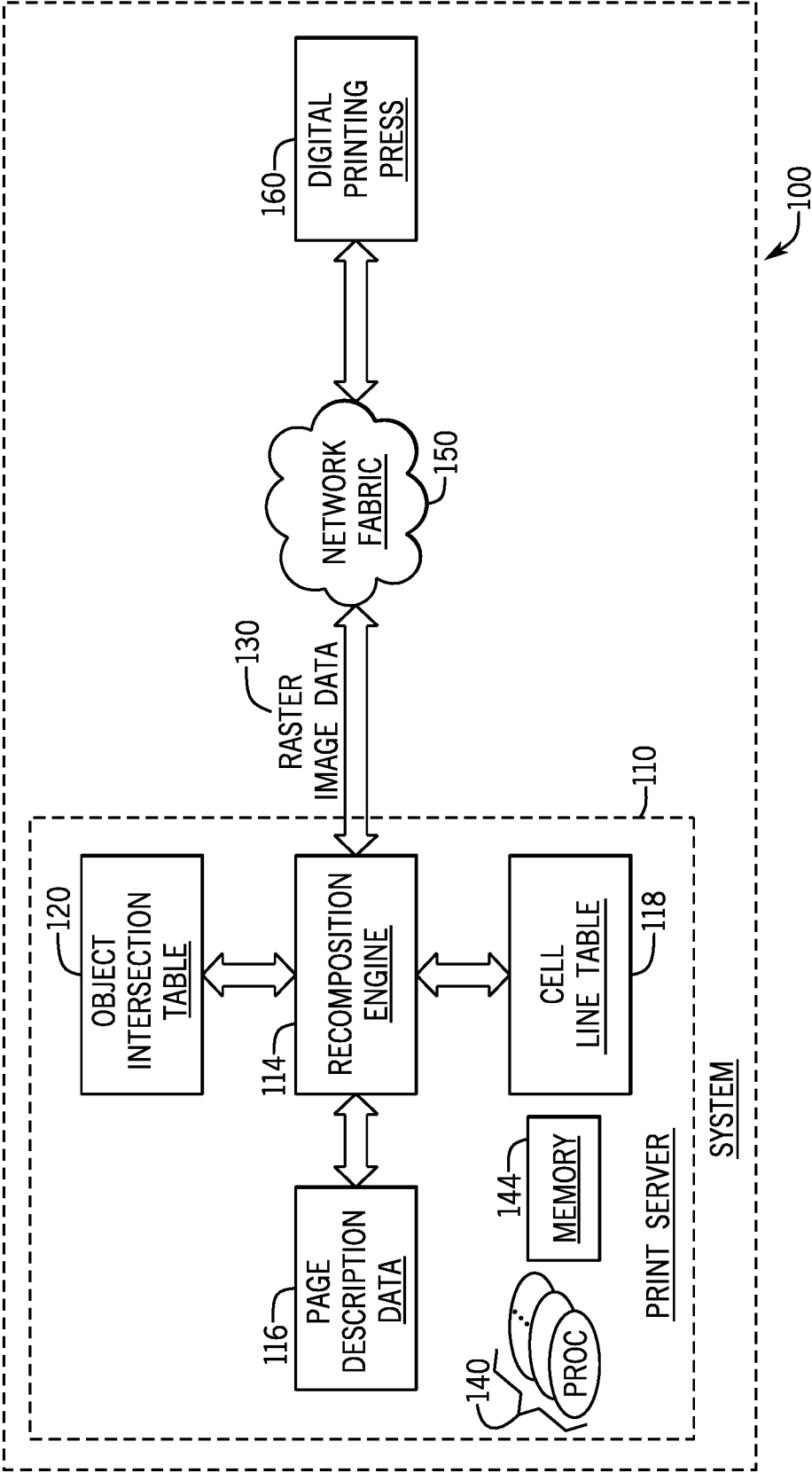
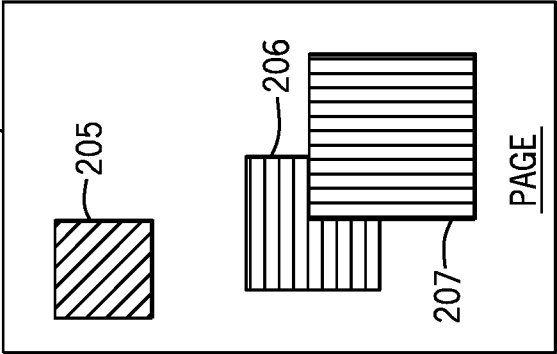


FIG. 1

FIG. 2A

118			
208	210	212	214
PAGE	CELL LINE	OBJECT COUNT	OBJECT ID & CELL LINE
1	1	0	-
	2	0	-
	3	1	1(1)
	4	1	1(2)
	5	1	1(3)
	6	1	1(4)
	7	1	1(5)
	8	0	-
	9	0	-
	10	1	2(1)
	11	2	2(2),3(2)
	12	2	2(3),3(2)
	...		
	...		
	4559	2	2(1500),3(1488)
	4560	1	3(1489)
	4561	1	3(1490)
	...		
	4560	1	4



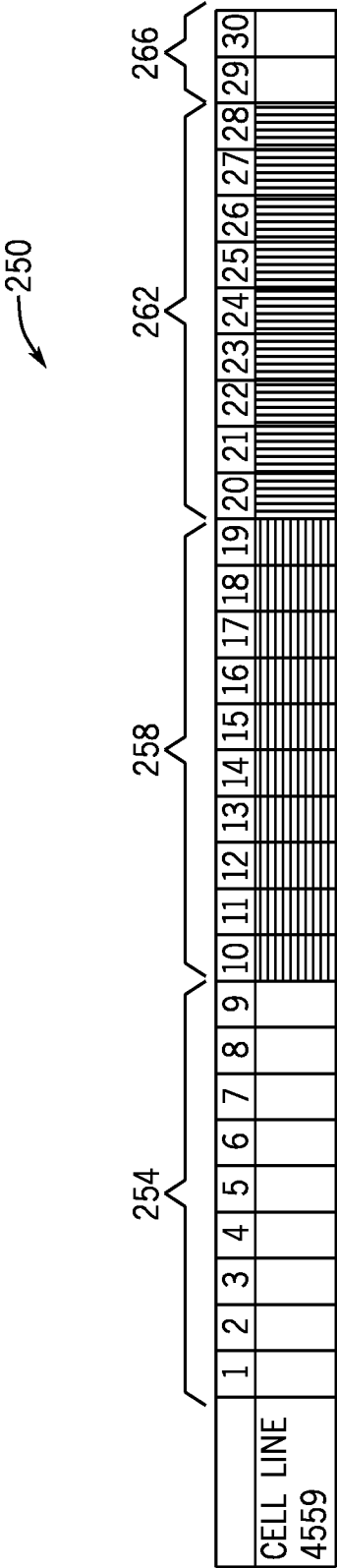


FIG. 2B

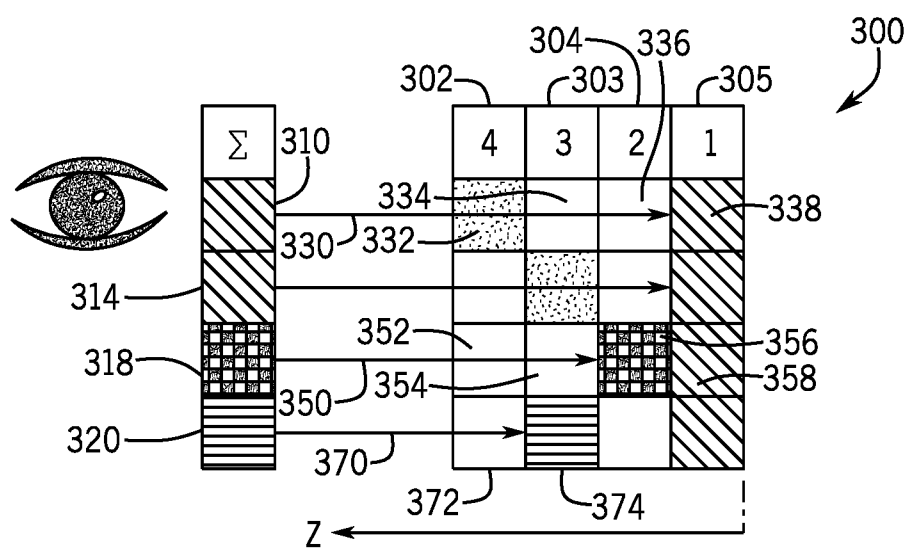


FIG. 3

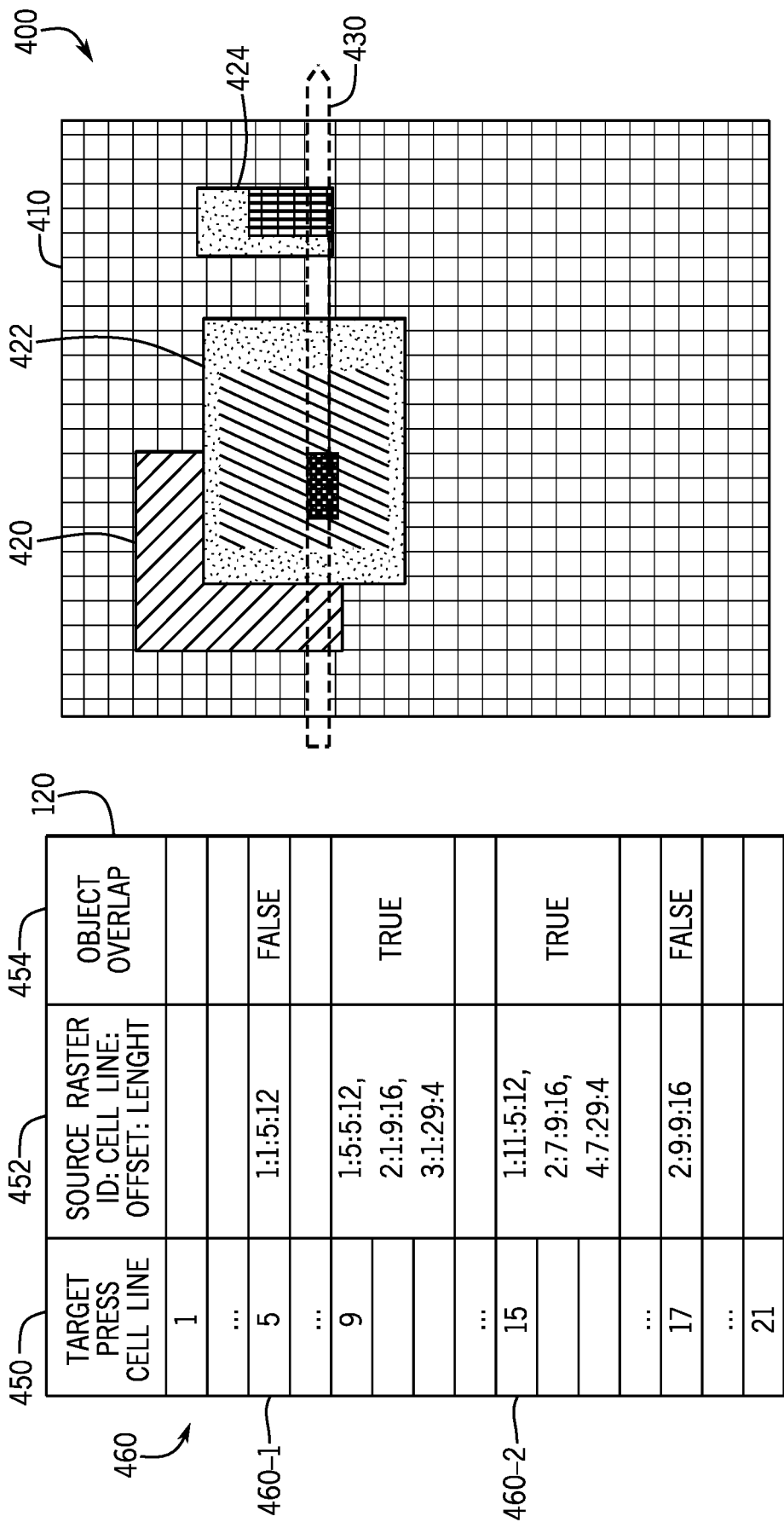


FIG. 4

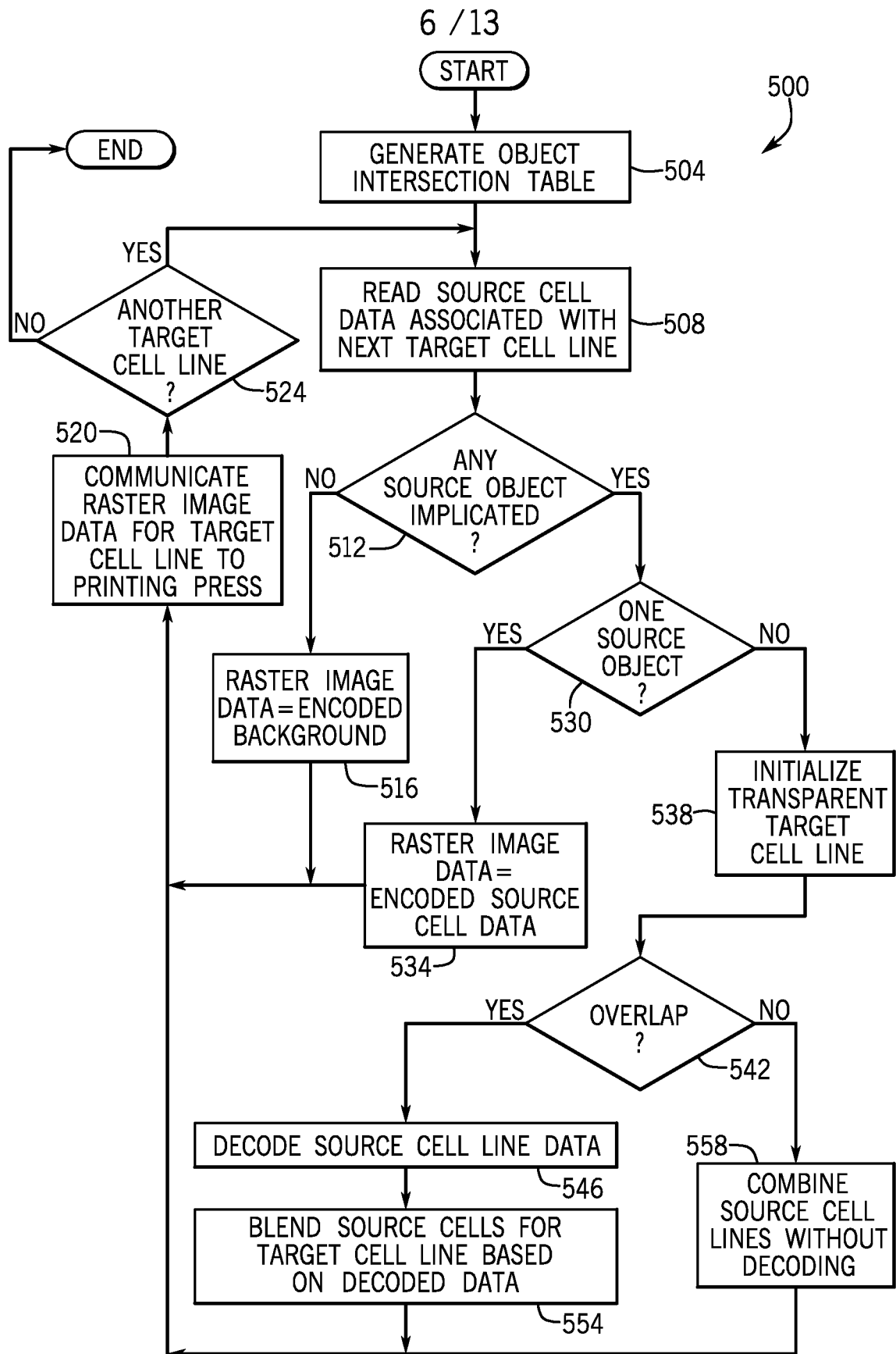


FIG. 5

7 / 13

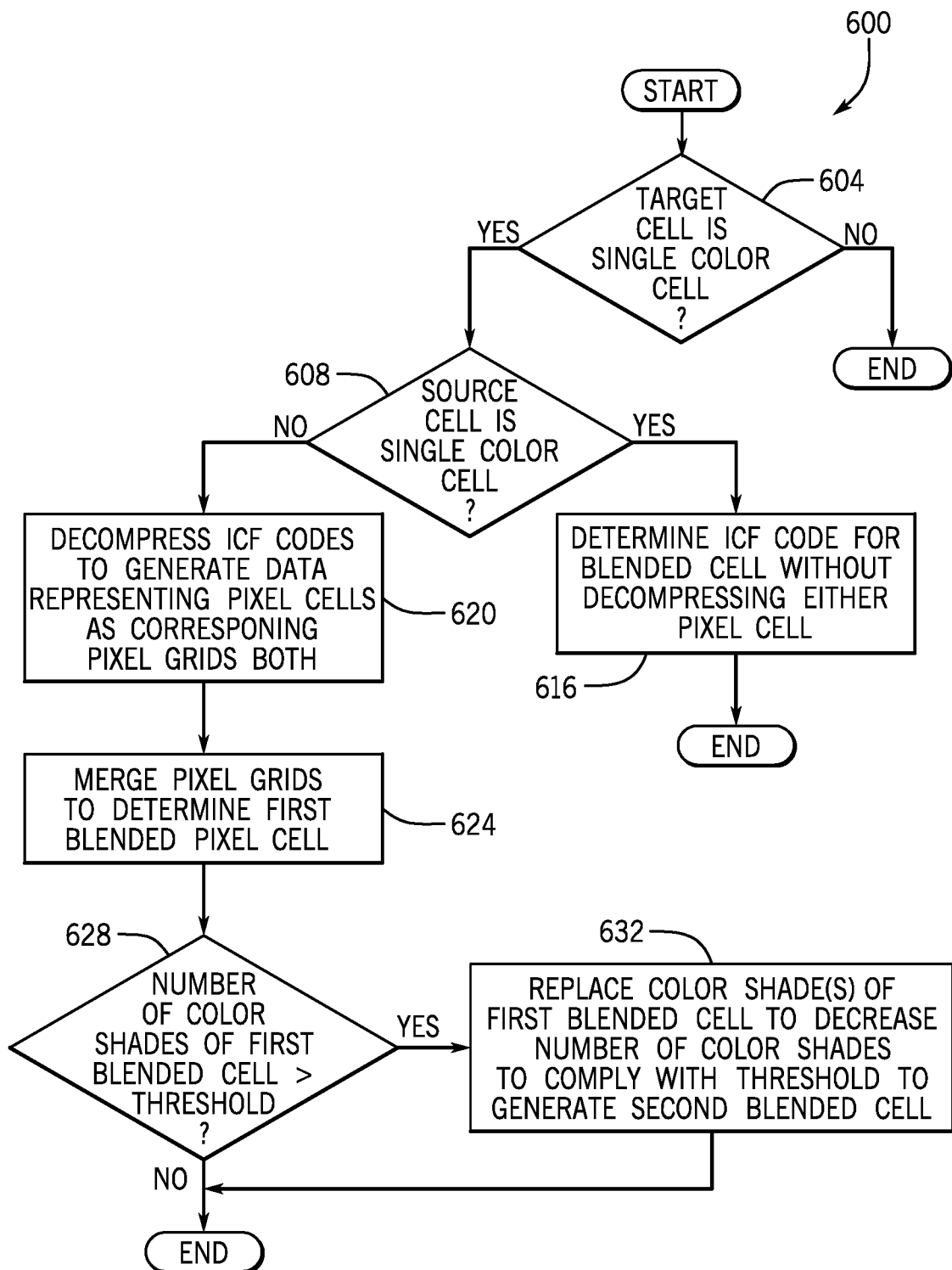
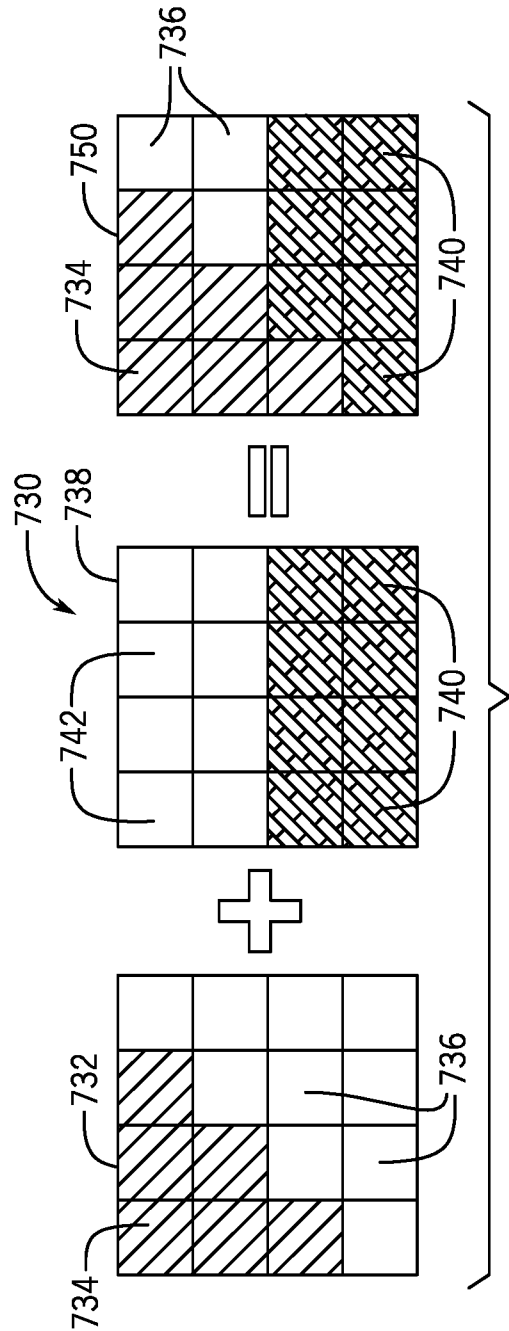
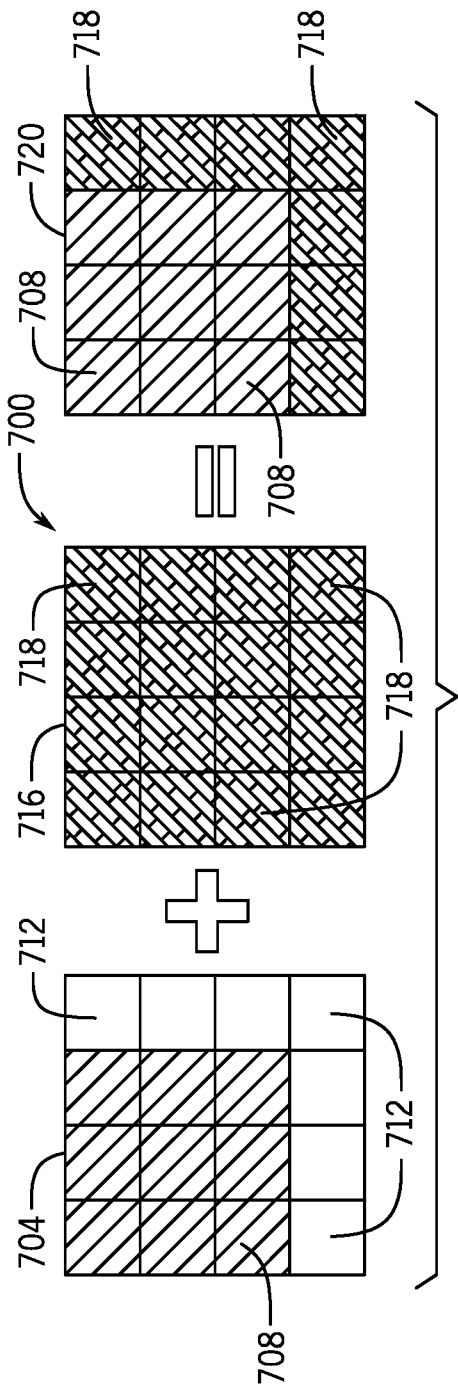
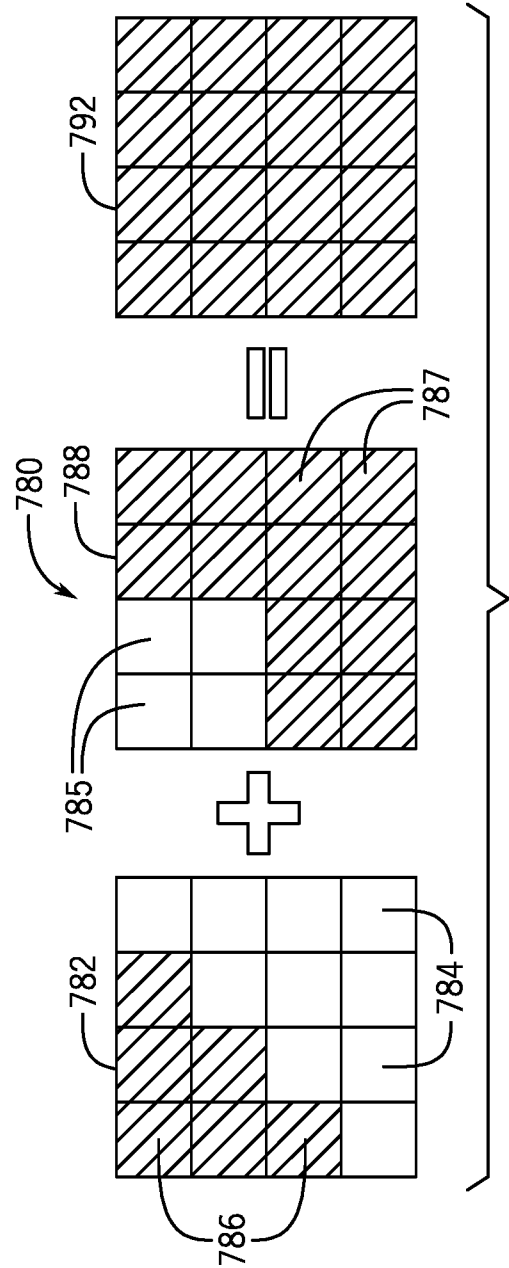
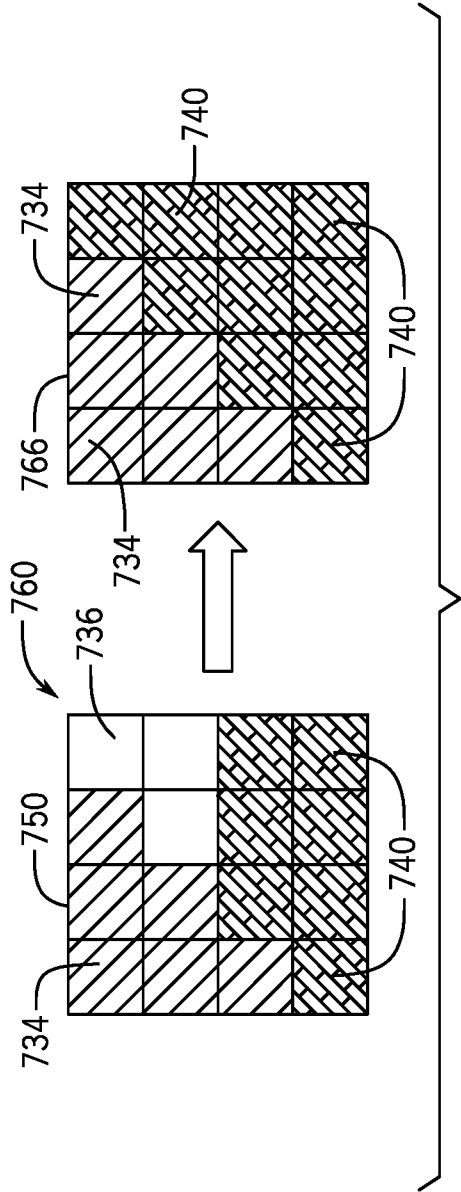


FIG. 6





10 / 13

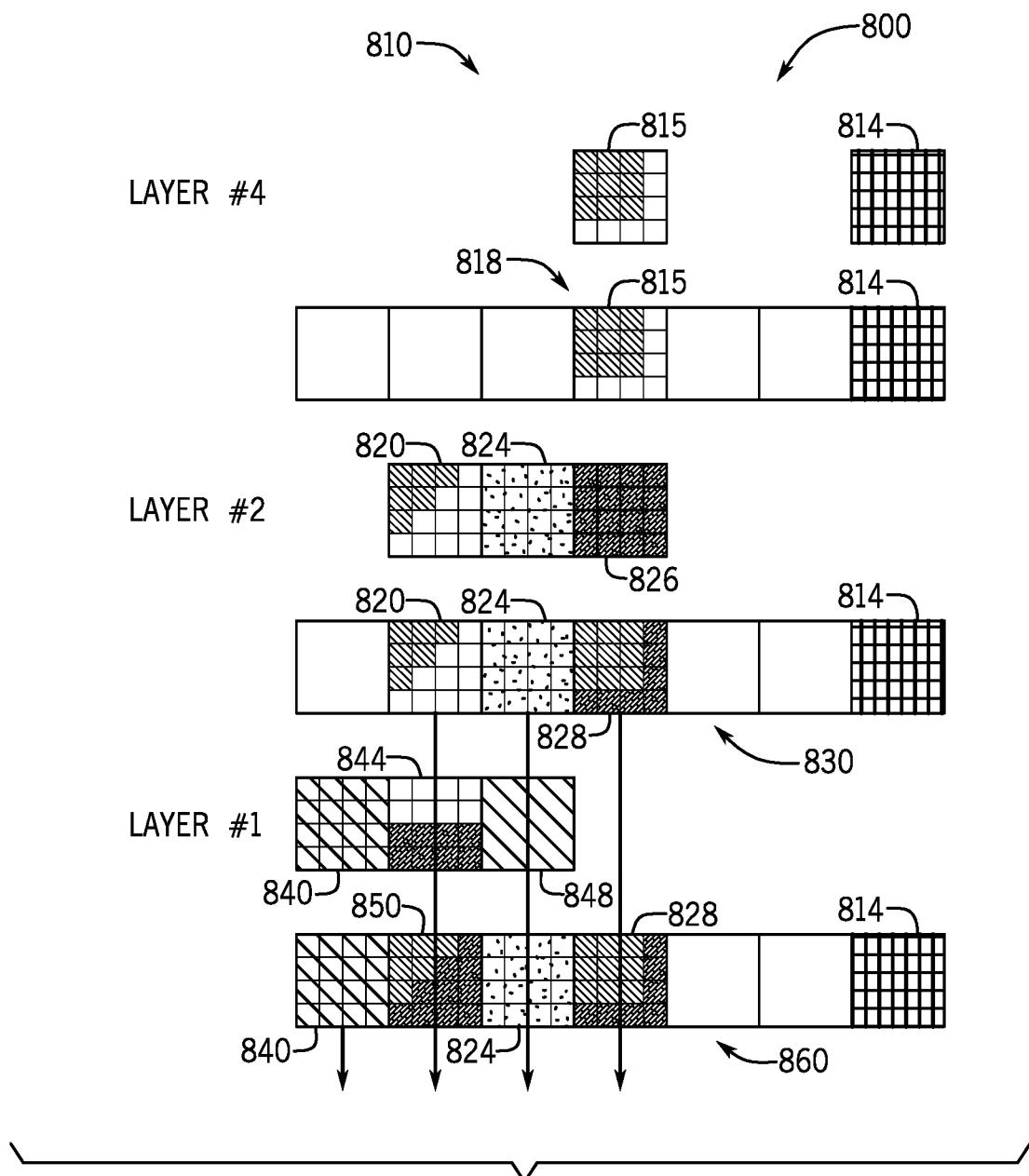


FIG. 8

11 / 13

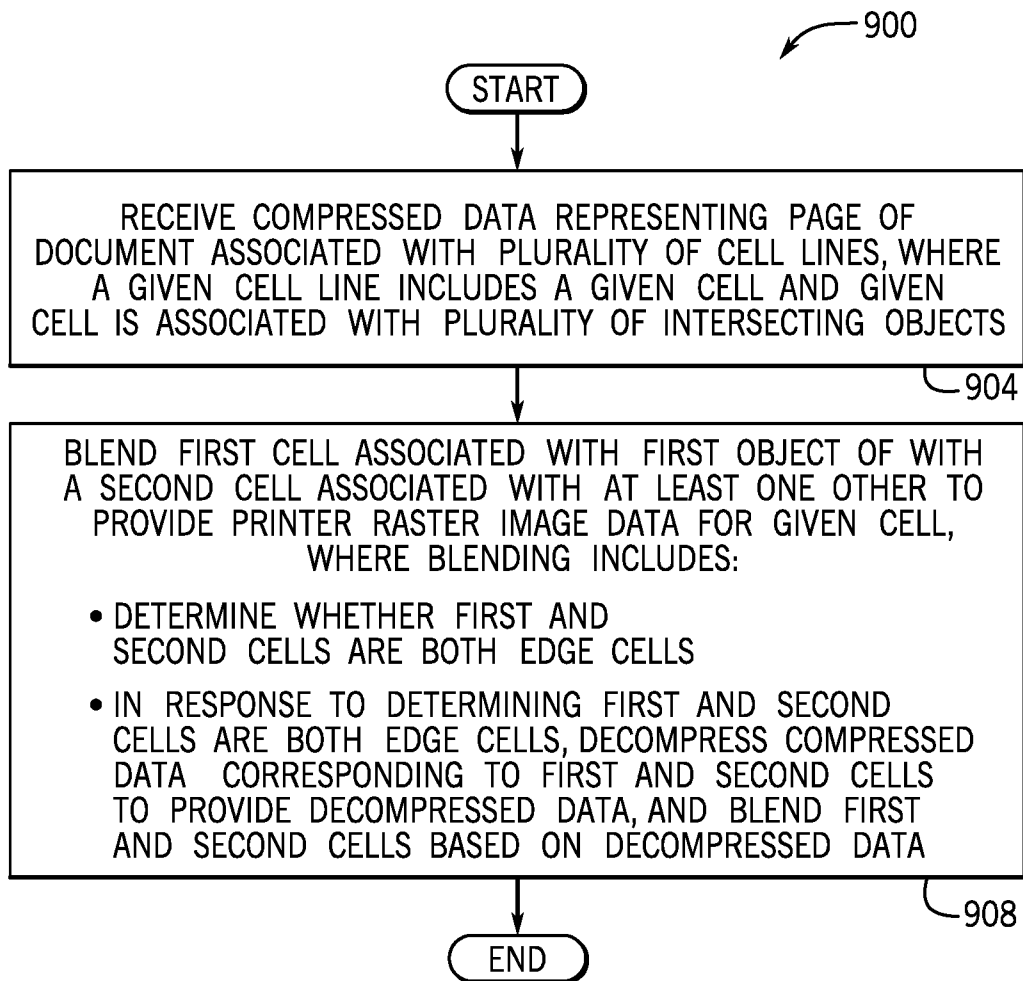


FIG. 9

12 / 13

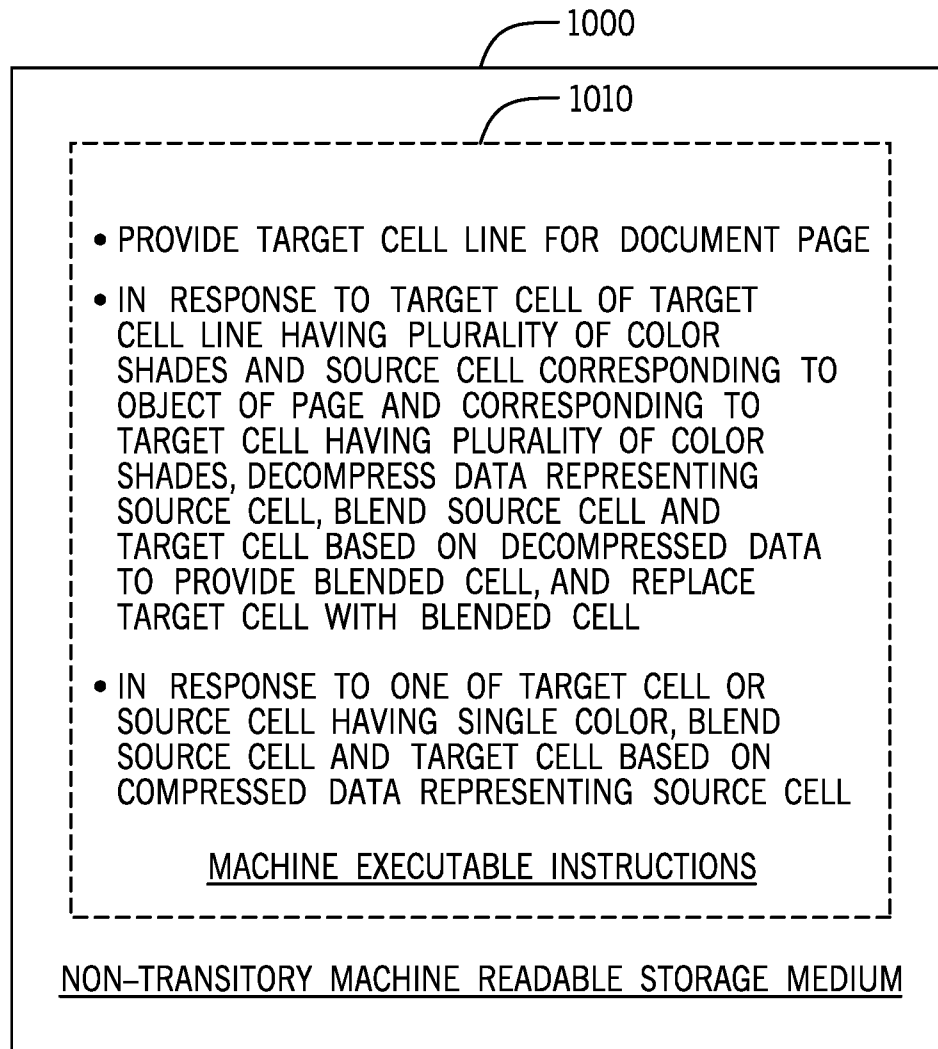


FIG. 10

13 / 13

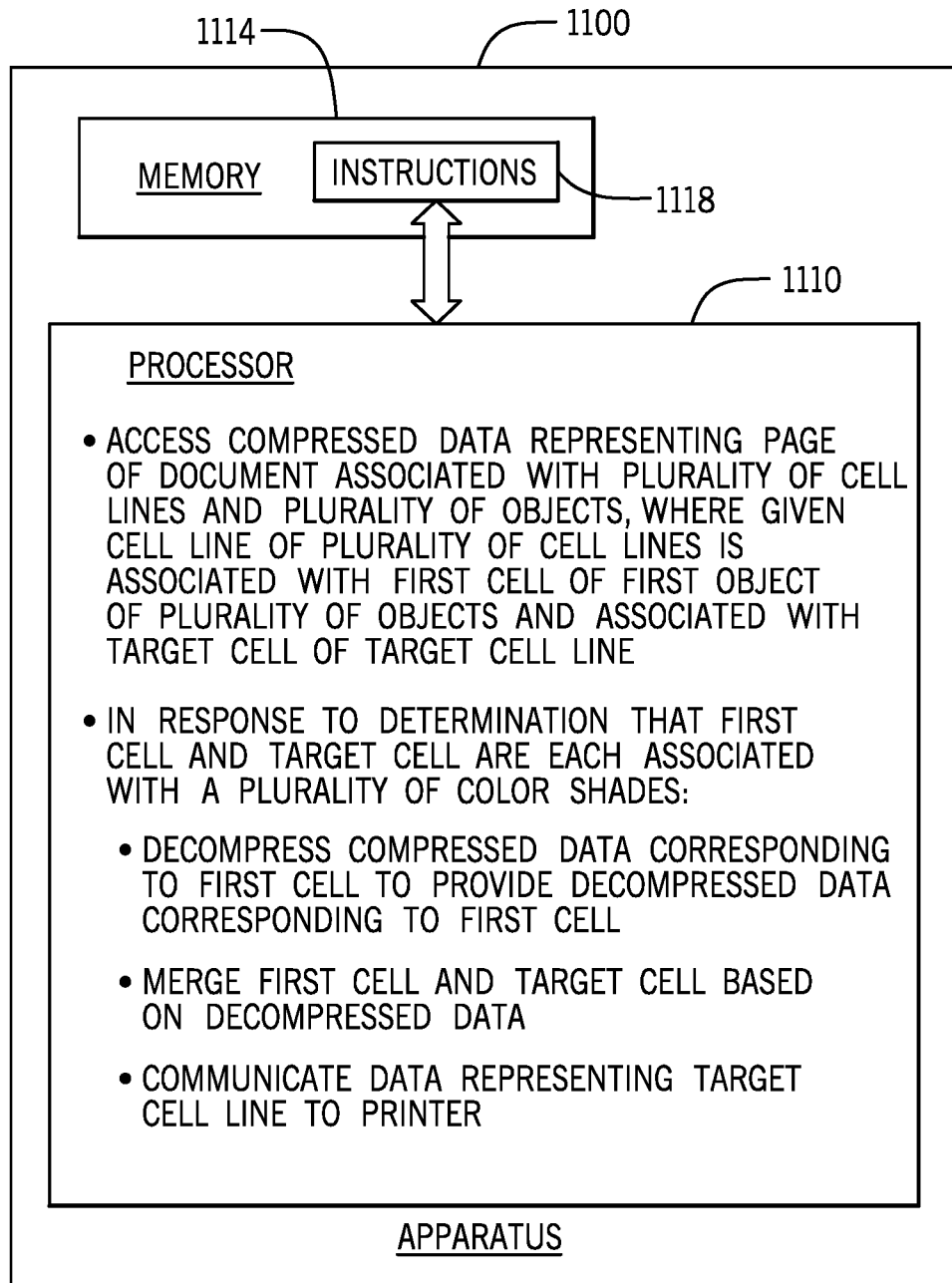


FIG. 11

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US 2017/062463

A. CLASSIFICATION OF SUBJECT MATTER

G06F 17/21 (2006.01)**H03M 7/30 (2006.01)**

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F 17/00- 17/21, H03M 7/00-7/30

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

PatSearch (RUPTO internal), USPTO, PAJ, Esp@cenet, DWPI, EAPATIS, PATENTSCOPE, Information Retrieval System of FIPS

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2006/0248454 A1 (FABIA GIANETTI) 02.11.2006	1-15
A	US 2007/0055929 A1 (HEWLETT PACKARD DEVELOPMENT COMPANY) 08.03.2007	1-15
A	US 2006/0023238 A1 (EASTMAN KODAK COMPANY) 02.02.2006	1-15
A	US 2008/0285074 A1 (GLOBAL GRAPHICS SOFTWARE LIMITED) 20.11.2008	1-15



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

“A” document defining the general state of the art which is not considered to be of particular relevance

“E” earlier document but published on or after the international filing date

“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

“O” document referring to an oral disclosure, use, exhibition or other means

“P” document published prior to the international filing date but later than the priority date claimed

“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

“&” document member of the same patent family

Date of the actual completion of the international search

27 June 2018 (27.06.2018)

Date of mailing of the international search report

02 August 2018 (02.08.2018)

Name and mailing address of the ISA/RU:
Federal Institute of Industrial Property,
Berezhkovskaya nab., 30-1, Moscow, G-59,
GSP-3, Russia, 125993
Facsimile No: (8-495) 531-63-18, (8-499) 243-33-37

Authorized officer

K. Afanasiev

Telephone No. (499) 240-25-91