process, this step is not necessarily needed. That is, while the electronic image data can be generated by scanning an original image, or the like, the electronic image data could have been generated at any time in the past. Moreover, the electronic image data need not have been generated from an original physical image, but could have been created from scratch electronically. Accordingly, if electronic image data of the image is already available to the image source, step **S1100** can be skipped, with control continuing directly from step **S1000** to step **S1200**.

In step **S1300**, image data is segmented into regions of different data types, including segments of at least one type of blocked data and segments of at least one type of non-blocked data, or segments of a plurality of types of blocked image data, as described above. Then, in step **S1400**, compressed image data is generated from the image segments. Next, in step **S1500**, the compressed image data is transmitted or possibly stored before being transmitted, to a device for decompressing the compressed image data.

In step **S1600**, the compressed image data is decompressed. Subsequently, in step **S1700**, the data is pasted into corresponding positions. Next, in step **S1800**, the image data is output. Then, in step **S1900** the control routine ends.

FIG. **6** outlines in greater detail the compression process of step **S1400**. Beginning in step **S1405**, the first segment is selected as the current segment. Then in step **S1410**, the control routine determines if the current segment is a text segment i.e., a non-blocked-data segment. If not, control jumps to step **S1420**. Otherwise, if the segment is a non-blocked-data segment, control continues to step **S1415**.

In step **S1415**, the non-blocked-data segment is compressed using the selected non-blocked-data compression technique, such as LZW for text data. Control then jumps to step **S1475**.

In step **S1420**, the control routine determines whether the current segment is a picture segment i.e., a blocked-data segment. If not, the segment contains data other than text or picture and control continues to step **S1425**. Otherwise, if the segment is a blocked-data segment, control jumps to step **S1430**.

In step **S1425**, the non-text and non-picture segment is compressed using other appropriate compression techniques, which themselves may be blocked-data compression techniques that use the recursive dilation technique described herein. Control then jumps to step **S1475**.

In step **S1430**, the blocked-data segment containing picture data is divided into picture blocks. Then, in step **S1435**, the first picture block is selected as the current block.

In step **S1440**, the control routine determines whether the current block contains missing data i.e., it is a boundary block. If the current block contains missing data, control continues to step **S1445**. Otherwise, if the current block not is missing any data, control jumps to step **S1465**.

In step **S1445**, the next pixel of the current block is input as the current pixel. Then, in step **S1450**, the control routine determines whether the current pixel is a picture pixel. If not, control continues to step **S1455**. Otherwise, if the pixel is a picture pixel, control jumps to step **S1460**.

In step **S1455**, the value of the closest boundary pixel to the current pixel is assigned to the current pixel. Control then continues to step **S1460**.

In step **S1460**, the control routine determines whether there are any more pixels. If so, control returns to step **S1445**. Otherwise, control continues to step **S1465**.

In step **S1465**, the picture block is compressed using the selected picture data compression technique, such as JPEG. Then in step **S1470**, the control routine determines whether there are any more blocks to be compressed. If so, control returns to **S1435**. Otherwise, if all of the blocks have been compressed, control continues to step **S1475**.

In step **S1475**, the control routine determines if there are any more data segments to be compressed. If so, control jumps back to step **S1405**. Otherwise, if all the data segments have been compressed, control continues to step **S1480**. In step **S1480**, control returns to step **S1500**.

It should be appreciated that, if additional compression techniques for compressing additional types of data are also implemented, the data type testing in steps **S1410** and **S1420** will be expanded to test for segments of these types. Thus, the boundary block testing and missing data filling steps **S1440–S1460**, and the compressing in steps **S1415**, **1425** and **S1465** will also be expanded or replicated to deal with these additional data types.

FIG. **7** outlines in greater detail the decompression and pasting processes of steps **S1600** and **S1700**. Beginning in step **S1610**, the first segment is selected as the current segment. Then, in step **S1620**, the control routine determines if the current segment is a text segment i.e., a non-blocked-data segment. If not, control jumps to step **S1630**. Otherwise, if the segment is a text segment, control continues to step **S1625**.

In step **S1625**, the text segment is decompressed using the selected text data decompression technique, such as LZW. Control then jumps to step **S1650**.

In step **S1630**, the control routine determines if the segment is a picture segment i.e., a blocked-data segment. If so, control jumps to step **S1640**. Otherwise, if the segment contains data other than text or picture data, control continues to step **S1635**.

In step **S1635**, the non-text and non-picture data segment is decompressed using other appropriate decompression techniques. Control then jumps to step **S1650**.

In step **S1640**, the data is decompressed using the selected picture data decompression technique, such as JPEG. Control then continues to step **S1645**.

In step **S1645**, the picture data is pasted into the appropriate locations. Control then continues to step **S1650**.

Then in step **S1650**, the control routine determines whether there are any more segments to be decompressed. If so, control returns to **S1610**. Otherwise, if all data have been decompressed, control continues to step **S1660**.

In step **S1660**, all unpasted segments, such as the text segments, are pasted into their appropriate locations. Then control continues to **S1670**.

In step **S1670**, control returns to step **S1800**.

FIG. **8** shows an example of the recursive dilation method of this invention for an exemplary 8×8 boundary block of an exemplary data type. The columns range from m=0 to m=7, and the rows range from n=0 to n=7. The vertical region boundary in FIG. **7** is located between columns **4** and **5**. The term "y" in the block indicates the brightness component. The data to the right of the region boundary is the missing data. According to the methods and systems of this invention, the recursive dilation method assigns the value of the closest boundary pixel to the pixel with missing data. Thus, for each row n, the value of the pixel (m−1,n), which is the picture pixel in the picture portion adjacent to the boundary, is assigned to the missing value at each pixel (m,n) for m>4. Thus, for the row "0", each missing pixel (5,0), (6,0) and (7,0) is assigned the value of the pixel just to the picture side of the boundary, i.e., the pixel (4,0).