US009270900B2

(12) **United States Patent**
Kitashou

(10) **Patent No.: US 9,270,900 B2**
(45) **Date of Patent: Feb. 23, 2016**

(54) **MOVIE PROCESSING APPARATUS AND CONTROL METHOD THEREFOR**

(71) Applicant: **CANON KABUSHIKI KAISHA**, Tokyo (JP)

(72) Inventor: **Tetsurou Kitashou**, Tokyo (JP)

(73) Assignee: **CANON KABUSHIKI KAISHA**, Tokyo (JP)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 288 days.

(21) Appl. No.: **13/909,364**

(22) Filed: **Jun. 4, 2013**

(65) **Prior Publication Data**

US 2013/0329133 A1 Dec. 12, 2013

(30) **Foreign Application Priority Data**

Jun. 11, 2012 (JP) 2012-132370

(51) **Int. Cl.**

G06T 3/00	(2006.01)
H04N 5/262	(2006.01)
H04N 9/31	(2006.01)
G06T 1/60	(2006.01)

(52) **U.S. Cl.**

CPC ***H04N 5/2628*** (2013.01); ***G06T 1/60*** (2013.01); ***G06T 3/00*** (2013.01); ***H04N 9/3185*** (2013.01)

(58) **Field of Classification Search**

CPC . H04N 5/2628; H04N 9/3185; H04N 9/2335; G06T 3/00; G06T 1/60
USPC 353/69, 121; 348/580
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,367,933	B1 *	4/2002	Chen et al.	353/69
2008/0204779	A1	8/2008	Koyama	
2009/0066812	A1 *	3/2009	Nakaya	H04N 1/047 348/231.99
2011/0210987	A1	9/2011	Furui	
2012/0237123	A1 *	9/2012	Kasahara	H04N 19/63 382/166
2012/0287298	A1	11/2012	Kitashou	348/222.1

FOREIGN PATENT DOCUMENTS

JP	3394551	B	4/2003
WO	00/21282	A	4/2000

OTHER PUBLICATIONS

Mar. 21, 2014 European Search Report concerning European Patent Application No. 13171528.6.

* cited by examiner

Primary Examiner — William C Dowling
Assistant Examiner — Magda Cruz
(74) *Attorney, Agent, or Firm* — Fitzpatrick, Cella, Harper & Scinto

(57) **ABSTRACT**

A movie processing apparatus includes an input unit that inputs an input image. A memory control unit stores, in a memory, a deformation image obtained by executing an image deformation process for the input image using a deformation parameter. A determination unit determines, from a memory region in which the deformation image is stored, a readout region to be accessed for obtaining an output image, on the basis of a size of the input image, a size of the output image, and the deformation parameter. A readout unit configured to access the readout region reads out pixel data included in the deformation image. An output unit outputs the image based on the pixel data readout by the readout unit.

12 Claims, 7 Drawing Sheets

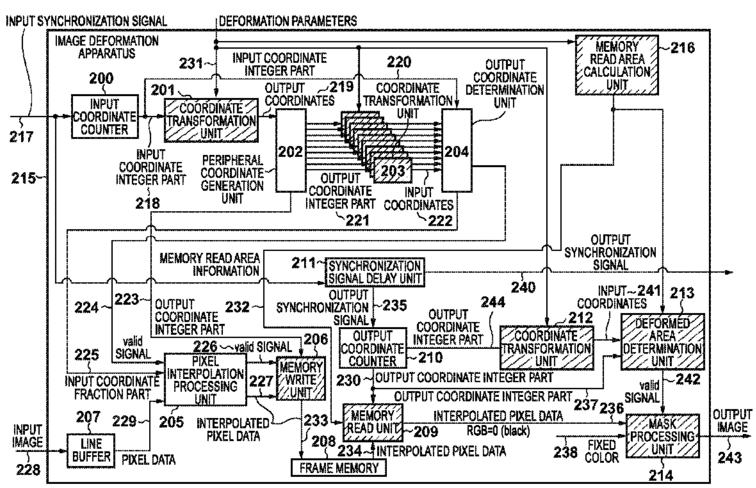


FIG. 1A

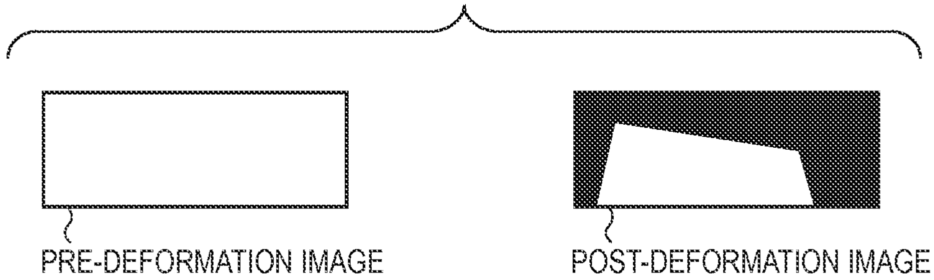


FIG. 1B

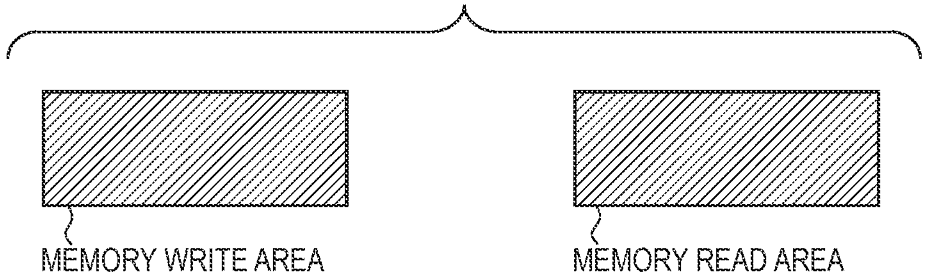


FIG. 1C

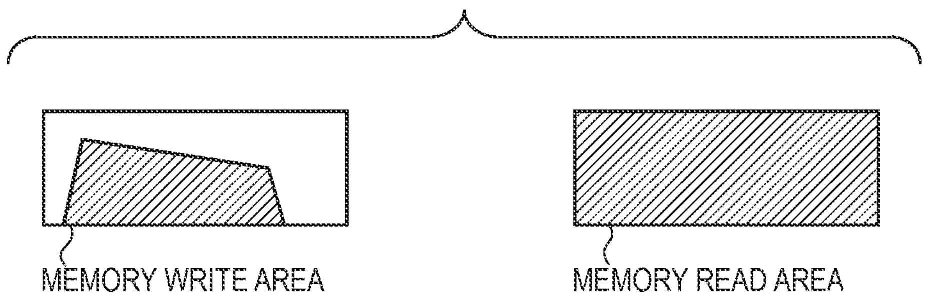


FIG. 1D

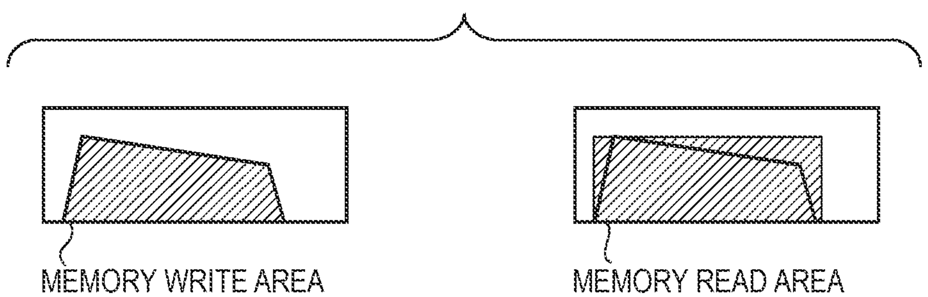
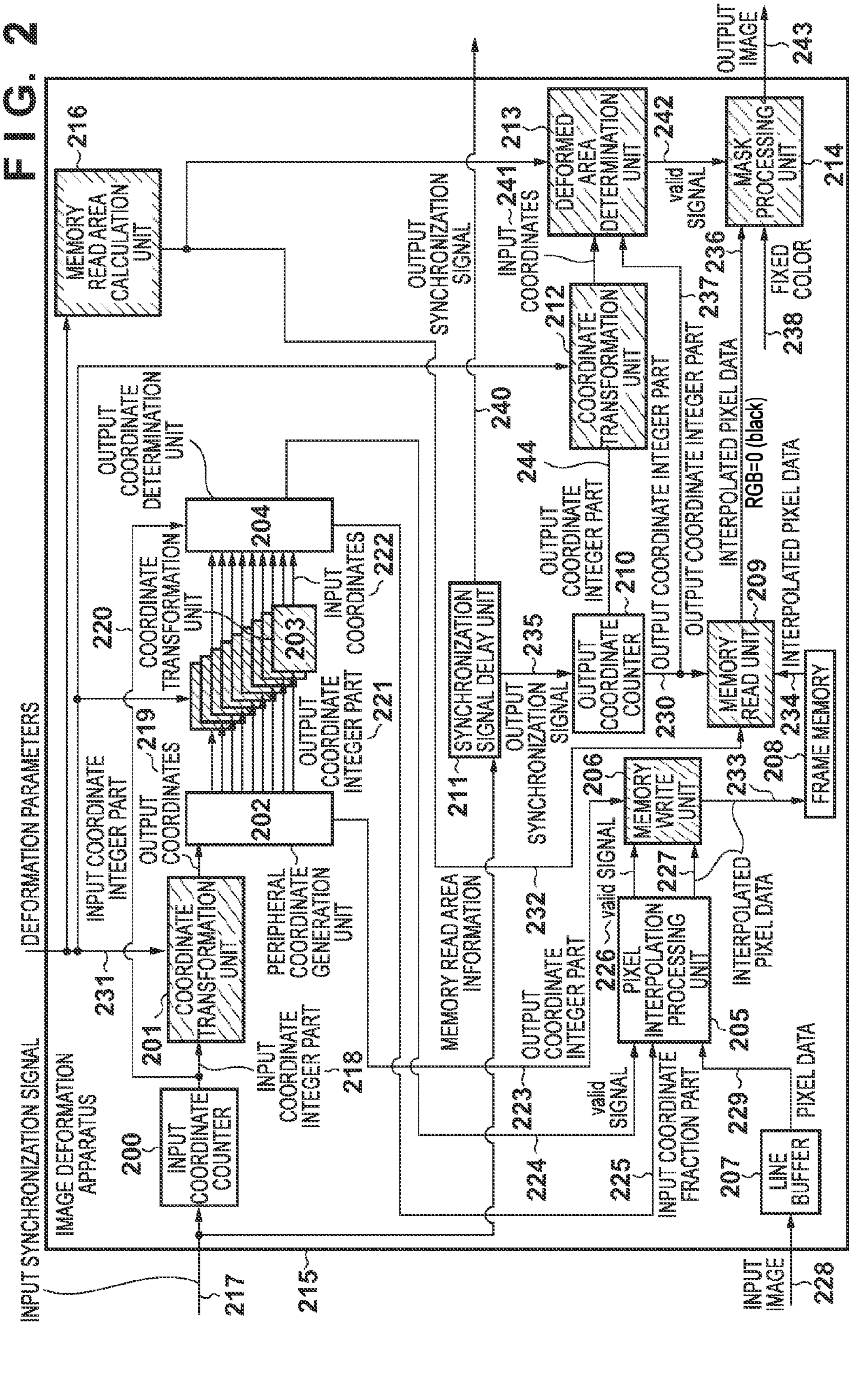


FIG. 2


INPUT SYNCHRONIZATION SIGNAL
217
215
IMAGE DEFORMATION APPARATUS
DEFORMATION PARAMETERS
231
200
INPUT COORDINATE COUNTER
201
COORDINATE TRANSFORMATION UNIT
INPUT COORDINATE INTEGER PART
218
INPUT COORDINATE INTEGER PART
OUTPUT COORDINATES
219
202
PERIPHERAL COORDINATE GENERATION UNIT
220
COORDINATE TRANSFORMATION UNIT
203
OUTPUT COORDINATE INTEGER PART
221
INPUT COORDINATES
222
204
OUTPUT COORDINATE DETERMINATION UNIT
216
MEMORY READ AREA CALCULATION UNIT
MEMORY READ AREA INFORMATION
223
224
225
valid SIGNAL
INPUT COORDINATE FRACTION PART
OUTPUT COORDINATE INTEGER PART
232
226
valid SIGNAL
PIXEL INTERPOLATION PROCESSING UNIT
205
227
INTERPOLATED PIXEL DATA
229
PIXEL DATA
207
LINE BUFFER
INPUT IMAGE
228
211
SYNCHRONIZATION SIGNAL DELAY UNIT
235
OUTPUT SYNCHRONIZATION SIGNAL
206
MEMORY WRITE UNIT
233
208
FRAME MEMORY
234
230
OUTPUT COORDINATE COUNTER
210
OUTPUT COORDINATE INTEGER PART
MEMORY READ UNIT
209
INTERPOLATED PIXEL DATA
OUTPUT COORDINATE INTEGER PART
244
212
COORDINATE TRANSFORMATION UNIT
240
OUTPUT SYNCHRONIZATION SIGNAL
INPUT COORDINATES
241
213
DEFORMED AREA DETERMINATION UNIT
242
valid SIGNAL
237
236
RGB=0 (black)
238
FIXED COLOR
MASK PROCESSING UNIT
214
OUTPUT IMAGE
243

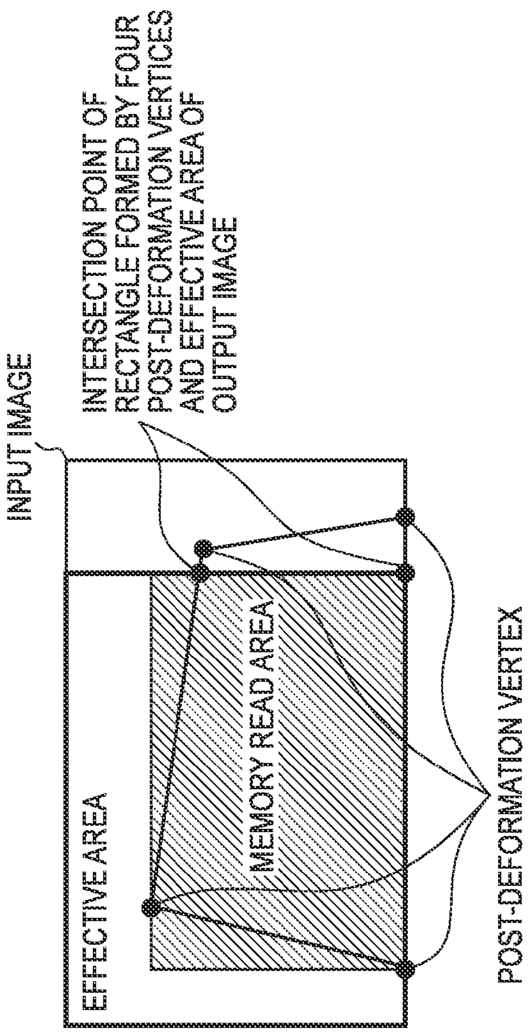


FIG. 3A

CASE IN WHICH INPUT IMAGE IS DIFFERENT FROM EFFECTIVE AREA

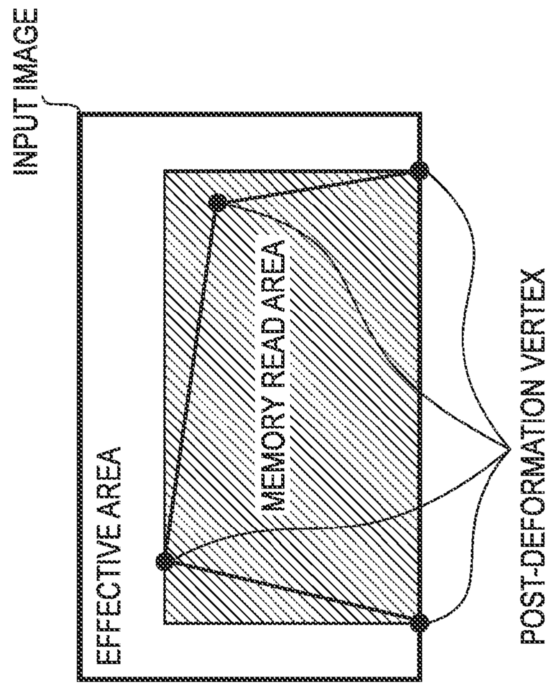


FIG. 3B

CASE IN WHICH INPUT IMAGE IS THE SAME AS EFFECTIVE AREA

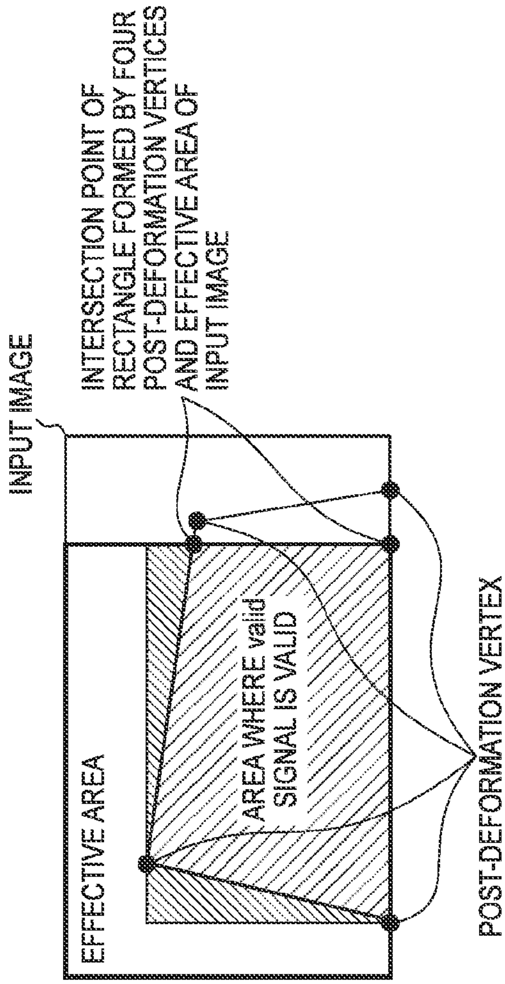


FIG. 4A

CASE IN WHICH INPUT IMAGE IS DIFFERENT FROM EFFECTIVE AREA

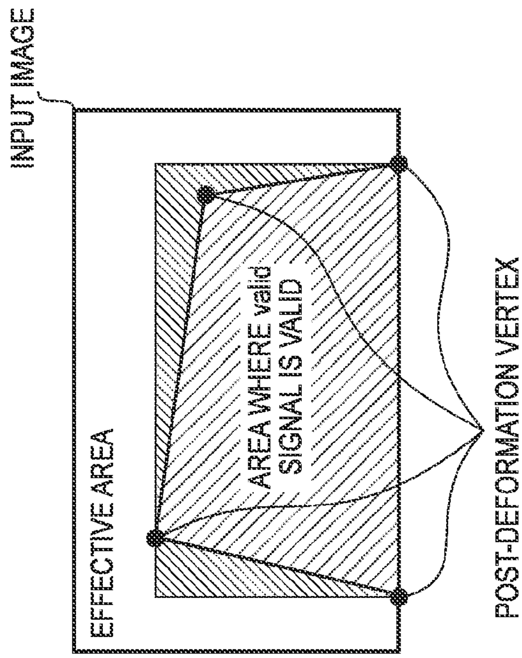


FIG. 4B

CASE IN WHICH INPUT IMAGE IS THE SAME AS EFFECTIVE AREA

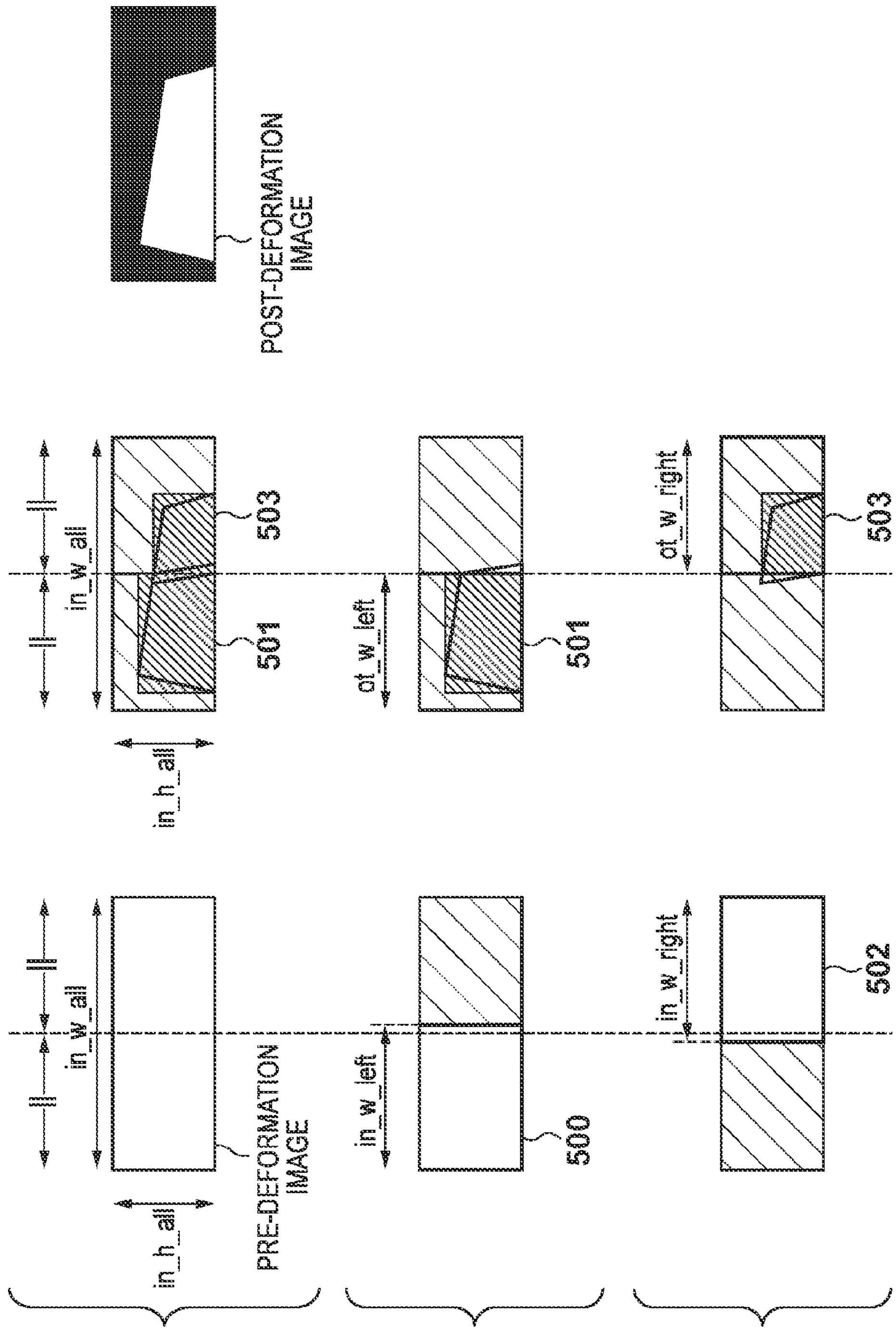


FIG. 5A

FIG. 5B

FIG. 5C

FIG. 6A

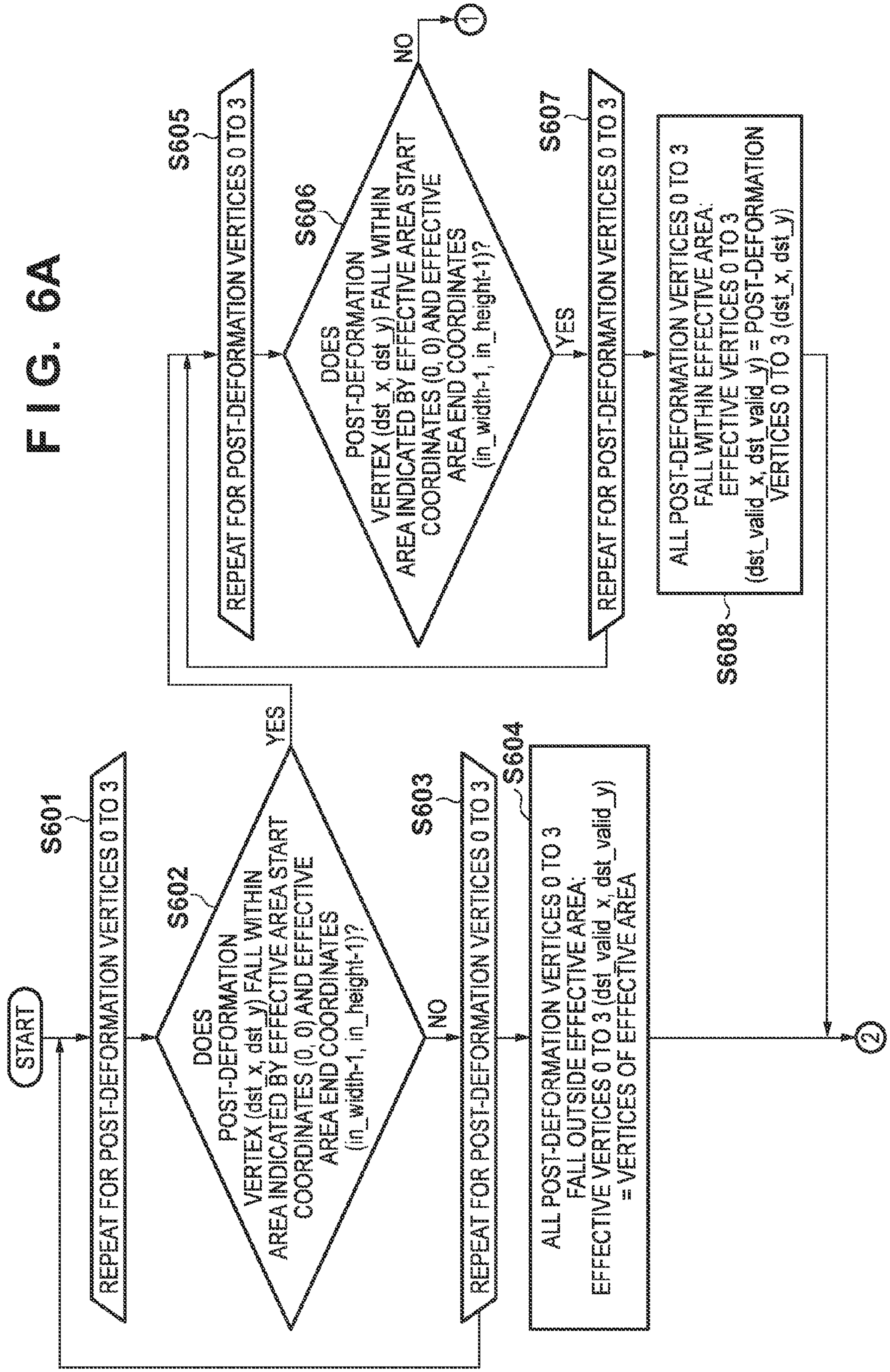
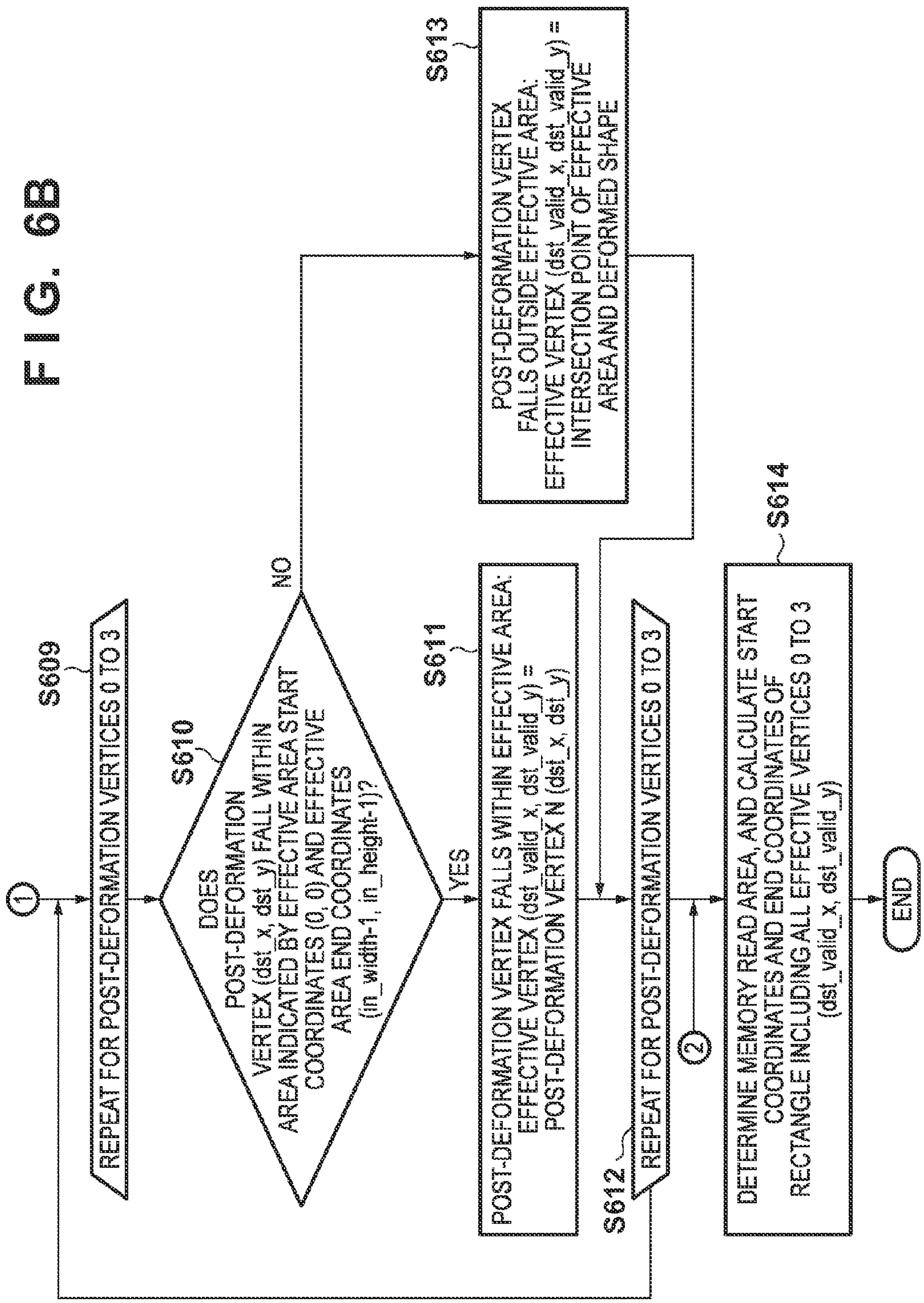

START
S601
REPEAT FOR POST-DEFORMATION VERTICES 0 TO 3
S602
DOES POST-DEFORMATION VERTEX (dst_x, dst_y) FALL WITHIN AREA INDICATED BY EFFECTIVE AREA START COORDINATES (0, 0) AND EFFECTIVE AREA END COORDINATES (in_width-1, in_height-1)?
YES
NO
S603
REPEAT FOR POST-DEFORMATION VERTICES 0 TO 3
S604
ALL POST-DEFORMATION VERTICES 0 TO 3 FALL OUTSIDE EFFECTIVE AREA: EFFECTIVE VERTICES 0 TO 3 (dst_valid_x, dst_valid_y) = VERTICES OF EFFECTIVE AREA
S605
REPEAT FOR POST-DEFORMATION VERTICES 0 TO 3
S606
DOES POST-DEFORMATION VERTEX (dst_x, dst_y) FALL WITHIN AREA INDICATED BY EFFECTIVE AREA START COORDINATES (0, 0) AND EFFECTIVE AREA END COORDINATES (in_width-1, in_height-1)?
NO
1
YES
S607
REPEAT FOR POST-DEFORMATION VERTICES 0 TO 3
S608
ALL POST-DEFORMATION VERTICES 0 TO 3 FALL WITHIN EFFECTIVE AREA: EFFECTIVE VERTICES 0 TO 3 (dst_valid_x, dst_valid_y) = POST-DEFORMATION VERTICES 0 TO 3 (dst_x, dst_y)
2

FIG. 6B


1
S609
REPEAT FOR POST-DEFORMATION VERTICES 0 TO 3
S610
DOES POST-DEFORMATION VERTEX (dst_x, dst_y) FALL WITHIN AREA INDICATED BY EFFECTIVE AREA START COORDINATES (0, 0) AND EFFECTIVE AREA END COORDINATES (in_width-1, in_height-1)?
YES
NO
S611
POST-DEFORMATION VERTEX FALLS WITHIN EFFECTIVE AREA: EFFECTIVE VERTEX (dst_valid_x, dst_valid_y) = POST-DEFORMATION VERTEX N (dst_x, dst_y)
S613
POST-DEFORMATION VERTEX FALLS OUTSIDE EFFECTIVE AREA: EFFECTIVE VERTEX (dst_valid_x, dst_valid_y) = INTERSECTION POINT OF EFFECTIVE AREA AND DEFORMED SHAPE
S612
REPEAT FOR POST-DEFORMATION VERTICES 0 TO 3
2
S614
DETERMINE MEMORY READ AREA, AND CALCULATE START COORDINATES AND END COORDINATES OF RECTANGLE INCLUDING ALL EFFECTIVE VERTICES 0 TO 3 (dst_valid_x, dst_valid_y)
END

MOVIE PROCESSING APPARATUS AND CONTROL METHOD THEREFOR

CLAIM OF PRIORITY

This application claims the benefit of Japanese Patent Application No. 2012-132370 filed Jun. 11, 2012, which is hereby incorporated by reference herein in its entirety.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to a movie processing technique for executing an image deformation process.

2. Description of the Related Art

As an application of a projector, or the like, for example, a geometric process is executed. A projector requires this geometric process to correct distortion on a projection plane, which occurs depending on its installation angle, and is generally known as a keystone correction process or a trapezoid distortion correction process. There are mainly two methods for implementing the above geometric process. One of them will be referred to as a memory read deformation method, and the other will be referred to as a memory write deformation method hereafter.

A process by the memory read deformation method is characterized by execution of an image deformation process when reading out data from a frame memory. More specifically, after storing one screen of an input image in the frame memory, weighting of a pixel near input image coordinates corresponding to each set of coordinates of an output image is calculated. As a weighting calculation method, for example, a method such as the bicubic method is generally known. After weighting, an image deformation process is completed by reading out a pixel near input image coordinates corresponding to each set of coordinates of the output image from the frame memory, performing a convolution operation in consideration of the above weighting, and generating each pixel of the output image.

On the other hand, a process by the memory write deformation method is characterized by execution of an image deformation process when writing data in the frame memory. More specifically, a pixel position of an output image corresponding to each pixel of an input image is calculated, and weighting of a pixel near each set of input image coordinates is calculated. As a weighting calculation method, the bicubic method, or the like, can be used, similar to the memory read deformation method. A convolution operation is performed for a pixel near input image coordinates corresponding to each pixel position of the output image in consideration of the above weighting, and each pixel of the output image is generated and written in the frame memory. After that, a deformed output image is obtained by sequentially reading out and outputting the pixels from the frame memory. A more practical process example is disclosed in, for example, Japanese Patent No. 3394551.

The image deformation process requires frequent access to the frame memory capable of storing an image of one screen, because the scan order of an input image is different from that of an output image. The frame memory is usually implemented by a dynamic RAM on an integrated circuit. Along with a recent increase in resolution and frame rate of an image, the number of memory access operations increases, thereby raising the power consumption.

The memory read deformation method and the memory write deformation method described above will be compared with respect to a memory access area. FIG. **1**A shows examples of images before and after deformation, in which a pre-deformation image is shown on the left side and a post-deformation image is shown on the right side. In FIG. **1**A, a rectangular image represents a pre-deformation image, and an image obtained by deforming the rectangular image into a trapezoidal image represents a post-deformation image.

FIG. **1**B shows a memory access area in the memory read deformation method in this case. A hatched portion represents a memory access area. As described above, since the memory read deformation method stores one screen of an input image in the frame memory, and then executes a deformation process by reading out the entire area, memory access for two screens occurs.

On the other hand, FIG. **1**C shows a memory access area in the memory write deformation method. A hatched portion represents a memory access area. As described above, since the memory write deformation method deforms an input image before storing it, and sequentially reads out and outputs pixels from the frame memory, memory access for fewer than two screens occurs. More specifically, memory write of a trapezoidal shape and memory read of a rectangular shape of one screen occur.

In either case, however, memory access for about two screens is necessary. To keep up with a recent increase in resolution and frame rate of an image, the power consumption rises due to an increase in number of memory access operations.

SUMMARY OF THE INVENTION

The present invention has been made in consideration of the above problem, and provides a technique for decreasing the number of access operations to a memory used for an image deformation process.

According to the first aspect of the present invention, a movie processing apparatus comprises a unit that stores, in a memory, an output image obtained by executing an image deformation process for an input image, a calculation unit that specifies, in the output image in the memory, a partial image within an effective area set in the memory, and calculates, as a readout area, an area including the specified partial image, and a unit that reads out the partial image by executing a readout process for the readout area in the memory.

According to the second aspect of the present invention, a control method for a movie processing apparatus comprises a step of storing, in a memory, an output image obtained by executing an image deformation process for an input image, a calculation step of specifying, in the output image in the memory, a partial image within an effective area set in the memory, and calculating, as a readout area, an area including the specified partial image, and a step of reading out the partial image by executing a readout process for the readout area in the memory.

Further features of the present invention will become apparent from the following description of exemplary embodiments with reference to the attached drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIGS. **1**A to **1**D are views for explaining memory access for a deformed image;

FIG. **2** is a block diagram showing an example of the arrangement of an image deformation apparatus **215**;

FIGS. **3**A and **3**B are views for explaining a memory read area;

FIGS. **4**A and **4**B are views each showing an area where a valid signal **242** is valid;

FIGS. **5**A to **5**C are views for explaining a case in which an input image **228** is divided and processed; and

FIGS. **6**A and **6**B are flowcharts each illustrating processing executed by a memory read area calculation unit **216**.

DESCRIPTION OF THE EMBODIMENTS

Embodiments of the present invention will be described below with reference to the accompanying drawings. Note that the embodiments to be explained below provide examples when the present invention is practically implemented, and are practical examples of an arrangement defined in the appended claims.

First Embodiment

An example of the arrangement of an image deformation apparatus **215** as a movie processing apparatus according to the embodiment will be described with reference to a block diagram shown in FIG. **2**. Note that the arrangement shown in FIG. **2** is an example of an arrangement for implementing each process to be explained below. Another arrangement can be adopted as long as it can implement a process similar to that to be described below. Although each function unit shown in FIG. **2** is formed by hardware in the following description, some or all of the units (except for memories) may be implemented by computer programs. In this case, when a computer program is stored in advance in an appropriate memory within the image deformation apparatus **215**, and the control unit of the image deformation apparatus **215** reads out and executes the computer program, it is possible to implement the function of a corresponding function unit.

The image deformation apparatus **215** receives an input synchronization signal **217**, an input image **228**, and deformation parameters **231**, and deforms the input image **228** into a shape designated by the deformation parameters **231**, thereby outputting an output synchronization signal **240** and an output image **243**.

The input synchronization signal **217** includes a vertical synchronization signal, a horizontal synchronization signal, and a pixel clock for indicating an effective area of one screen of the input image **228**. The deformation parameters **231** are used to designate a deformed shape. The deformation parameters **231** include matrices S2D_MTX and D2S_MTX, the width in_width and height in_height of the input image, and the width ot_width and height ot_height of the output image. Furthermore, the deformation parameters **231** include the effective area start coordinates and effective area end coordinates of the output image **243**. These parameters will be shown below. Note that the matrix S2D_MTX is a matrix for transforming input coordinates into output coordinates. The matrix D2S_MTX is the inverse matrix of the matrix S2D_MTX, that is, a matrix for transforming output coordinates into input coordinates.

$$\text{S2D\_MTX} = \begin{bmatrix} m00 & m01 & m02 \\ m10 & m11 & m12 \\ m20 & m21 & m22 \end{bmatrix}$$

$$\text{D2S\_MTX} = \text{S2D\_MTX}^{-1}$$

input image width: in_width, input image height: in_height output image width: *ot*_width, output image height: *ot*_height effective area start coordinates of output image: (*ot*_*bgn*_*x*, *ot*_*bgn*_*y*) effective area end coordinates of output image: (*ot*_end_*x*, *ot*_end_*y*) (1)

In this embodiment, for descriptive convenience, assume that the vertical and horizontal sizes of the input image **228** are respectively equal to those of the output image **243**, and the entire area of the output image **243** is an effective area. This example is not a special case, but a general case for deformation of an image of one screen. Details of this example are as follows:

input image width: in_width, input image height: in_height output image width: in_width, output image height: in_height effective area start coordinates of output image: (0, 0) effective area end coordinates of output image: (in_width-1, in_height-1) (2)

In this case, the coordinates of four input coordinate vertices, that is, the coordinates of the four corner positions of the input image are as follows:

input coordinate vertex 0: (*src*_*x*, *src*_*y*)=(0, 0) input coordinate vertex 1: (*src*_*x*, *src*_*y*)=(in_width-1, 0) input coordinate vertex 2: (*src*_*x*, *src*_*y*)=(in_width-1, in_height-1) input coordinate vertex 3: (*src*_*x*, *src*_*y*)=(0, in_height-1) (3)

The input synchronization signal **217** is input to an input coordinate counter **200**, which counts coordinates according to the pixel clock included in the input synchronization signal **217** and outputs the counted coordinates as a coordinate value (input coordinates) on the input image **228**. More specifically, the input coordinate counter **200** outputs an input coordinate integer part **218** (src_x, src_y). Furthermore, each pixel of the input image **228** is input to a line buffer **207** in synchronism with the pixel clock, vertical synchronization signal, and horizontal synchronization signal.

A coordinate transformation unit **201** receives the input coordinate integer part **218** (src_x, src_y) and deformation parameters **231**, and calculates the following equations using the received values, thereby calculating output coordinates **219** (dst_x, dst_y).

$$\text{SRC\_POINT} = \begin{bmatrix} \text{src\_x} \\ \text{src\_y} \\ 1 \end{bmatrix} \tag{4}$$

$$\text{DST\_POINT} = \begin{bmatrix} \text{dst\_x} \\ \text{dst\_y} \\ 1 \end{bmatrix}$$

$$\text{DST\_POINT} = \frac{\text{S2D\_MTX} \cdot \text{SRC\_POINT}}{(\text{S2D\_MTX} \cdot \text{SRC\_POINT})[2]}$$

where (S2D_MTX□SRC_POINT)[2] represents a scalar component in the third row of a 3×1 matrix as a calculation result of (S2D_MTX·SRC_PONT).

A peripheral coordinate generation unit **202** receives the output coordinates **219**, and outputs, as an output coordinate integer part **221**, the coordinates of each of nine pixels including a pixel having the integer part of the output coordinates **219** as coordinates and eight pixels (each having integer coordinates) adjacent to the pixel.

A coordinate transformation unit **203** calculates corresponding input coordinates **222** based on the output coordinate integer part **221** of each of the nine pixels by:

$$\text{SRC\_POINT} = \frac{\text{D2S\_MTX} \cdot \text{DST\_POINT}}{(\text{D2S\_MTX} \cdot \text{DST\_POINT})[2]} \tag{5}$$

where the above deformation parameters **231** are used.

An output coordinate determination unit **204** receives an input coordinate integer part **220** output from the input coordinate counter **200**, and nine sets of input coordinates **222** output from the coordinate transformation unit **203**, and outputs a valid signal **224** and a fraction part **225** of each set of input coordinates **222**.

The valid signal **224** is information indicating whether each set of input coordinates **222** is valid. If each set of input coordinates **222** is effective, a post-deformation pixel is generated based on each set of input coordinates **222**. If the input coordinate integer part **220** coincides with the integer part of each set of input coordinates **222**, the valid signal **224** is valid. The fraction part **225** of the input coordinates **222** is formed by a fraction part obtained by omitting the integer part from the input coordinates **222**.

On the other hand, the input image **228** input to the image deformation apparatus **215** is input to the line buffer **207**, which holds a plurality of lines of the input image **228** and then outputs pixel data **229** including an input pixel and its neighboring pixels. In this embodiment, assume that interpolation is performed by the bicubic method, and thus the pixel data **229** of 4×4=16 pixels is output.

The valid signal **224** and input coordinate fraction part **225** output from the output coordinate determination unit **204**, and the pixel data **229** output from the line buffer **207** are input to a pixel interpolation processing unit **205**, which then outputs a valid signal **226** and interpolated pixel data **227**.

The pixel data **229** is data of 4×4=16 pixels, as described above. Interpolated pixel data **227** is data generated by interpolating the pixel data **229** by the bicubic method.

The valid signal **226** is obtained by holding the valid signal **224** for a period of time during which the pixel interpolation processing unit **205** generates the interpolated pixel data **227**, and is a set of valid signals **224** for the respective pixels of the interpolated pixel data **227**.

The valid signal **226** and interpolated pixel data **227** output from the pixel interpolation processing unit **205**, and an output coordinate integer part **223** output from the peripheral coordinate generation unit **202** are input to a memory write unit **206**.

The memory write unit **206** writes as interpolated pixel data **233** only data of a pixel, of the interpolated pixel data **227**, for which a corresponding valid signal is valid, at an address on a frame memory **208** designated by the output coordinate integer part **223** corresponding to the pixel.

When the above processing is executed for all of the pixel data of the input image **228**, the post-deformation image is stored in the frame memory **208**.

Processing after a readout process from the frame memory **208** will be described next.

The deformation parameters **231** input to the image deformation apparatus **215** are also input to a memory read area calculation unit **216**, which then calculates memory read area information **232** using the deformation parameters **231**, and outputs it. In this embodiment, a case in which the entire area of the output image is an effective area is exemplified. Therefore, the deformation parameters **231** are represented by:

$$\text{S2D\_MTX} = \begin{bmatrix} m00 & m01 & m02 \\ m10 & m11 & m12 \\ m20 & m21 & m22 \end{bmatrix}$$

$$\text{D2S\_MTX} = \text{S2D\_MTX}^{-1}$$

input image width: in_width, input image height: in_height output image width: in_width, output image height: in_height effective area start coordinates of output image: (0, 0) effective area end coordinates of output image: (in_width-1, in_height-1) (6)

In this case, using the four input coordinate vertices at this time, equations for calculating the four post-deformation vertices are represented as follows. More specifically, input coordinate vertices 0 to 3 are substituted into SRC_POINT to perform calculation according to the following equations, thereby setting, as post-deformation vertices, coordinates obtained as DST POINT.

input coordinate vertex 0: (*src_x, src_y*)=(0, 0) input coordinate vertex 1: (*src_x, src_y*)=(in_width-1, 0) input coordinate vertex 2: (*src_x, src_y*)=(in_width-1, in_height-1) input coordinate vertex 3: (*src_x, src_y*)=(0, in_height-1)

$$\text{SRC\_POINT} = \begin{bmatrix} \text{src\_x} \\ \text{src\_y} \\ 1 \end{bmatrix} \tag{7}$$

$$\text{DST\_POINT} = \begin{bmatrix} \text{dst\_x} \\ \text{dst\_y} \\ 1 \end{bmatrix}$$

$$\text{DST\_POINT} = \frac{\text{S2D\_MTX} \cdot \text{SRC\_POINT}}{(\text{S2D\_MTX} \cdot \text{SRC\_POINT})[2]}$$

Based on the calculated four post-deformation vertices, the memory read area information **232** is calculated. The memory read area information **232** is information indicating an area including the post-deformation vertices existing within the effective area of the output image **243**, and intersection points of a rectangle formed by the four post-deformation vertices and the effective area of the output image. The relationship is shown in FIGS. **3**A and **3**B.

FIG. **3**A shows a case in which the size of the input image is different from that of the effective area. On the other hand, FIG. **3**B shows a case in which the size of the input image is equal to that of the effective area. In this embodiment, the case shown in FIG. **3**B is applied. Note that a description of FIG. **3**A will be provided later as the second embodiment.

Processing executed by the memory read area calculation unit **216** to obtain the memory read area information **232** will be described with reference to FIGS. **6**A and **6**B, each of which is a flowchart illustrating the processing. Assume that when the processing according to the flowcharts shown in FIGS. **6**A and **6**B starts, the coordinates of the four post-deformation vertices have been calculated.

Processing in steps S**601** to S**603** is performed for each of post-deformation vertices 0 to 3. If, however, the memory read area calculation unit **216** determines in step S**602** that the coordinates (dst_x, dst_y) of a post-deformation vertex X (0≤X≤3) fall within the effective area of the output image, the process advances to step S**605**. If it is determined that the coordinates (dst_x, dst_y) of each of all post-deformation vertices 0 to 3 fall outside the effective area of the output image, the process advances to step S**604** through step S**603**.

In step S**604**, since all post-deformation vertices 0 to 3 fall outside the effective area, the memory read area calculation unit **216** sets the four vertices of the effective area as effective vertices 0 to 3 indicating the vertices of the deformed shape within the effective area.

Processing in steps S**605** to S**607** is executed for each of post-deformation vertices 0 to 3. If, however, the memory read area calculation unit **216** determines in step S**606** that the coordinates (dst_x, dst_y) of the post-deformation vertex X (0≤X≤3) fall outside the effective area of the output image, the process advances to step S**609**. Alternatively, if it is determined that the coordinates (dst_x, dst_y) of each of all post-deformation vertices 0 to 3 fall within the effective area of the output image, the process advances to step S608 through step S607.

In step S608, since all of the post-deformation vertices 0 to 3 fall within the effective area, the memory read area calculation unit 216 sets post-deformation vertices 0 to 3 as effective vertices 0 to 3 indicating the vertices of the deformed shape within the effective area.

Processing in steps S609 to S612 is executed for each of post-deformation vertices 0 to 3. If, however, the memory read area calculation unit 216 determines in step S610 that the coordinates (dst_x, dst_y) of the post-deformation vertex X ($0 \le X \le 3$) fall within the effective area of the output image, the process advances to step S611. On the other hand, if it is determined that the coordinates (dst_x, dst_y) of the post-deformation vertex X ($0 \le X \le 3$) fall outside of the effective area of the output image, the process advances to step S613.

In step S611, the memory read area calculation unit 216 sets the post-deformation vertex X as an effective vertex X indicating a vertex of the deformed shape within the effective area. On the other hand, in step S613, the memory read area calculation unit 216 sets, as a target side, one of sides, among the sides of the post-deformation area formed by the four post-deformation vertices, which have the post-deformation vertex X as one end point and intersect the effective area. The unit 216 then calculates the intersection point of the target side and the effective area, and sets it as the effective vertex X indicating the vertex of the deformed shape within the effective area.

In step S614, the memory read area calculation unit 216 calculates a rectangular area including all calculated effective vertices 0 to 3, and then calculates the coordinates of diagonal vertices (for example, the coordinates of the vertex at the upper left corner and the vertex at the lower right corner) of the rectangular area as start coordinates and end coordinates, respectively. This rectangular area will be referred to as a memory read area. The memory read area calculation unit 216 outputs the memory read area information 232 including the start and end coordinates of the memory read area. Note that the area represented by the memory read area information 232 is indicated by a hatched portion shown in FIG. 3A or 3B.

A synchronization signal delay unit 211 receives an input synchronization signal 217, and delays it by a given time, thereby outputting it as an output synchronization signal 235. This processing is executed to control the timing of writing data in the frame memory 208 and reading out data from the frame memory 208 not to cause the readout operation to pass the write operation.

An output coordinate counter 210 receives the output synchronization signal 235, and counts coordinates according to a clock included in the output synchronization signal 235, thereby outputting the counted coordinates as a coordinate value (output coordinates) on the output image 243. More specifically, the counter 210 outputs the output coordinate integer parts 230, 237, and 244.

A memory read unit 209 receives the output coordinate integer part 230 and the memory read area information 232, and reads out interpolated pixel data 234 from the frame memory 208, thereby outputting it to the succeeding stage. More specifically, if the output coordinate integer part 230 represents coordinates within the area indicated by the memory read area information 232, the unit 209 reads out, as the interpolated pixel data 234, a pixel stored at an address corresponding to the coordinates in the frame memory 208. A practical example of the memory read area information 232 is shown in FIG. 3A or 3B.

A coordinate transformation unit 212 receives the deformation parameters 231 and the output coordinate integer part 244, and transforms the output coordinate integer part 244 into input coordinates 241 using the deformation parameters 231, thereby outputting them. This transformation is represented by:

$$\text{SRC\_POINT} = \begin{bmatrix} \text{src\_x} \\ \text{src\_y} \\ 1 \end{bmatrix} \quad (8)$$

$$\text{DST\_POINT} = \begin{bmatrix} \text{dst\_x} \\ \text{dst\_y} \\ 1 \end{bmatrix}$$

$$\text{SRC\_POINT} = \frac{\text{D2S\_MTX} \cdot \text{DST\_POINT}}{(\text{D2S\_MTX} \cdot \text{DST\_POINT})[2]}$$

A deformed area determination unit 213 receives the input coordinates 241, output coordinate integer part 237, and memory read area information 232, and outputs a valid signal 242. When the input coordinates 241 fall within the area of the output image 243 and within the memory read area, the valid signal 242 is valid. FIG. 4A or 4B shows an area where the valid signal 242 is valid.

A mask processing unit 214 receives the valid signal 242, interpolated pixel data 236, and a fixed color 238, and outputs the output image 243. More specifically, the unit 214 refers to the corresponding valid signal 242 for each pixel. If the valid signal 242 referred to is valid, the unit 214 selects and outputs the interpolated pixel data 236; otherwise, the unit 214 selects and outputs the fixed color 238. The fixed color 238 is a color output to an area outside the deformed area. For example, black is output in keystone correction by a projector.

Note that, as described above, the arrangement explained in this embodiment is merely an example, and only provides an example of the following arrangement. That is, an output image obtained by executing an image deformation process for an input image is stored in a memory. In the output image in the memory, a partial image within an effective area set in the memory is specified, and an area including the specified partial image is calculated as a readout area. The partial image is then read out by executing a readout process for the readout area in the memory.

Second Embodiment

Processing of dividing an input image 228 and processing it will be described with reference to FIGS. 5A to 5C. An example shown in FIGS. 5A to 5C shows a case in which an image deformation process shown in FIG. 5A is divided into two processes for left and right portions. A case in which two individual chips respectively perform the divided processes is shown in this example. The division method is not limited to this, and two cores within one chip may perform the processes.

FIG. 5B shows the process for the left portion. Even if the overall deformation process is divided into two as shown in FIG. 5A, a size input on one side needs to be equal to or larger than half the entire size, as indicated by in_w_left in FIGS. 5A to 5C. This is because, in order to output an output image having a size obtained by halving the horizontal size, a post-deformation area on one side needs to cross a process division boundary so that the divided areas are smoothly connected at this boundary. In this case, an area 501 indicates a memory read area.

A description will be provided by associating FIGS. **5**B and **3**A with each other. The input image shown in FIG. **3**A indicates an in_w_left×in_h_all area corresponding to a left area **500** shown in FIG. **5**B. The effective area shown in FIG. **3**A corresponds to an ot_w_left×in_h_all area shown in FIG. **5**B. Furthermore, the memory read area shown in FIG. **3**A corresponds to the memory read area **501** shown in FIG. **5**B. A method of calculating the memory read area **501** is the same as the processing according to the flowcharts shown in FIGS. **6**A and **6**B, and a description thereof will be omitted. This corresponds to the case shown in FIG. **3**A. As described above, if a process is divided, the size of an input image may be different from that of the effective area of an output image. A process shown in FIG. **5**C is the same as that shown in FIG. **5**B, and a description thereof will be omitted.

Note that, in either embodiment, the memory read area is not limited to a rectangular area, and may have an arbitrary shape.

According to each of the above embodiments, if, as shown in FIG. **1**D, a memory access area has a post-deformation shape reduced with respect to the area of an output image like keystone correction, or the like, by a projector, the memory read area becomes smaller than that in the conventional method. This can reduce the memory bandwidth, and decrease the power consumption.

Other Embodiments

Aspects of the present invention can also be realized by a computer of a system or an apparatus (or devices such as a CPU or an MPU) that reads out and executes a program recorded on a memory device to perform the functions of the above-described embodiment(s), and by a method, the steps of which are performed by a computer of a system or an apparatus by, for example, reading out and executing a program recorded on a memory device to perform the functions of the above-described embodiment(s). For this purpose, the program is provided to the computer, for example, via a network or from a recording medium of various types serving as the memory device (for example, a computer-readable medium).

While the present invention has been described with reference to exemplary embodiments, it is to be understood that the invention is not limited to the disclosed exemplary embodiments. The scope of the following claims is to be accorded the broadest interpretation so as to encompass all such modifications and equivalent structures and functions.

What is claimed is:

1. A movie processing apparatus comprising:

an input unit configured to input an input image;

a memory control unit configured to store, in a memory, a deformation image obtained by executing an image deformation process for the input image using a deformation parameter;

a determination unit configured to determine, from a memory region in which the deformation image is stored, a readout region to be accessed for obtaining an output image, on the basis of a size of the input image, a size of the output image, and the deformation parameter;

a readout unit, configured to access the readout region, that reads out pixel data included in the deformation image; and

an output unit configured to output the image based on the pixel data readout by said readout unit.

2. The apparatus according to claim **1**, wherein, if all four vertices of the deformation image fall within an effective area that is determined on the basis of the size of the input image, the size of the output image, and the deformation parameter, said determination unit determines, as the readout area, an area including the four vertices.

3. The apparatus according to claim **1**, wherein, if all four vertices of the deformation image fall outside an effective area that is determined on the basis of the size of the input image, the size of the output image, and the deformation parameter, said determination unit determines the effective area to be the readout area.

4. The apparatus according to claim **1**, wherein, if at least one of four vertices of the deformation image falls within an effective area that is determined on the basis of the size of the input image, the size of the output image, and the deformation parameter, said determination unit determines an intersection point of the effective area and one of sides, among sides of the deformation image, which have as one end point a vertex outside of the effective area and intersect the effective area, and determines, as the readout area, an area including the intersection point and the vertex falling within the effective area.

5. The apparatus according to claim **1**, wherein said memory control unit controls operation to store pixel data included in the deformation image and coordinate values obtained by transforming coordinate values within the input image using the deformation parameter in association with each other,

said readout unit accesses the readout region in order to readout pixel data at coordinate values that correspond to the readout region among coordinate values obtained by transforming coordinate values within the input image using the deformation parameter, and

said readout unit does not access the readout region in order to readout pixel data at coordinate values that do not correspond to the readout region among coordinate values obtained by transforming coordinate values within the input image using the deformation parameter.

6. The apparatus according to claim **1**, further comprising a mask processing unit configured, among pixel data readout by said readout unit, to perform a mask process for pixel data that correspond to a region that is specified based on the deformation parameter, wherein said output unit outputs, as the output image, an image obtained by the mask process performed by said mask processing unit.

7. A movie processing method comprising:

an inputting step to input an input image;

a memory control step to store, in a memory, a deformation image obtained by executing an image deformation process for the input image using a deformation parameter;

a determining step to determine, from a memory region in which the deformation image is stored, a readout region to be accessed for obtaining an output image, on the basis of a size of the input image, a size of the output image, and the deformation parameter;

a readout step, configured to access the readout region, for reading out pixel data included in the deformation image; and

an outputting step to output the image based on the pixel data readout in said readout step.

8. The method according to claim **7**, wherein, if all four vertices of the deformation image fall within an effective area that is determined on the basis of the size of the input image, the size of the output image, and the deformation parameter, said determining step determines, as the readout area, an area including the four vertices.

9. The method according to claim **7**, wherein, if all four vertices of the deformation image fall outside an effective area that is determined on the basis of the size of the input image, the size of the output image, and the deformation parameter, said determining step determines the effective area to be the readout area.

10. The method according to claim **7**, wherein, if at least one of four vertices of the deformation image falls within an effective area that is determined on the basis of the size of the input image, the size of the output image, and the deformation parameter, said determining step determines an intersection point of the effective area and one of sides, among sides of the deformation image, which have as one end point a vertex outside of the effective area and intersect the effective area, and determines, as the readout area, an area including the intersection point and the vertex falling within the effective area.

11. The method according to claim **7**, wherein said memory control step controls operation to store pixel data included in the deformation image and coordinate values obtained by transforming coordinate values within the input image using the deformation parameter in association with each other,

said readout step accesses the readout region in order to readout pixel data at coordinate values that correspond to the readout region among coordinate values obtained by transforming coordinate values within the input image using the deformation parameter, and

said readout step does not access the readout region in order to readout pixel data at coordinate values that do not correspond to the readout region among coordinate values obtained by transforming coordinate values within the input image using the deformation parameter.

12. The method according to claim **7**, further comprising a mask processing step configured, among pixel data readout in said readout step, to perform a mask process for pixel data that correspond to a region that is specified based on the deformation parameter, wherein said outputting step outputs, as the output image, an image obtained by the mask process performed in said mask processing step.

* * * * *