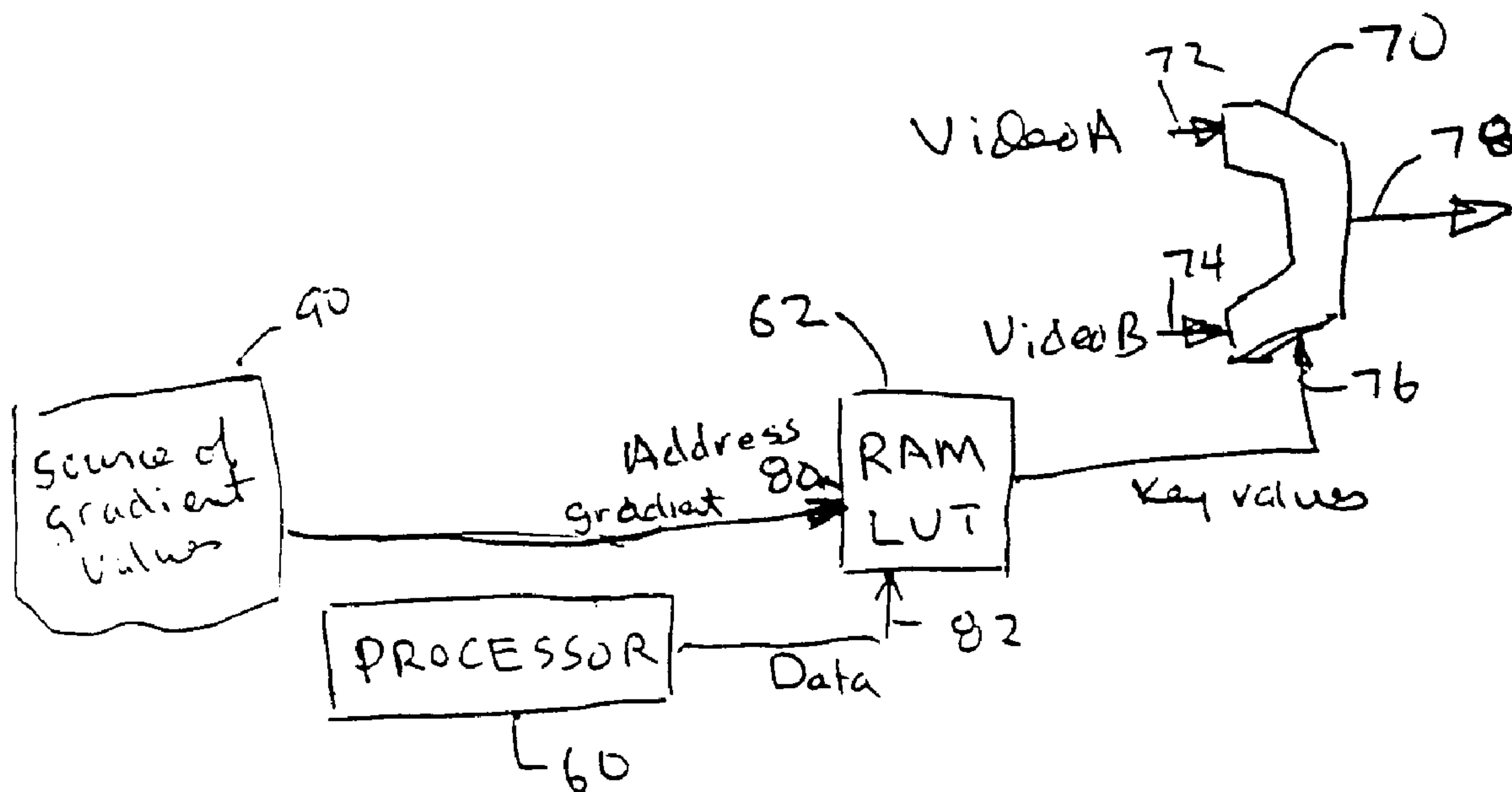




(86) Date de dépôt PCT/PCT Filing Date: 2000/09/20  
 (87) Date publication PCT/PCT Publication Date: 2001/03/29  
 (85) Entrée phase nationale/National Entry: 2002/03/20  
 (86) N° demande PCT/PCT Application No.: US 2000/040944  
 (87) N° publication PCT/PCT Publication No.: 2001/022739  
 (30) Priorité/Priority: 1999/09/21 (09/400,372) US

(51) Cl.Int.<sup>7</sup>/Int.Cl.<sup>7</sup> H04N 9/74  
 (71) Demandeur/Applicant:  
 MEDIA 100, INC., US  
 (72) Inventeurs/Inventors:  
 HATALSKY, JEFFREY F., US;  
 FIERKE, JOHN R., US  
 (74) Agent: SMART & BIGGAR

(54) Titre : REALISATION DE TRANSITIONS VIDEO ENTRE DES CHAINES VIDEO  
 (54) Title: EFFECTING VIDEO TRANSITIONS BETWEEN VIDEO STREAMS



(57) Abrégé/Abstract:

Combining first (72) and second (74) video streams into a combined output video stream (78) by using a gradient table (90) listing gradient values for pixels in a video frame to address a look up table (62) of key values, and combining pixel values of the first and second video streams based upon respective key values read from the look up table (62).

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
29 March 2001 (29.03.2001)

PCT

(10) International Publication Number  
WO 01/22739 A1

(51) International Patent Classification<sup>7</sup>: H04N 9/74

(21) International Application Number: PCT/US00/40944

(22) International Filing Date:  
20 September 2000 (20.09.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:  
09/400,372 21 September 1999 (21.09.1999) US

(71) Applicant: MEDIA 100, INC. [US/US]; 290 Donald Lynch Boulevard, Marlboro, MA 01752 (US).

(72) Inventors: HATALSKY, Jeffrey, F.; 23 Johnson Street, Framingham, MA 01701 (US). FIERKE, John, R.; 10 Wildlife Drive, Hopkinton, MA 01748 (US).

(74) Agent: BOOTH, William, E.; Fish & Richardson, P.C., 225 Franklin Street, Boston, MA 02110-2804 (US).

(81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

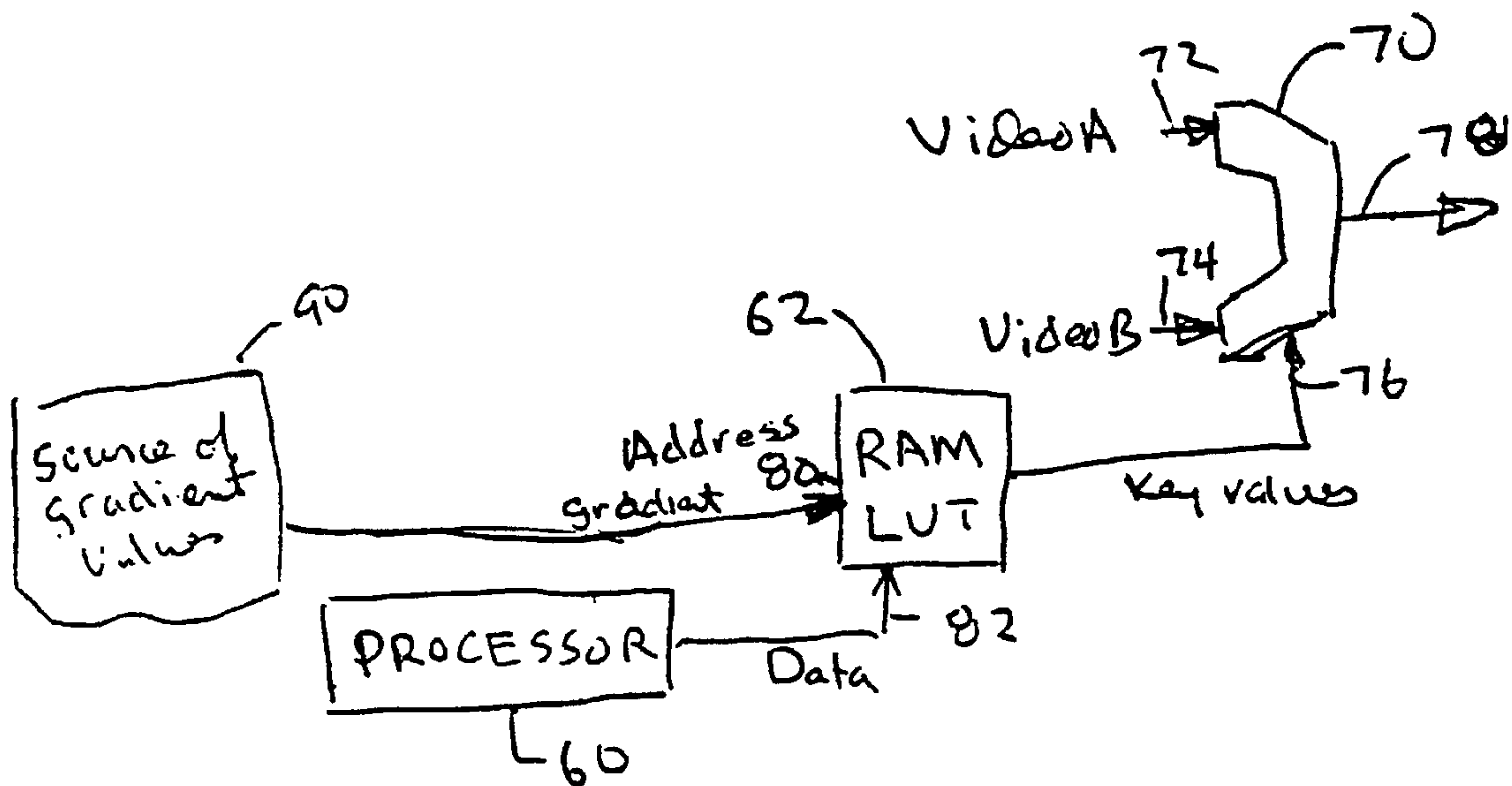
(84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

**Published:**

- With international search report.
- Before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments.

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: EFFECTING VIDEO TRANSITIONS BETWEEN VIDEO STREAMS



(57) Abstract: Combining first (72) and second (74) video streams into a combined output video stream (78) by using a gradient table (90) listing gradient values for pixels in a video frame to address a look up table (62) of key values, and combining pixel values of the first and second video streams based upon respective key values read from the look up table (62).

WO 01/22739 A1

EFFECTING VIDEO TRANSITIONS BETWEEN VIDEO STREAMSBackground of the Invention

The invention relates to effecting video transitions between video streams.

5 In editing of video in the creation of a video program, it is often desirable to provide a gradual transition from one video stream to another video stream. For example, in a wipe transition, a line may move across the screen from left to right, with the old scene  
10 gradually disappearing at the right while more and more of the new scene appears at the left. In a so-called "iris wipe," the boundary is an expanding circle, and one video stream appears in an expanding circle, and the other is outside of the circle. A solid color could also be used  
15 as one stream to provide an image that gradually appears or disappears across the screen or an image that gradually expands or contracts.

Referring to Fig. 1, transitions can be created at a video combiner 10 having a first video input 12 for a  
20 first video stream, a second video input 14 for a second video stream, a key input 16 to receive key values indicating how the two inputs are to be combined, and a video output 18 for the output video that is some combination of the two inputs. The video streams include  
25 frames (each frame providing an image for a screen) made up of horizontal lines, each of which includes individual "pixels" (picture elements) across the line. Each pixel value includes a number of bit values that describe the color and intensity at that particular pixel at a  
30 particular time. New frames are provided at the video rate, e.g., 30 frames per second, and the video streams include sequential digital data describing sequential pixel values for lines of video data in a frame. The pixels for the first and second video streams that are  
35 input to combiner 10 are synchronized, as are the associated key values. The key value for a given pixel indicates whether the output for that pixel is the input

from one stream or the other stream or a combination of the two.

While a unique key value could be stored and accessed for each pixel in each frame, to reduce bandwidth, a single table of gamma values (also referred to as gradient values herein) can be used. For example, a single table with one value for each pixel can be used to define the wipe over the sequence of frames. In the table, the gamma value essentially indicates the time at which the transition appears at that pixel. Fig. 2 is a simplified diagram of a table of gamma values for an iris wipe of a square (instead of circular) transition. (A real table for NTSC would have 720 horizontal entries and 480 vertical entries.) As the time goes from frames 1 to 2 to 3 etc, a "threshold" value similarly goes from 1 to 2 to 3 etc (or 1, 1, 2, 2, 3, 3 or 1, 2, 3, 3, 4 to spread things out, though the latter example would not be uniform). The threshold is compared to the value in the table to determine the key value (e.g., a 0 key value meaning all of video stream A, a 100 key value meaning all of video stream B, and a 50 key value meaning equal amounts of A and B) that is fed to the mixer. The key values that are generated cause the video output to switch over from one video input to the other. As the threshold increases, the boundary of the transition provided by use of the Fig. 2 table expands. The following algorithm employs a direct comparison that provides an abrupt transition as the threshold goes from 0 to 4, as shown in displays 22, 24, 26, 28, 30 in Figs. 3A-3E, respectively.

```
If (gradient < threshold)
    then wipe = transparent
else
    wipe = opaque
```

"Softness" can be added to the key values generated so that the change from one video to the other is not abrupt but instead is gradual (with decreasing amounts of one

video stream and increasing amounts of another) in a discrete number of adjacent pixels as the transition passes a pixel location. For example, the following algorithm can be used to produce a gradual transition  
 5 region N pixels wide.

```

If (gradient < (threshold -N) then
    wipe = transparent
else if (gradient >= (threshold-N) AND (gradient < threshold) then
10    wipe = ½ *opaque* (1-(threshold - gradient)/N)
else if (gradient >= threshold) AND (gradient < (threshold + N)) then
    wipe = ½ *opaque* (1 + (gradient - threshold)/N)
else if (gradient >= (threshold +N))
15    wipe = opaque
  
```

15 With this algorithm, the transition from one video stream to the other is linear, though the human eye sees brightness more as a cube root function. Also, the threshold may only be moved on integer pixel boundaries,  
 20 which can cause abrupt changes in movement when the wipe moves X pixels in Y frames, and X and Y are not multiples of each other.

#### Summary of the Invention

25 The invention features, in general, combining first and second video streams into a combined output video stream by using a gradient table listing gradient values for pixels in a video frame to address a look up table of key values, and combining pixel values of the first and  
 30 second video streams based upon respective key values read from the look up table.

Particular embodiments of the invention may include one or more of the following features. The video is combined at a video combiner having an input for the first  
 35 video stream, an input for the second video stream, an input for the key values, and a video output. New key values are loaded into the look up table between frames.

The new values loaded into the look up table provide a nonlinear interpolation. The new values loaded into the look up table provide a transition over a noninteger number of pixels per frame. The new values loaded into  
5 the look up table provide nonlinear softness on the edges of the transition.

Embodiments of the invention may include one or more of the following advantages. Use of a look up table instead of thresholds permits one to create soft  
10 transitions that vary in other than a linear way. The look up table also permits one to move the transition boundary by less than one pixel at one time.

Other advantages of the invention will be apparent from the following description of a particular embodiment  
15 thereof and from the claims.

#### Brief Description of the Drawings

Fig. 1 is a block diagram showing a prior art video combiner for combining first and second video streams.

20 Fig. 2 is a diagram of a prior art table of gamma values used to generate key values for the Fig. 1 video combiner.

Figs. 3A-3E are diagrams of sequential frames showing a transition resulting from the table of Fig. 2.

25 Fig. 4 shows a video editing system.

Fig. 5 shows some of the components of a video editing card used in the Fig. 6 system.

Fig. 6 is a table providing an exponential mapping of gradient values into key values in the Fig. 4 system.

30 Figs. 7A-7E are a sequence of tables showing a mapping of gradient values into key values that results in moving a transition over a distance other than an integer pixel in the Fig. 4 system.

#### Description of the Preferred Embodiment

Referring to Fig. 4, video editing system 40 is implemented by computer 42, video editing software 44

running on computer 42, and video editing expansion card 46 plugged into computer 42. VTR 48 is a source of video streams that can be stored on disk or other mass storage 50 and randomly accessed by computer 42. Keyboard 52 and 5 mouse 54 are user input devices, and monitor 56 is used to provide a video editing interface including display of a program being created. An additional monitor (not shown) can also be used to play the video program. U.S. Patents Nos. 5,506,932; 5,488,695; 5,471,577; 5,909,250, which are 10 hereby incorporated by reference, describe video editing systems implemented on a computer.

Referring to Fig. 5, video editing card 46 includes on-board microprocessor 60, random access memory look up table (RAM LUT) 62 and video combiner 70, which has first 15 and second video inputs 72, 74, key input 76, and video output 78. RAM LUT 62 receives a stream of gradient values from source of gradient values 90 at address pins 80. Typically, source 90 would be a RAM storing the gradient table, and the RAM would be addressed with 20 sequential addresses to generate a stream of gradient values. A new gradient value, for the next pixel in the frame, is inputted with each pixel clock at address pins 80 of RAM LUT 62. New data values from microprocessor 60 are inputted via data pins 82 into RAM LUT 62 between each 25 frame.

In operation, RAM LUT 62 is used to provide a stream of key values to video combiner 70 in synchronization with the pixel values for video streams A and B being inputted to inputs 72 and 74. At each pixel 30 clock, the pixel values for the next pixel in the line are inputted to first and second video inputs 72, 74, and the corresponding key value from RAM LUT 62 is input to key input 76. At each pixel clock a new gradient value is read from the gradient table and applied to the address 35 pins 80 of RAM LUT 62, and at each pixel clock, a new key value is read from RAM LUT 62. The video streams A, B (Fig. 5) are delayed as necessary with respect to the

gradient stream so that keys at key input 76 of video combiner 70 line up with respective pixel values at video inputs 72, 74 of video combiner 70. The video at output 78 is the combination of video A and B indicated by the key value. Between each frame, new key values are loaded into RAM LUT 62 by microprocessor 60.

While loading new key values into RAM LUT 62 is more computationally intensive than setting a threshold value, it is still significantly easier than generating the entire wipe every frame, and much of the temporal compression afforded by gradient wipes is still present. RAM LUT 62 can be loaded with any desired type of transition and is thus not limited to linear transitions as with the algorithm described above in the Background section. For example, at a threshold of 43, an exponential transition (0 = all video A, 100 = all video B) could be provided by the entries shown in the table of Fig. 6. While the table has been truncated, it still demonstrates a non-linear transition region.

RAM LUT 62 can also be used to move the transition by less than one pixel at a time or by a noninteger value. Consider the following simple case, a hard edge threshold at 57 to be moved to 59 over 4 frames. With a simple threshold (as described above in the Background section) one would need to repeat a threshold value in the sequence of four frames, e.g., use 57, 58, 58, 59 or 57, 57, 58, 59 or a similar sequence. Even with a method for adding softness to either side of the threshold (as described above in the Background section), the threshold itself may still only move an integer pixel every frame.

With RAM LUT 62, one can achieve a move from 57 to 58 over four frames using the entries in the tables in Figs. 7A-7E. In essence, the transition can be moved one pixel over four frames. This example has two simplifications that are not limitations of the architecture. The example (Figs. 7A-7E) does not have softness (as noted above, any softness, linear or other,



can be added), and the interpolation method used is a simple linear interpolation (any desired temporal change can be employed by selection of the key values in the table).

5           The values of the tables can be generated on video editing card 46 or precomputed and stored on local memory on video editing card 46 and easily loaded into RAM LUT 62 between frames.

10           Other embodiments of the invention are within the scope of the appended claims.

What is claimed is:

1. A method of combining first and second video streams into a combined output video stream comprising providing a gradient table listing gradient values for pixels in a video frame made up of lines of pixels,  
5 each said gradient value relating to a time at which a transition crosses said pixel, said transition being from said first video stream to said second video stream,  
reading gradient values for sequential said pixels from said table,  
10 using said gradient values to address a look up table of key values,  
reading out key values addressed by said gradient values, and  
combining pixel values of said first and second  
15 video streams based upon respective said key values.
2. The method of claim 1 wherein said combining is carried out at a video combiner having an input for said first video stream, an input for said second video stream,  
20 an input for said key values, and a video output.
3. The method of claim 1, further comprising loading new values into said look up table between frames.
- 25 4. The method of claim 3 wherein said new values loaded into said look up table provide a nonlinear interpolation.
5. The method of claim 3 wherein said new values  
30 loaded into said look up table provide a transition over a noninteger number of pixels per frame.
6. The method of claim 3 wherein said new values loaded into said look up table provide nonlinear softness  
35 on the edges of said transition.
7. Apparatus for combining first and second video

streams into a combined output video stream comprising  
circuitry providing sequential gradient values from  
a gradient table listing gradient values for pixels in a  
video frame made up of lines of pixels, each said gradient  
5 value relating to a time at which a transition crosses  
said pixel, said transition being from said first video  
stream to said second video stream,

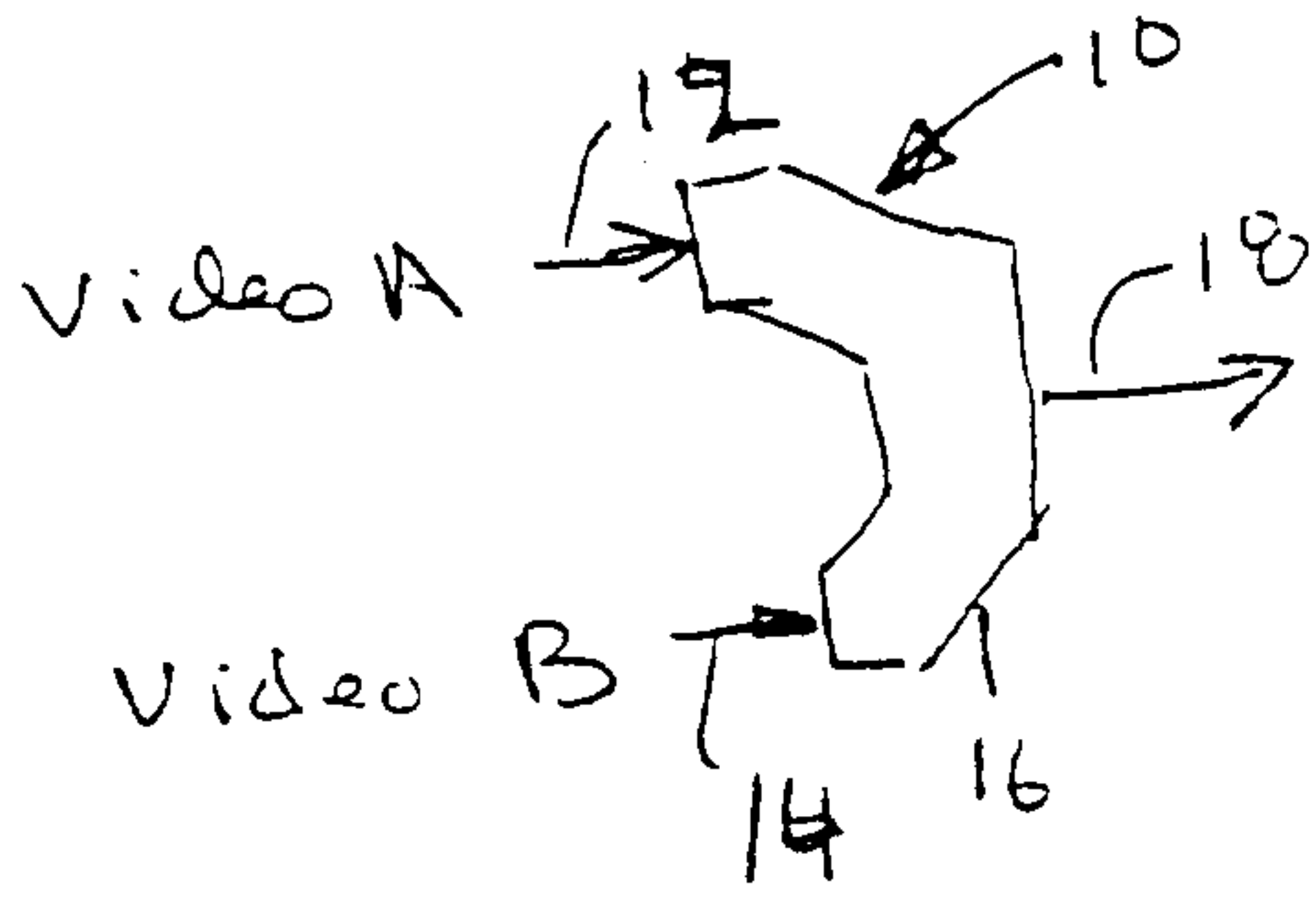
a random access memory look up table (RAM LUT) of  
key values that is connected to be addressed by said  
10 sequential gradient values of said gradient table, and  
a video combiner connected to receive said key  
values read from said RAM LUT and having inputs for first  
and second video streams, said combiner combining pixel  
values of said first and second video streams based upon  
15 respective said key values.

8. The apparatus of claim 7 further comprising a  
processor that loads new values into said RAM LUT between  
frames.  
20

9. The apparatus of claim 8 wherein said processor  
loads new values into said RAM LUT to provide a nonlinear  
interpolation.

25 10. The apparatus of claim 8 wherein said processor  
loads new values into said RAM LUT to provide a transition  
over a noninteger number of pixels per frame.

11. The apparatus of claim 8 wherein said processor  
30 loads new values into said RAM LUT to provide nonlinear  
softness on the edges of said transition.



PRIOR ART

Fig. 1

4	4	4	4	4	4	4
4	3	3	3	3	3	4
4	3	2	2	2	3	4
4	3	2	1	2	3	4
4	3	2	2	2	3	4
4	3	3	3	3	3	4
4	4	4	4	4	4	4

20

Fig. 2

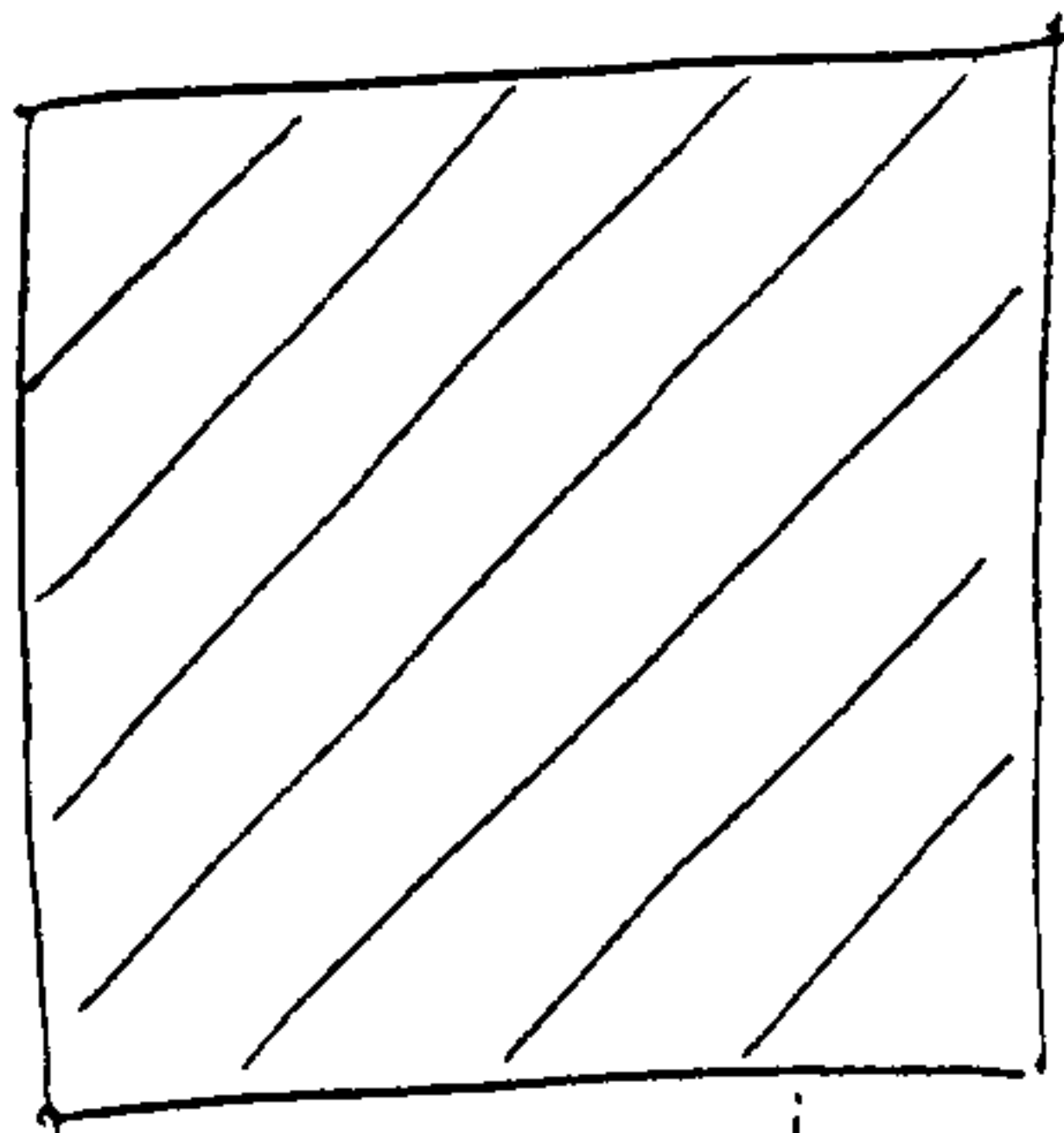


Fig. 3A

L22

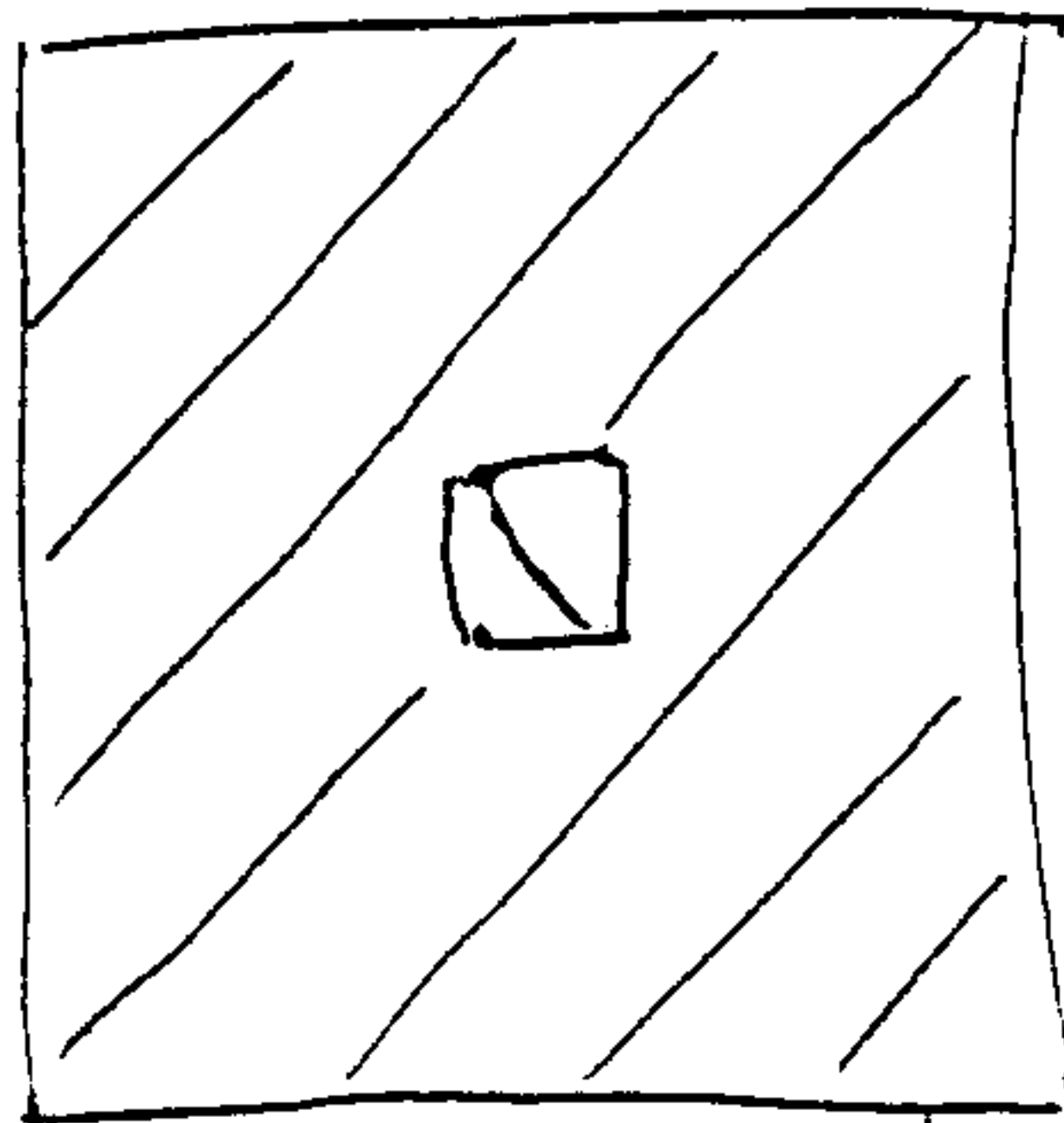


Fig. 3B

L24

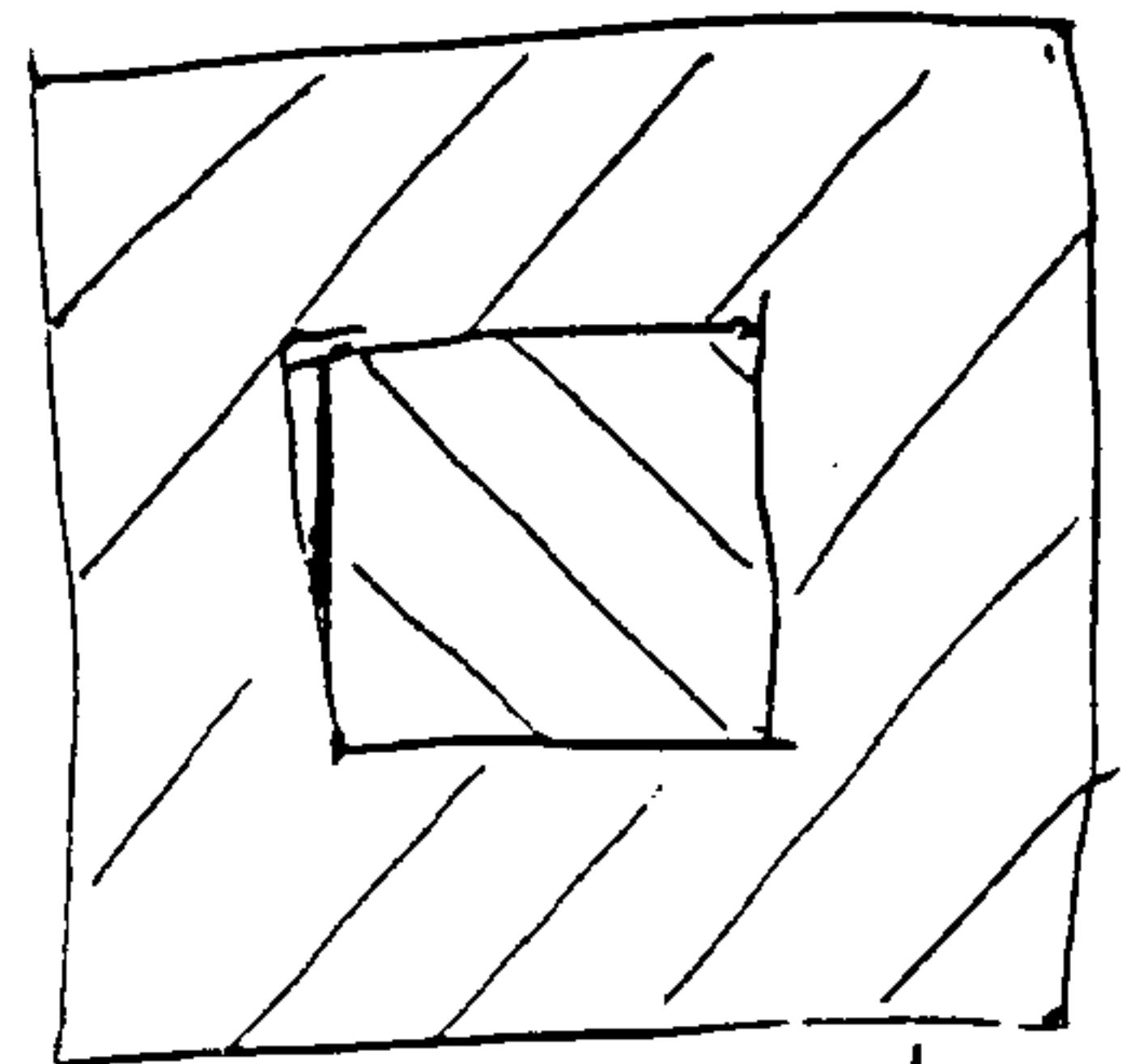


Fig. 3C

L26

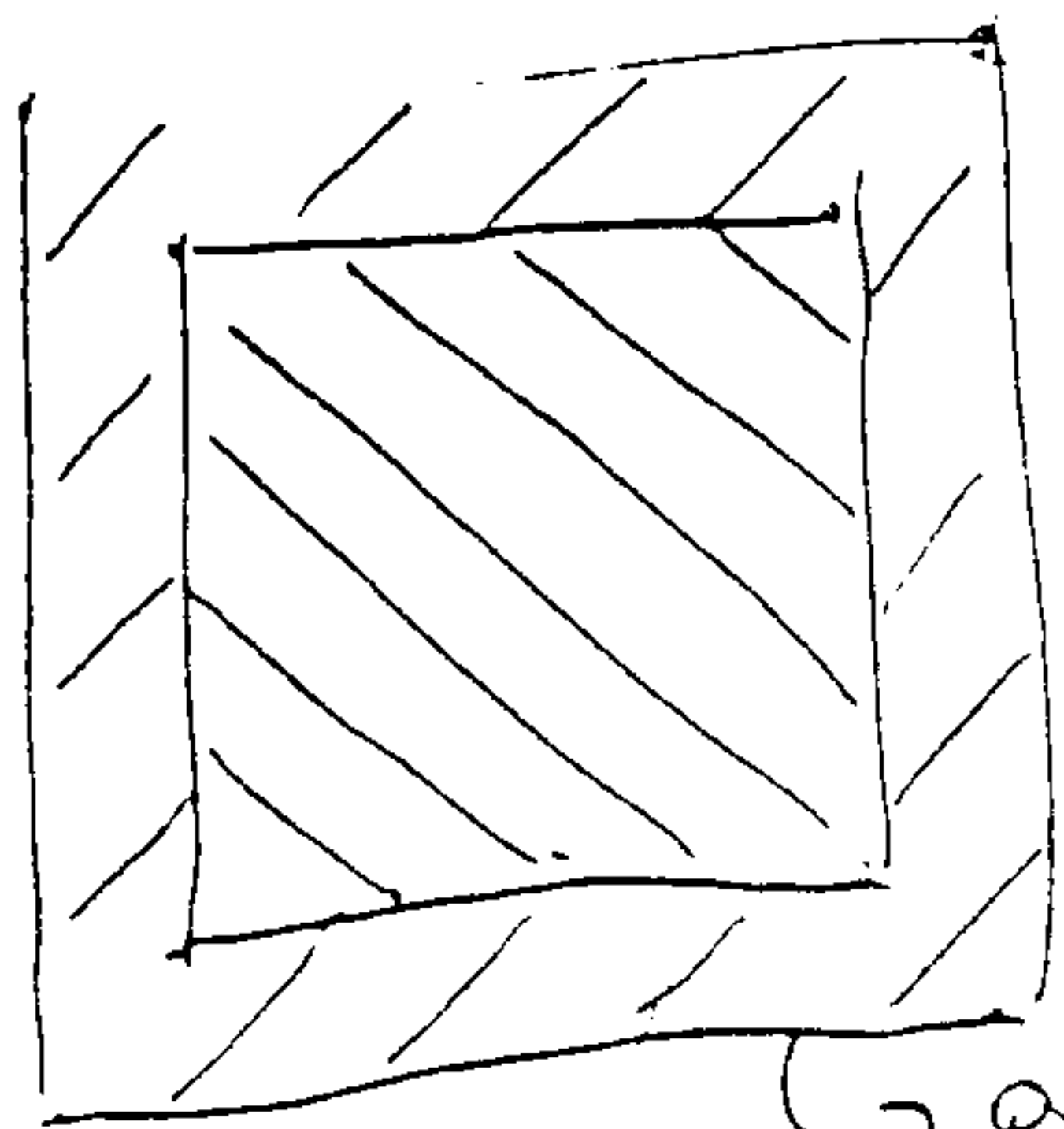


Fig. 3D

L28

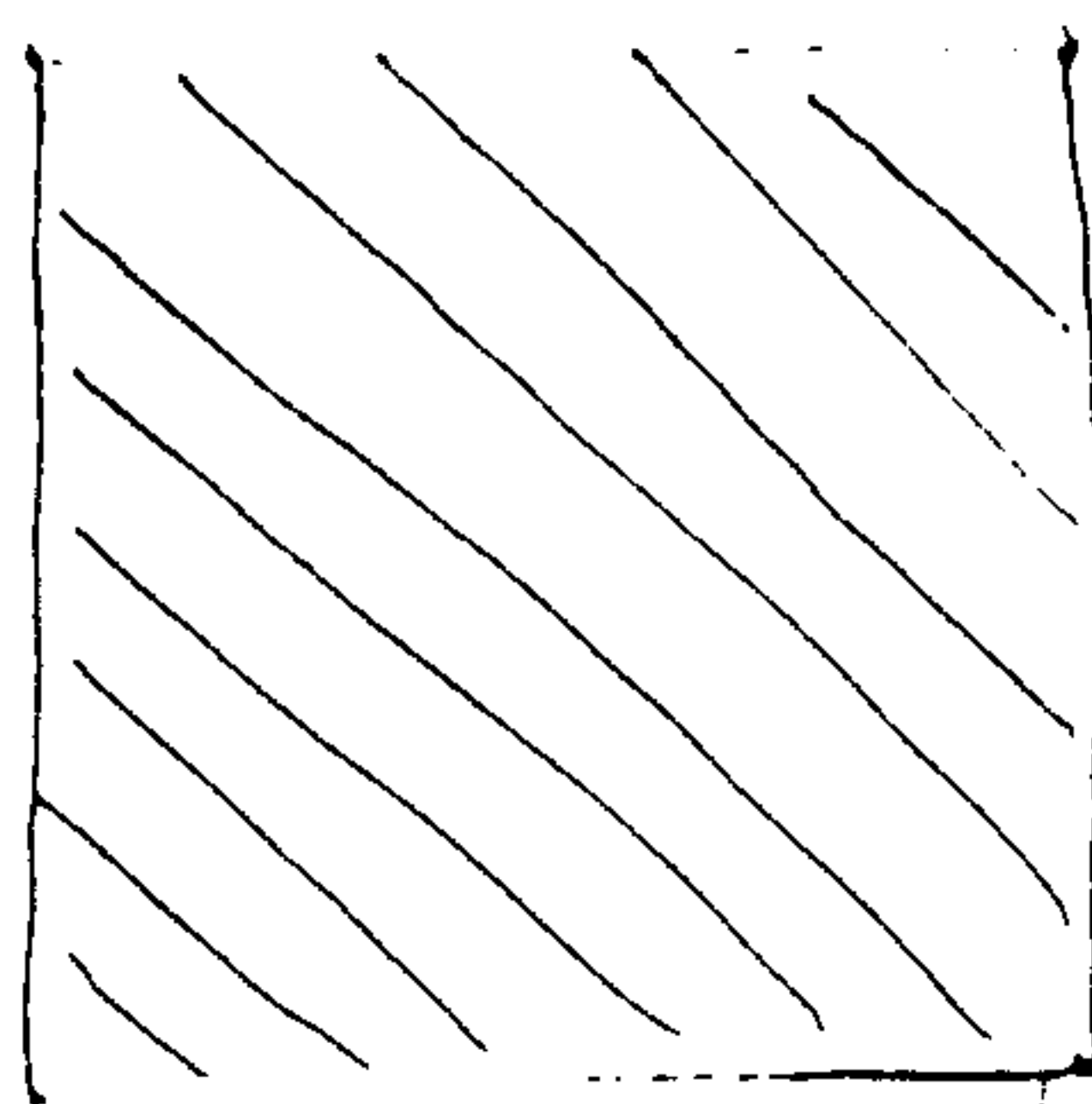


Fig. 3E

L30

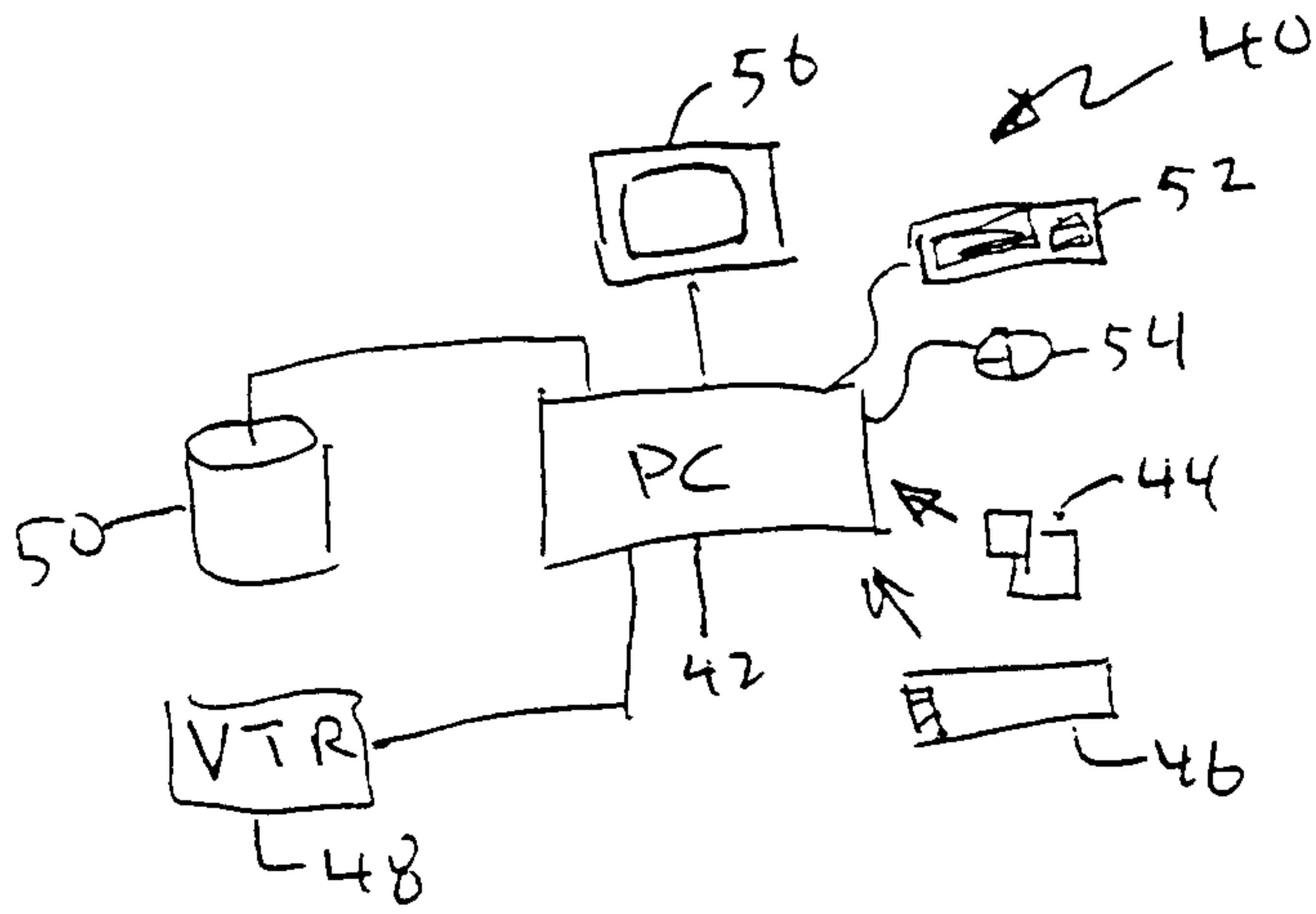


Fig. 4

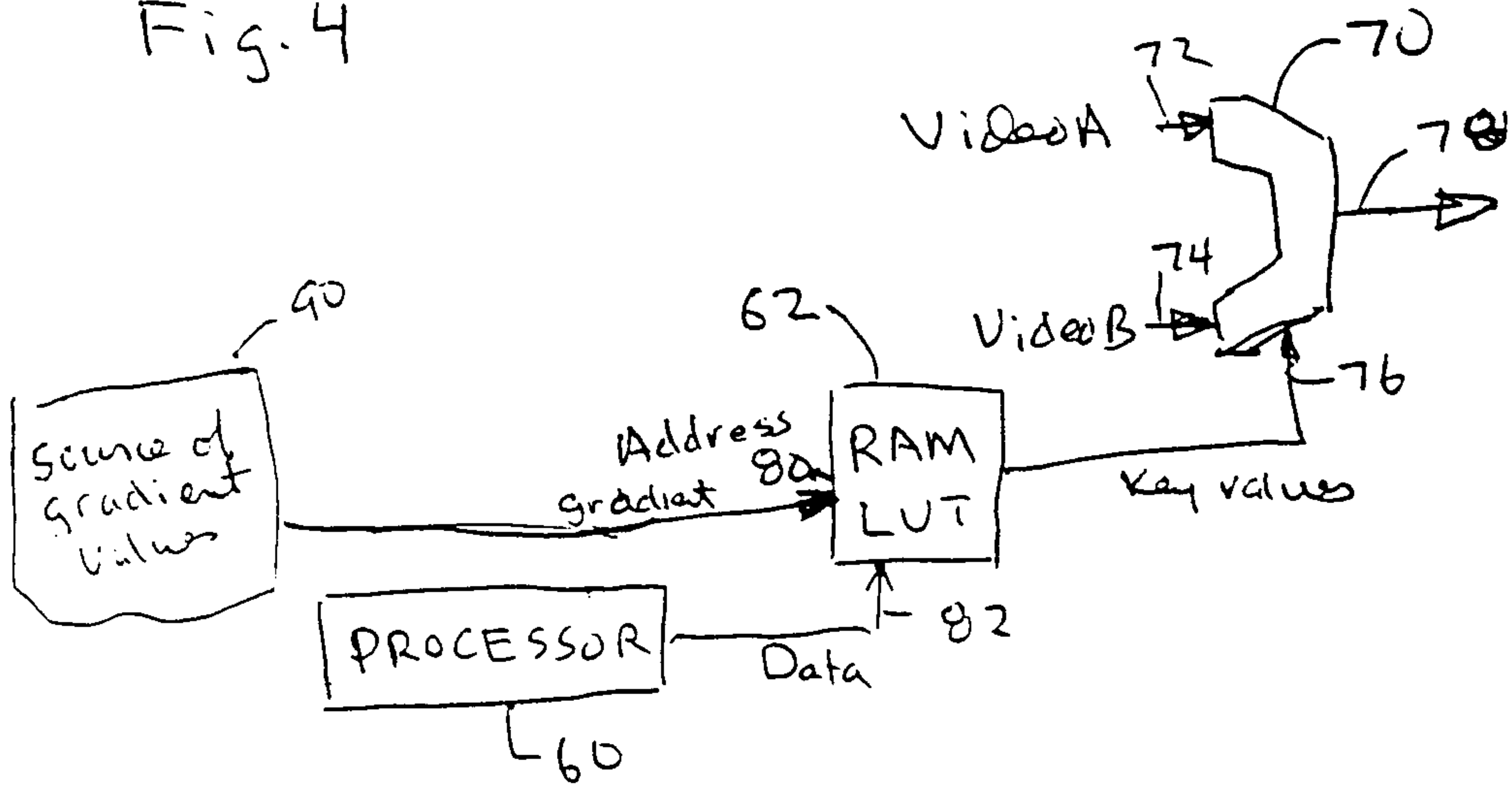


Fig. 5

LUT address	Value
37	30
38	32
39	35
40	38
41	42
42	46
43	50
44	55
45	60
46	65
47	72
48	79
49	86
50	94
51-100	100

Fig. 6

LUT address	Wipe value
55	0
56	0
57	100
58	100
59	100
60	100
61	100

Fig. 7A

LUT address	Wipe value
55	0
56	0
57	75
58	100
59	100
60	100
61	100

Fig. 7B

LUT address	Wipe value
55	0
56	0
57	50
58	100
59	100
60	100
61	100

Fig. 7C

LUT address	Wipe value
55	0
56	0
57	25
58	100
59	100
60	100
61	100

Fig. 7D

LUT address	Wipe value
55	0
56	0
57	0
58	100
59	100
60	100
61	100

Fig. 7E

