



(51) International Patent Classification:

G09G 5/393 (2006.01) G09G 5/399 (2006.01)  
G09G 5/395 (2006.01)

(21) International Application Number:

PCT/GB2021/050028

(22) International Filing Date:

06 January 2021 (06.01.2021)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

2000127.7 06 January 2020 (06.01.2020) GB

(71) Applicant: **DISPLAYLINK (UK) LIMITED** [GB/GB];  
22 Cambridge Science Park, Milton Road, Cambridge CB4  
0GH (GB).

(72) Inventors: **PARKS, Matthew**; 1A Saberton Close, Wa-  
terbeach, Cambridge Cambridgeshire CB25 9QW (GB).

**SKAN, Peter**; 23 Tarrant Drive, Harpenden Hertfordshire  
AL5 1RP (GB).

(74) Agent: **HIRSZ, Christopher**; Mathys & Squire, The  
Shard, 32 London Bridge Street, London Greater London  
SE1 9SG (GB).

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

(84) Designated States (unless otherwise indicated, for every  
kind of regional protection available): ARIPO (BW, GH,

(54) Title: MANAGING DISPLAY DATA

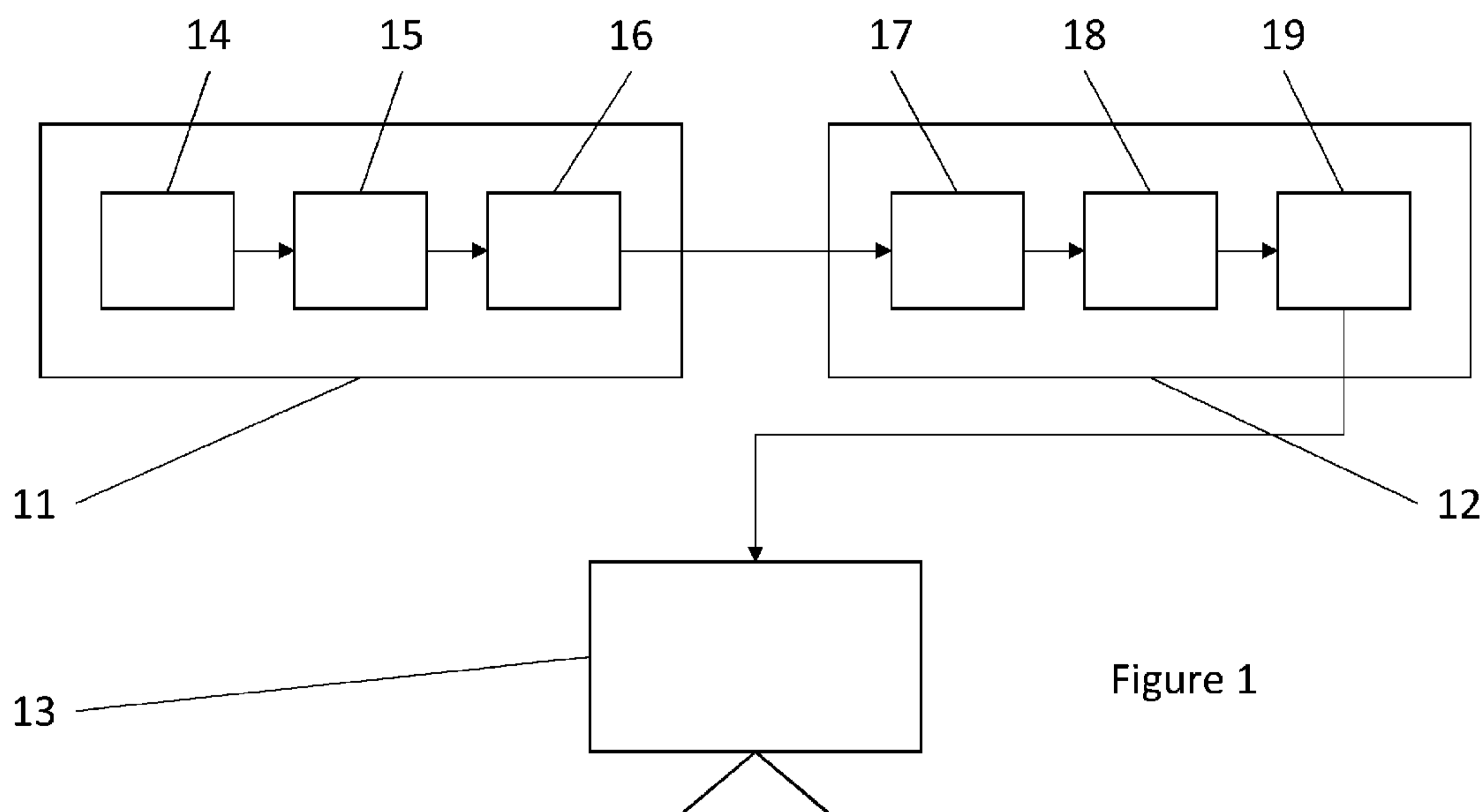


Figure 1

(57) Abstract: Disclosed herein is a method of writing data to, and reading data from, one or more buffers. The method comprises: determining a write rate of writing data into a first buffer; determining a read rate of reading data from the first buffer; determining, using the write rate and the read rate, a portion of the first buffer; writing data into the portion of the first buffer; starting to read data from the first buffer when the writing of data to the portion of the first buffer has finished; and writing data into a remaining part of the first buffer, different from the portion of the first buffer. The portion of the first buffer is determined such that the reading of data from the first buffer does not overtake the writing of data into the first buffer.

WO 2021/140324 A1

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).

**Published:**

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

# Managing Display Data

## *Background*

This invention relates, generally, to managing data, for example display data. In many computing systems, especially display systems, it is common to stream data through a buffer: i.e. for one process to write data into the buffer and a second process to read the data. The read and write processes ideally traverse the data as close together in time as possible such that there is as small a delay as possible between a specific datum being written and being read, and ideally this time delay is constant. However, it is common for read processes to be faster than write processes, which can result in the processes losing synchronisation and possibly the read process attempting to read data that has not been written. It will be apparent that a buffer may well have previously written data stored in it, or may be at least partly empty (no data, which is not the same as data that is zero). When a buffer is written to, the write process overwrites what was previously there, whether no data, or previously written data. Similarly, when a buffer is read, the read process reads whatever data (or no data) it comes to, whether that data has been written in the current writing process or has been written previously.

In the case of a display system, it is common for the loss of synchronisation to be mitigated by the use of multiple buffers such that one buffer is written while the other is read, and a read process can only begin on a particular buffer once the write process has finished. Commonly, three buffers are provided in order to ensure that there will be no overlap, and once the write process has finished writing data to a given buffer it sets a flag indicating which buffer it will write to next. When the read process reaches the end of a given buffer it proceeds to the next buffer, according to the flag. However, this results in increased and unpredictable latency since an entire frame of display data must be written before the directional flag can be set, followed by an unknown length of time for the read process to catch up, ranging from immediate to the time it takes to read an entire frame.

These increased and unpredictable delays are unacceptable in display and other streaming systems, especially as technology improves and delays in updates become more noticeable to the user. The method of the invention therefore seeks to solve or at least mitigate these problems.

## *Summary*

Aspects of the invention are set out in the independent claims and preferred features are set out in the dependent claims.

In one aspect the invention provides a method of writing data to, and reading data from, one or more buffers, the method comprising:

determining a write rate of writing data into a first buffer;  
determining a read rate of reading data from the first buffer;  
determining, using the write rate and the read rate, a portion of the first buffer;  
writing data into the portion of the first buffer;

5 starting to read data from the first buffer when the writing of data to the portion of the first buffer has finished; and

writing data into a remaining part of the first buffer, different from the portion of the first buffer;

10 wherein the portion of the first buffer is determined such that the reading of data from the first buffer does not overtake the writing of data into the first buffer.

The first buffer may be a first-in first-out, FIFO, buffer.

The method may further comprise:

determining a write rate of writing data into a second buffer;  
determining a read rate of reading data from the second buffer;

15 determining, using the write rate and the read rate, a portion of the second buffer;

writing data into the portion of the second buffer when the writing of data to the remaining part of the first buffer has finished;

continuing to read data from the first buffer whilst data is written to the portion of the second buffer;

20 starting to read data from the second buffer when the writing of data to the portion of the second buffer is finished; and

writing data into a remaining part of the second buffer, different from the portion of the second buffer;

25 wherein the portion of the second buffer is determined such that the reading of data from the second buffer does not overtake the writing of data into the second buffer.

The second buffer may be a first-in first-out, FIFO, buffer.

The step of determining the portion of the second buffer may be based on a ratio of the read rate of reading data from the second buffer and the write rate of writing data to the second buffer.

30 The step of determining the portion of the first buffer may be based on a ratio of the read rate of reading data from the first buffer and the write rate of writing data to the first buffer.

The portion of the first buffer may be calculated using the formula:

$$P = B \left(1 - \frac{1}{R}\right)$$

where P is the portion, B is the size of the first buffer, and R is the read rate of reading data from the first buffer divided by the write rate of writing data to the first buffer.

5 The portion of the first buffer may be calculated using the formula:

$$P = B \left(1 - \frac{1}{R}\right) + C$$

where P is the portion, B is the buffer size of the first buffer, R is the read rate of reading data from the first buffer divided by the write rate of writing data to the first buffer, and C is a constant.

In one example, only updated parts of the first buffer are written to.

10 In a second aspect the invention provides a method of writing data to, and reading data from, a buffer, the method comprising:

determining a write rate of writing data into the buffer;

determining a read rate of reading data from the buffer;

writing data into the buffer;

15 determining, using the write rate and the read rate, a time at which data is to begin to be read from the buffer, wherein the time is determined such that the reading of data from the buffer will not overtake the writing of data into the buffer; and

starting to read data from the buffer at the determined time.

The buffer may be a first-in first-out, FIFO, buffer.

20 In one example, only updated parts of the buffer are written to; and the time at which data is to begin to be read from the buffer is calculated based on the time at which the writing of the updated parts to the buffer will be complete.

In one example, all of the data in the buffer is read.

The data may be display data. The data may comprise tiles of display data.

25 In a third aspect the invention provides a computing device comprising a plurality of frame buffers, wherein the computing device is configured to perform the method according to the first aspect or the second aspect.

In a fourth aspect the invention provides a system comprising:

a non-transitory memory storing instructions; and

one or more hardware processors coupled to the non-transitory memory and configured to execute the instructions from the non-transitory memory to cause the system to perform operations comprising:

determining a write rate of writing data into a first buffer;

5 determining a read rate of reading data from the first buffer;

determining, using the write rate and the read rate, a portion of the first buffer;

writing data into the portion of the first buffer;

starting to read data from the first buffer when the writing of data to the portion of the first buffer has finished; and

10 writing data into a remaining part of the first buffer, different from the portion of the first buffer;

wherein the portion of the first buffer is determined such that the reading of data from the first buffer does not overtake the writing of data into the first buffer.

In a fifth aspect the invention provides a system comprising:

15 a non-transitory memory storing instructions; and

one or more hardware processors coupled to the non-transitory memory and configured to execute the instructions from the non-transitory memory to cause the system to perform operations comprising:

determining a write rate of writing data into a buffer;

20 determining a read rate of reading data from the buffer;

writing data into the buffer;

determining, using the write rate and the read rate, a time at which data is to begin to be read from the buffer, wherein the time is determined such that the reading of data from the buffer will not overtake the writing of data into the first buffer; and

25 starting to read data from the first buffer at the determined time.

In one example, a method of streaming data through a pair of buffers comprises:

1. Determining relative rates of a write process and a read process (the Ratio);
2. Determining a portion of each buffer based on the Ratio (the Portion);
3. The write process writing data into a first buffer up to the end of the Portion;
- 30 4. The read process receiving a switch signal and beginning to read the first buffer;
5. The read process reading the first buffer repeatedly until it receives a further switch signal;

6. When the write process has written data into the whole of the first buffer, the write process writing data into the second buffer up to the end of the Portion;
7. The read process receiving a further switch signal and beginning to read the second buffer; and
- 5 8. When the write process has written data into the whole of the second buffer, the process returning to step 3 above.

Steps 3-8 are repeated until the data stream is complete.

This method is beneficial because it co-ordinates the write and read processes based on the relative time each takes to write data to and read it from the buffers, respectively. Since the read process in a particular buffer begins as soon as enough data has been written to the buffer, there is a reduced delay before the data can be streamed to an endpoint, but because the readiness of the data is determined based on the relative rates of the write and read process the chance of out-of-date data being read is beneficially reduced.

This method can be used in any streaming system but is advantageously used where the data is display data and each of the two buffers are frame buffers.

The Ratio is advantageously determined by a formula:

$$Ratio = \frac{Time\ to\ Write\ Buffer}{Time\ to\ Read\ Buffer}$$

Or alternatively:

$$Ratio = \frac{Rate\ of\ Read\ Process}{Rate\ of\ Write\ Process}$$

Which gives the same result for the Ratio since the time required for a process and the rate of the process are inversely proportional, such that a process which proceeds at a rate of 2x will take half the time of a process that proceeds at a rate of x.

Therefore, the Portion can therefore be determined by a second formula:

$$Portion = Buffer\ Size \left( 1 - \frac{1}{Ratio} \right)$$

This formula can advantageously be amended to allow an offset to the Portion to be added in order to allow for delays in the write process, such that the write process writes a constant amount beyond the Portion before the signal is sent to the read process.

The Portion and the Buffer Size can be measured in any appropriate unit of measurement. For example:

- Data volume in bits, nibbles, bytes, kilobytes, etc.

- Number of pixels, tiles, slices, rows, etc.
- Physical location in an area of memory

It will also be appreciated that the method may comprise calculating a Proportion of the buffer according to:

$$5 \quad \textit{Proportion} = 1 - \frac{1}{\textit{Ratio}}$$

In this case, the method may comprise calculating, using the proportion, a time at which data may begin to be read from the buffer such that the read process will not overtake the write process.

Any system feature as described herein may also be provided as a method feature, and vice versa. As used herein, means plus function features may be expressed alternatively in terms of their  
10 corresponding structure.

Any feature in one aspect of the invention may be applied to other aspects of the invention, in any appropriate combination. In particular, method aspects may be applied to system aspects, and vice versa. Furthermore, any, some and/or all features in one aspect can be applied to any, some and/or all features in any other aspect, in any appropriate combination.

15 It should also be appreciated that particular combinations of the various features described and defined in any aspects of the invention can be implemented and/or supplied and/or used independently.

### ***Brief Description of the Drawings***

Embodiments will now be described, by way of example only and with reference to the  
20 accompanying drawings, in which:

Figure 1 shows an overview of a display system;

Figure 2 shows processes interacting with a buffer;

Figure 3 shows a process;

Figure 4 shows an example with a first write process rate;

25 Figure 5 shows an example with a second write process rate; and

Figure 6 shows a block diagram of a computer system.

### ***Detailed Description of the Drawings***

The abbreviations used in the Figures and the equations below are as follows:

- W: The write process



- R: The read process
- T: The time delay between the read and write processes
- $T_R$ : The time required to read the whole buffer
- $T_W$ : The time required to write the whole buffer

5 Figure 1 shows an overview of an example system in which the method of an embodiment of the invention could be used. In this example, the system is a display system, comprising a host computing device [11] which produces and transmits display data, connected to a display control device [12] which prepares the received display data for display and which is connected to a display device [13]. Naturally, this is an example only, and the embodiments of the invention could be used in other  
10 streaming contexts, including contexts where the data is transmitted within a single device or where data is received from or transmitted to multiple devices. Accordingly, the connections between the devices and between the components described in this Figure may be over any internal, local, or network interface.

The host computing device [11] incorporates an application [14] which generates a stream of  
15 frames of display data, for example as part of a video stream. The application [14] runs in a processor [14] which is connected to an encoder [15]. The encoder [15] compresses the display data in preparation for transmission across a limited-bandwidth connection to the display control device [12]. The encoder [15] may also encrypt the data or carry out any other processing required. The encoder [15] is in turn connected to a frame buffer [16] which stores the data prior to transmission. There may  
20 also be a frame buffer between the application [14] and the encoder [15], but it is not shown here.

The display control device [12] incorporates a decoder [17] which reverses the processing carried out by the encoder [15], for example decompression. The decoder [17] is connected to a frame buffer [18] which stores the decoded data before it is sent to the display device [13] for display. Accordingly, this frame buffer [18] is connected to an output engine [19] which transmits the display data to the  
25 display device [13]. There may also be a frame buffer which stores the received data before it is decoded, but it is not shown here.

Streaming can be carried out through any of the frame buffers in the system, and any such frame buffer may actually be implemented as two or more frame buffers. For the purposes of this description, streaming takes place through the frame buffer [18] that stores the decoded data before it  
30 is sent to the display device [13].

Figure 2 shows the operation of conventional read and write processes within a single buffer.

The uppermost arrow shown in the Figure represents the read process, and the lower arrow represents the write process. The write process writes data into the buffer and the read process follows behind, reading the data. Ideally, there is a time delay, illustrated by the arrow labelled 'T', between

the write and read processes which remains constant and the read process does not overtake the write process. However, if this does occur, the read process may output out-of-date data, which in a display system such as that shown in Figure 1 may result in visual artefacts such as tearing, and in an audio system may result in audio artefacts such as skipping. The time delay between the write and read processes is added to the overall latency of the system.

In practice, the time delay between the write and read processes cannot be maintained because the read process is commonly faster than the write process.

For example, the time required to read the whole buffer may be half of the time required to write the whole buffer:

$$T_R = \frac{T_W}{2}$$

In this case, if the read and write process begin simultaneously, by the time the buffer has been filled by the write process it would have been read twice.

Figure 3 shows a process in a display system such as that shown in Figure 1, assuming the buffer to which the process is applied is the frame buffer [18] connected to the decoder [17] in the display control device [12], which as previously mentioned may be implemented as two or more buffers that may share a single area of memory. This is an example only; similar methods could also be used for the frame buffer [16] connected to the encoder [15] in the host computing device [11], for example.

At Step S31, the decoder [17] receives a frame of display data across the connection from the host computing device [11] and decodes it. It then begins to write the display data into the first frame buffer [18/FB1] at Step S32. When it has written up to a predetermined place in the frame buffer [18/FB1] at Step S33, the decoder [17] sends a signal to the output engine [19] at Step S34 indicating that it should begin the read process, which begins at Step S35.

This process is described again in more detail in Figures 4 and 5.

Figure 4 shows an example in a display system such as that shown in Figure 1, where

$$T_R = \frac{T_W}{2}$$

as mentioned above.

Since the ratio between  $T_R$  and  $T_W$  is 2:1, the predetermined point mentioned at Step S33 is determined as the point at which the write process has written  $\frac{1}{2}$  of the frame into the buffer [18] – i.e. has written  $\frac{1}{2}$  of the buffer [18], if the buffer is the same size as a complete frame. Accordingly, in Step S41 shown in Figure 4a, the write process writes the first half of a new frame – received in a step corresponding to Step S31 but not shown in Figure 4 – into the first frame buffer [18/FB1]. This is shown by the arrow labelled W in the first part of Figure 4a, as previously described.

If the first frame buffer [18/FB1] were read at this point, the image output would consist of half the new frame (a sun) and half of a previous frame (a cloud). This is an example of tearing. In contrast, in the present example, the read process is delayed; in Figure 4a it is shown to be reading the second frame buffer [18/FB2] which contains a complete preceding frame (a face), but if the frame being  
5 written were the first in the stream the read process might be inactive during Step S41.

When the write process has completed the first half of the frame, a signal is sent to the output engine [19] at Step S34. The output engine [19] then begins to read the first frame buffer [18/FB1] as illustrated at Step S42 by the arrow labelled R. This corresponds to Step S35 in Figure 3.

While the read process is being carried out on the first frame buffer [18/FB1], the write process  
10 continues to write the remainder of the data into the first frame buffer [18/FB1]. This means that by the time the read process has reached the halfway point in the frame buffer [18/FB1], the write process will have reached the three-quarter point and so on, so that the two processes reach the end of the frame together but the read process never overtakes the write process.

When the write process has completed the first frame buffer [18/FB1], it moves to the second  
15 frame buffer [18/FB2] at Step S43 of Figure 4b to begin writing a new frame. This corresponds to a return to Step S32 of Figure 3.

As previously described with regard to the first frame buffer [18/FB1] at Step S41, the write process writes the first half of the new frame (a heart) into the second frame buffer [18/FB2]. Meanwhile, the read process is carried out a second time on the first frame buffer [18/FB1], which is  
20 complete so there can be no tearing. As previously described, when the write process has written the first half of the frame buffer [18/FB2] (Step S33) the decoder [17] sends a signal to the output engine [19] (Step S34) and the output engine [19] begins to read the second frame buffer [18/FB2] at Step S44, corresponding again to step S35 of Figure 3. As previously described, while the read process is being carried out on the second frame buffer [18/FB2] the write process continues and completes the  
25 frame and the read process cannot overtake it because the time the read process begins is determined based on the relative speeds of the two processes.

When the write process has finished writing the data into the second frame buffer [18/FB2], it returns to the first frame buffer [18/FB1] and the process is repeated from Step S41, corresponding once again to Step S32.

30 The signal that triggers the transition of the read process between frame buffers [18/FB1/18/FB2] is described in Figure 4b as being provided by the decoder [17]. However, it could also be an external signal from, for example, the display device [13] indicating that it expects a new frame, a clock signal, or any other suitable external or internal signal. The signal is independent of the progress of the read process through a buffer and therefore if there is a delay in the read process for any reason this delay  
35 is not propagated through the system; the read process can be sent to the beginning of the ready frame

at any time, unlike in conventional systems where the read process must progress through all complete frames in order.

This version of the method will therefore result in an output on the display device [13] such as that shown in Figure 4c: at Step S41, the face is shown. The read process then transitioned to the first frame buffer [18/FB1] and therefore the sun is shown at Step S42. At Step S43 the read process read the first frame buffer [18/FB1] a second time and therefore the sun is shown again; this repetition occurs where the read process is faster than the write process. At Step S44, the read process once again transitioned to the second frame buffer [18/FB2] and therefore the heart is shown. It would be shown again in the next iteration and so forth.

The method is not restricted to a ratio of 2:1 between the rates of the read and write processes and can be extended to an arbitrary ratio. Figure 5 shows a further iteration of the method where

$$T_R = \frac{T_W}{3}$$

This means that the read process is three times faster than the write process. The frame buffer [18] therefore cannot be evenly divided as where the ratio between the rates of the read and write processes is 2:1. However, the predetermined stage can be calculated using the following proportion:

$$Proportion = 1 - 1\left(\frac{1}{3}\right)$$

i.e. 2/3 of the way through the buffer [18]. This could be calculated every time a new display session begins or could be hard-coded, for example at manufacture or into a display driver. In practice, “2/3 of the way through the buffer [18]” could be implemented based on data volume: for example, in a buffer which holds 3GB of data, the read process may begin once 2GB of data have been written:

$$Portion = 3GB - \frac{3GB}{3} = 2GB$$

or location in the frame: for example, if a frame comprises 60 tiles, 40 tiles have been written:

$$Portion = 60 Tiles - \frac{60 Tiles}{3} = 40 Tiles$$

It will also be appreciated that the portion need not necessarily be a contiguous area of memory of the buffer. For example, when the read process is to begin after 40 tiles have been written, the 40 tiles may be any combination of the tiles to be written to the buffer. Moreover, whilst in the above examples the read process has been described as starting once the write process completed writing to a determined proportion of the frame buffer, size of the frame buffer, or number of tiles, the method may instead calculate a time at which the read process is to begin based on the ratio of  $T_R$  to  $T_W$ , such that the read process will not overtake the write process, and the read process may begin at the calculated time irrespective of how much data the write process has actually written to the buffer.

At Step S51, corresponding to Step S32 as previously described, the decoder [17] writes data (a sun) into the first third of the first frame buffer [18/FB1] while the output engine [19] reads the whole of the second frame buffer [18/FB2], in this example containing an existing frame (a face). At Step S52, the decoder [17] continues to write the second third of the frame into the first frame buffer [18/FB1] while the output engine [19] reads the whole of the second frame buffer [18/FB2] again.

At the end of Step S52, two thirds of the frame buffer [18/FB1] have been written and therefore the write process has reached the predetermined stage at Step S33 of Figure 3. The decoder [17] therefore sends a signal to the output engine [19] to begin reading the first frame buffer [18/FB1] at Step S34 and the output engine [19] begins to read the first frame buffer [18/FB1] at Step S53 in Figure 5b, corresponding to Step S35. As previously described, in the time it takes for the read process to complete in the first frame buffer [18/FB1], the write process has written the remainder of the frame into the first frame buffer [18/FB1].

At Step S54, corresponding to a return to Step S32, the decoder [17] begins writing the next frame (a heart) into the second frame buffer [18/FB2] while the output engine [19] reads the first frame buffer [18/FB1] a second time. The output engine [19] reads the first frame buffer [18/FB1] a third time at Step S55 of Figure 5c, while the write process writes the second third of the new frame into the second frame buffer [18/FB2]. At the end of this stage the predetermined point has been reached, corresponding to Step S33, and therefore a signal is sent to the output engine [19] (Step S34) to indicate that it should begin reading the second frame buffer [18/FB2], which it does at Step S56, corresponding to Step S35. During Step S56, the decoder [17] also finishes writing the frame into the second frame buffer [18/FB2], as previously described.

The resulting output is shown in Figure 5d: at Step S51 the initial frame from the second frame buffer [18/FB2], the face, is shown. It persists at Step S52 before the output engine [19] transitioned to reading from the first frame buffer [18/FB1] at S53 and the sun is therefore shown. The output engine [19] continued reading from the first frame buffer [18/FB1] at Steps S54 and S55 while the decoder [17] wrote the first two-thirds of the frame into the second frame buffer [18/FB2], and then the output engine [19] transitioned to reading from the second frame buffer [18/FB2] and the heart is displayed at Step S56. As before, this results in repetition, because of the nature of a system where the read process is faster than the write process.

The above description assumes that the entire frame is updated. However, in some systems only part of the data may be updated at a time, for example only parts of a frame of display data that have changed compared to a previous frame, known sometimes as “dirty rectangles”. This means that the volume and location of data written to the frame buffer may change from frame to frame. While the write process could still scan down the frame buffer in the conventional way and only make updates in the appropriate locations, acting with regard to the read process as described above, it may be more

efficient for the write process to only access and update the appropriate locations and for the read process to begin as soon as possible for this limited write, as opposed to the full write.

In many conventional systems, the read process will still read the full buffer regardless of which parts have been updated and therefore  $T_R$  will be fixed. However, the time required to write the updates to the buffer will depend on how much of the buffer has been updated. This information is known to the write process and can be used to determine when to begin the read process.

For example, if the buffer is split into  $N$  rows of data to be read in a range from 0 to  $N-1$ , read in order, the time at which a given row  $n$  is read can be described as  $t_{r,n}$ , which can be determined by a formula:

$$t_{r,n} = t_{r,0} + \frac{(n+1)T_R}{N}$$

I.e, the time at which row  $n$  is read is the time at which the read process started plus a time offset based on the value of  $n$  and the time required to read the whole buffer, divided by the number of rows in total.

Assuming that an updated part consists of  $M$  rows of data to be written in a range from 0 to  $M-1$ , written in order, the time at which the write function has completed a given row  $m$  can be described as  $t_{w,m}$ , which can be determined by a similar formula:

$$t_{w,m} = t_{w,0} + \frac{(m+1)T_W}{N}$$

I.e, the time at which the write process finishes writing row  $m$  is the time at which the write process started plus a time offset based on the value of  $m$  and the time required to write the whole buffer, divided by the number of rows in total.

The  $M$  rows in the updated part may in fact be split into multiple updated parts, and even a single updated part may not begin at the beginning of the buffer. Accordingly, it is necessary to map each row  $m$  to a row  $n$  in the buffer depending on the locations of the rows in the buffer:

$$n = f(m)$$

I.e. each updated row in the updated part is associated with a row in the buffer.

The time at which the read process begins is then selected such that, as an approximation:

$$t_{r,0} + \frac{(f(m)+1)T_R}{N} \geq t_{w,m}$$

So in an example where:

- $m = 0$
- $f(m) = n = 3$

- $N = 8$
- $T_R = x$
- $T_W = 2x$

$$t_{r,0} + \frac{(3 + 1)x}{8} \geq t_{w,0} + \frac{(0 + 1)2x}{8}$$

5 Thus

$$t_{r,0} + \frac{x}{2} \geq t_{w,0} + \frac{x}{4}$$

Naturally, this inequality must be determined for every value of  $m$  in the updated part or all updated parts to produce a final value.

The time at which the read process begins can then be selected in order to preserve this inequality; there need no longer be a precise ratio maintained between the size of the portion of the buffer that has been updated and the time at which the read process begins. This calculation may be carried out on a per-frame basis by the device that determines the time at which the read process begins as previously described – in the examples above, the decoder [17].

Accordingly, under these circumstances, the Portion used in the methods may be reduced if it is known that the write process is not writing the entire frame; for example, if it were known that only the second half of a frame in the system described in Figure 4 were being updated, the read process could begin at the same time as the write process as the case would essentially be those shown in Steps S42 and S44. The use of this method may reduce predictability of the system, but it may also reduce the latency caused by double-buffering and allow the system to be efficiently used with partial updates.

Fig. 6 is a block diagram of a computer system [600] suitable for implementing one or more embodiments of the present disclosure, including the host device [11], or the display control device [12]. In various implementations, the host device [11] may include a mobile cellular phone, personal computer (PC), laptop, wearable computing device, etc. adapted for wireless communication, and each of the host device [11] and the display control device [12] may include a network computing device. Thus, it should be appreciated that these devices [12, 13] may be implemented as the computer system [600] in a manner as follows.

The computer system [600] includes a bus [612] or other communication mechanism for communicating information data, signals, and information between various components of the computer system [600]. The components include an input/output (I/O) component [604] that processes a user action, such as selecting keys from a keypad/keyboard, selecting one or more buttons or links, etc., and sends a corresponding signal to the bus [612]. The I/O component [604] may also include an output component, such as a display [602] and a cursor control [608] (such as a keyboard,

keypad, mouse, etc.). An optional audio input/output component [606] may also be included to allow a user to use voice for inputting information by converting audio signals. The audio I/O component [606] may allow the user to hear audio. A transceiver or network interface [620] transmits and receives signals between the computer system [600] and other devices via network [622]. In one  
5 embodiment, the transmission is wireless, although other transmission mediums and methods may also be suitable. A processor [614], which can be a micro-controller, digital signal processor (DSP), or other processing component, processes these various signals, such as for display on the computer system [600] or transmission to other devices via a communication link [624]. The processor [614] may also control transmission of information, such as cookies or IP addresses, to other devices.

10 The components of the computer system [600] also include a system memory component [610] (e.g., RAM), a static storage component [616] (e.g., ROM), and/or a disk drive [618] (e.g., a solid-state drive, a hard drive). The computer system [600] performs specific operations by the processor [614] and other components by executing one or more sequences of instructions contained in the system memory component [610]. For example, the processor [614] could be utilised to perform the  
15 functions of the application [14] of the host device [11].

Executable logic for performing any described functions may be encoded in a computer readable medium, which may refer to any medium that participates in providing instructions to the processor [614] for execution. Such a medium may take many forms, including but not limited to, non-volatile media, volatile media, and transmission media. In various implementations, non-volatile media  
20 includes optical or magnetic disks, volatile media includes dynamic memory, such as the system memory component [610], and transmission media includes coaxial cables, copper wire, and fiber optics, including wires that comprise the bus [612]. In one embodiment, the logic is encoded in non-transitory computer readable medium, such as a magnetic or optical disk or other magnetic/optical storage medium, or FLASH or other solid-state memory (e.g. integrated into a device or in the form of  
25 a memory card). In one example, transmission media may take the form of acoustic or light waves, such as those generated during radio wave, optical, and infrared data communications.

In various embodiments of the present disclosure, execution of instruction sequences to practice the present disclosure may be performed by the computer system [600]. In various other embodiments of the present disclosure, a plurality of computer systems [600] coupled by the communication link  
30 [624] to the network (e.g., such as a LAN, WLAN, PTSN, and/or various other wired or wireless networks, including telecommunications, mobile, and cellular phone networks) may perform instruction sequences to practice the present disclosure in coordination with one another.

The above embodiments and examples are to be understood as illustrative examples. Where applicable, various embodiments provided by the present disclosure may be implemented using  
35 hardware, software, or combinations of hardware and software. Also, where applicable, the various



hardware components and/or software components set forth herein may be combined into composite components comprising software, hardware, and/or both without departing from the spirit of the present disclosure. Where applicable, the various hardware components and/or software components set forth herein may be separated into sub-components comprising software, hardware, or both  
5 without departing from the scope of the present disclosure. In addition, where applicable, it is contemplated that software components may be implemented as hardware components and vice-versa.

Software in accordance with the present disclosure, such as program code and/or data, may be stored on one or more computer readable mediums. It is also contemplated that software identified herein may be implemented using one or more general purpose or specific purpose computers and/or  
10 computer systems, networked and/or otherwise. Where applicable, the ordering of various steps described herein may be changed, combined into composite steps, and/or separated into sub-steps to provide features described herein.

The various features and steps described herein may be implemented as systems comprising one or more memories storing various information described herein and one or more processors coupled to  
15 the one or more memories and a network, wherein the one or more processors are operable to perform steps as described herein, as non-transitory machine-readable medium comprising a plurality of machine-readable instructions which, when executed by one or more processors, are adapted to cause the one or more processors to perform a method comprising steps described herein, and methods performed by one or more devices, such as a hardware processor, user device, server, and other  
20 devices described herein.

**Claims**

1. A method of writing data to, and reading data from, one or more buffers, the method comprising:
  - determining a write rate of writing data into a first buffer;
  - determining a read rate of reading data from the first buffer;
  - determining, using the write rate and the read rate, a portion of the first buffer;
  - writing data into the portion of the first buffer;
  - starting to read data from the first buffer when the writing of data to the portion of the first buffer has finished; and
  - writing data into a remaining part of the first buffer, different from the portion of the first buffer;
  - wherein the portion of the first buffer is determined such that the reading of data from the first buffer does not overtake the writing of data into the first buffer.
2. The method according to claim 1, wherein the first buffer is a first-in first-out, FIFO, buffer.
3. The method according to claim 1 or 2, the method further comprising:
  - determining a write rate of writing data into a second buffer;
  - determining a read rate of reading data from the second buffer;
  - determining, using the write rate and the read rate, a portion of the second buffer;
  - writing data into the portion of the second buffer when the writing of data to the remaining part of the first buffer has finished;
  - continuing to read data from the first buffer whilst data is written to the portion of the second buffer;
  - starting to read data from the second buffer when the writing of data to the portion of the second buffer is finished; and
  - writing data into a remaining part of the second buffer, different from the portion of the second buffer;
  - wherein the portion of the second buffer is determined such that the reading of data from the second buffer does not overtake the writing of data into the second buffer.
4. The method according to claim 3, wherein the second buffer is a first-in first-out, FIFO, buffer.

5. The method according to claim 3 or 4, wherein the step of determining the portion of the second buffer is based on a ratio of the read rate of reading data from the second buffer and the write rate of writing data to the second buffer.
6. The method according to any preceding claim, wherein the step of determining the portion of the first buffer is based on a ratio of the read rate of reading data from the first buffer and the write rate of writing data to the first buffer.
7. The method according to any preceding claim, wherein the portion of the first buffer is calculated using the formula:

$$P = B \left(1 - \frac{1}{R}\right)$$

where P is the portion, B is the size of the first buffer, and R is the read rate of reading data from the first buffer divided by the write rate of writing data to the first buffer.

8. The method according to claim 6, wherein the portion of the first buffer is calculated using the formula:

$$P = B \left(1 - \frac{1}{R}\right) + C$$

where P is the portion, B is the buffer size of the first buffer, R is the read rate of reading data from the first buffer divided by the write rate of writing data to the first buffer, and C is a constant.

9. The method according to any preceding claim, wherein only updated parts of the first buffer are written to.
10. A method of writing data to, and reading data from, a buffer, the method comprising:
  - determining a write rate of writing data into the buffer;
  - determining a read rate of reading data from the buffer;
  - writing data into the buffer;
  - determining, using the write rate and the read rate, a time at which data is to begin to be read from the buffer, wherein the time is determined such that the reading of data from the buffer will not overtake the writing of data into the buffer; and
  - starting to read data from the buffer at the determined time.
11. The method according to claim 10, wherein the buffer is a first-in first-out, FIFO, buffer.
12. The method according to claim 10, wherein
  - only updated parts of the buffer are written to; and

the time at which data is to begin to be read from the buffer is calculated based on the time at which the writing of the updated parts to the buffer will be complete.

13. The method according to claim 12, wherein all of the data in the buffer is read.
14. The method according to any preceding claim, wherein the data is display data.
15. The method according to claim 14, wherein the data comprises tiles of display data.
16. A computing device comprising a plurality of frame buffers, wherein the computing device is configured to perform the method of any of claims 1 to 15.

17. A system comprising:

a non-transitory memory storing instructions; and

one or more hardware processors coupled to the non-transitory memory and configured to execute the instructions from the non-transitory memory to cause the system to perform operations comprising:

determining a write rate of writing data into a first buffer;

determining a read rate of reading data from the first buffer;

determining, using the write rate and the read rate, a portion of the first buffer;

writing data into the portion of the first buffer;

starting to read data from the first buffer when the writing of data to the portion of the first buffer has finished; and

writing data into a remaining part of the first buffer, different from the portion of the first buffer;

wherein the portion of the first buffer is determined such that the reading of data from the first buffer does not overtake the writing of data into the first buffer.

18. A system comprising:

a non-transitory memory storing instructions; and

one or more hardware processors coupled to the non-transitory memory and configured to execute the instructions from the non-transitory memory to cause the system to perform operations comprising:

determining a write rate of writing data into a buffer;

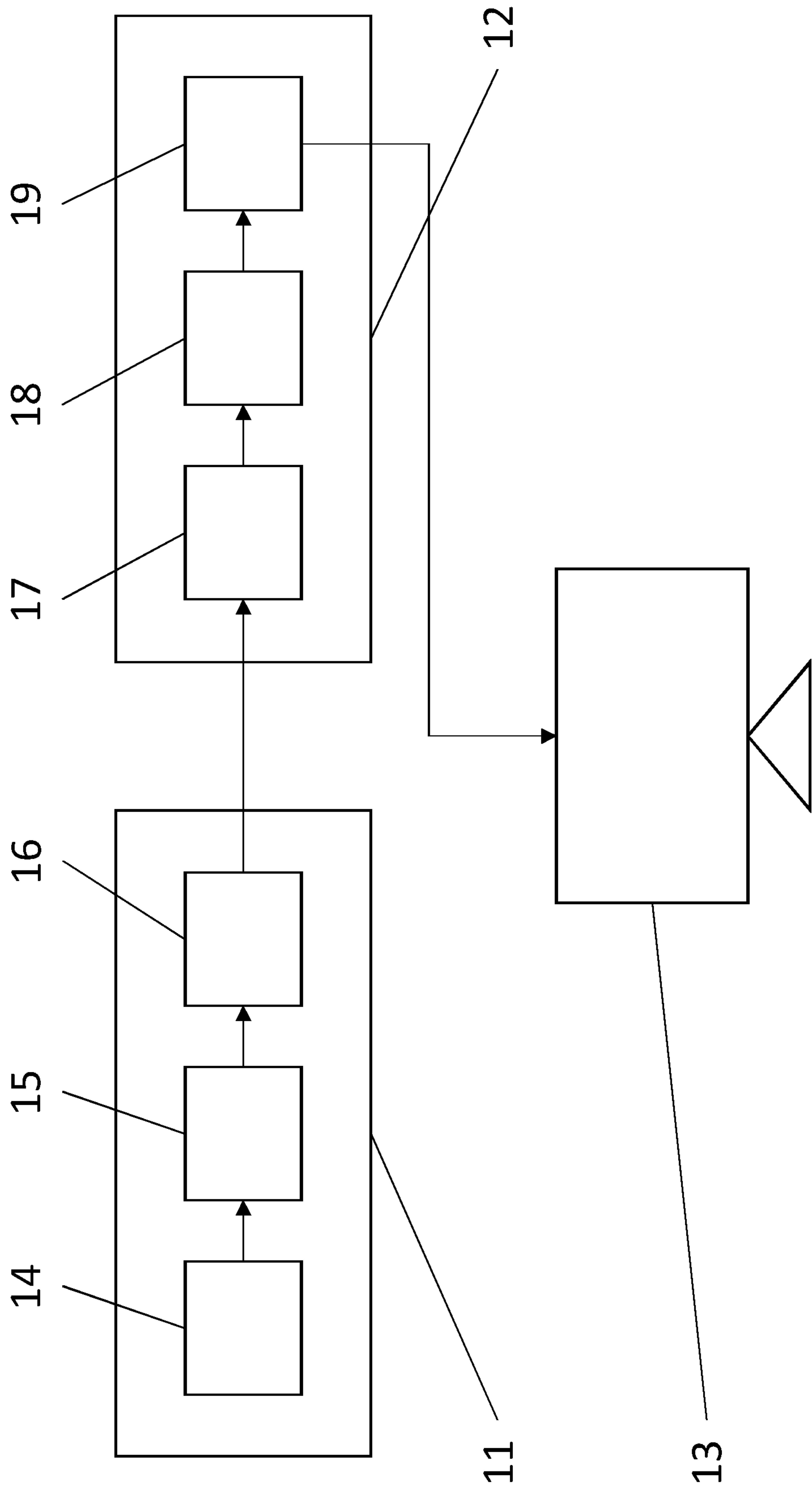
determining a read rate of reading data from the buffer;

writing data into the buffer;

determining, using the write rate and the read rate, a time at which data is to begin to be read from the buffer, wherein the time is determined such that the reading of data from the buffer will not overtake the writing of data into the first buffer; and

starting to read data from the first buffer at the determined time.

Figure 1



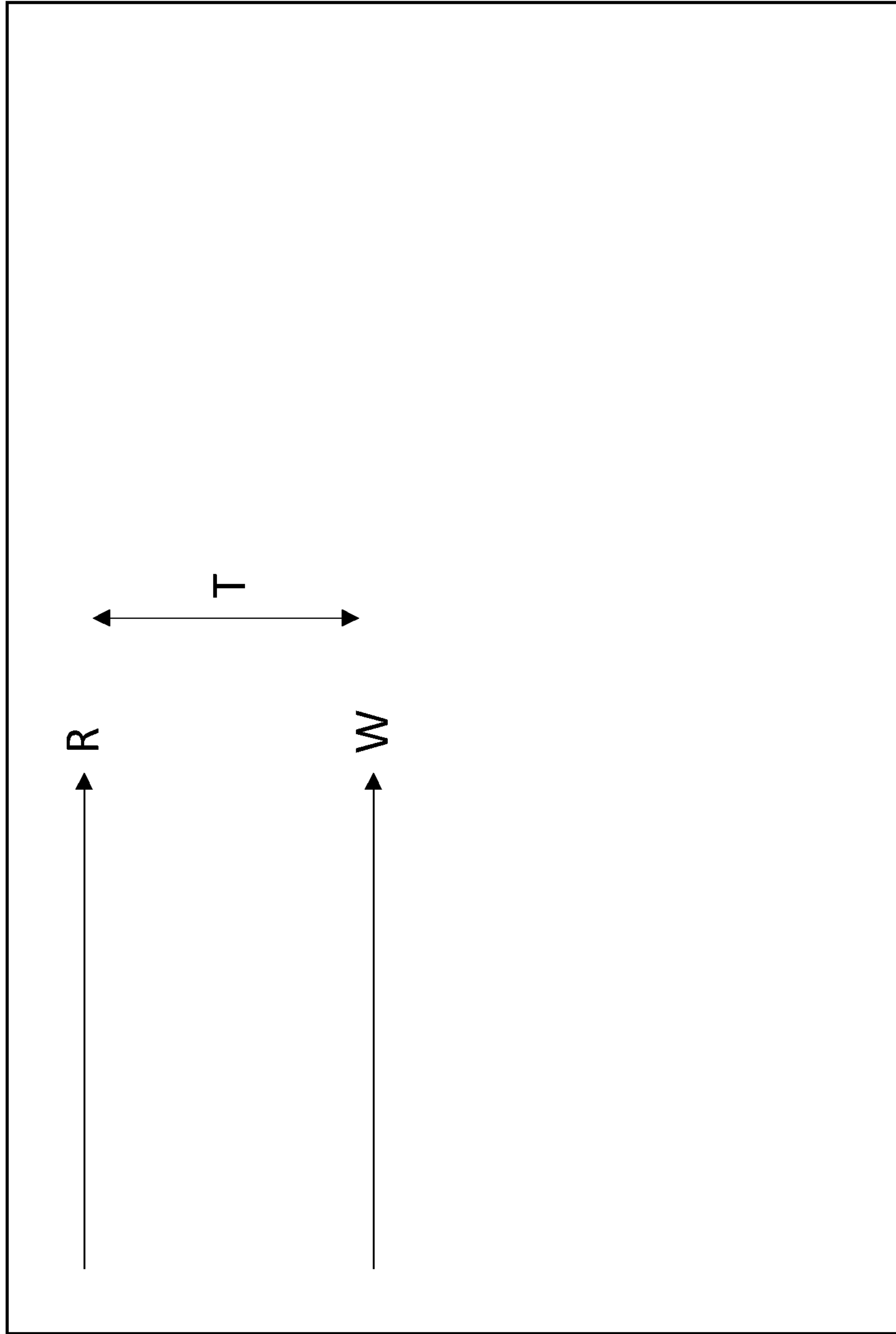


Figure 2

Figure 3

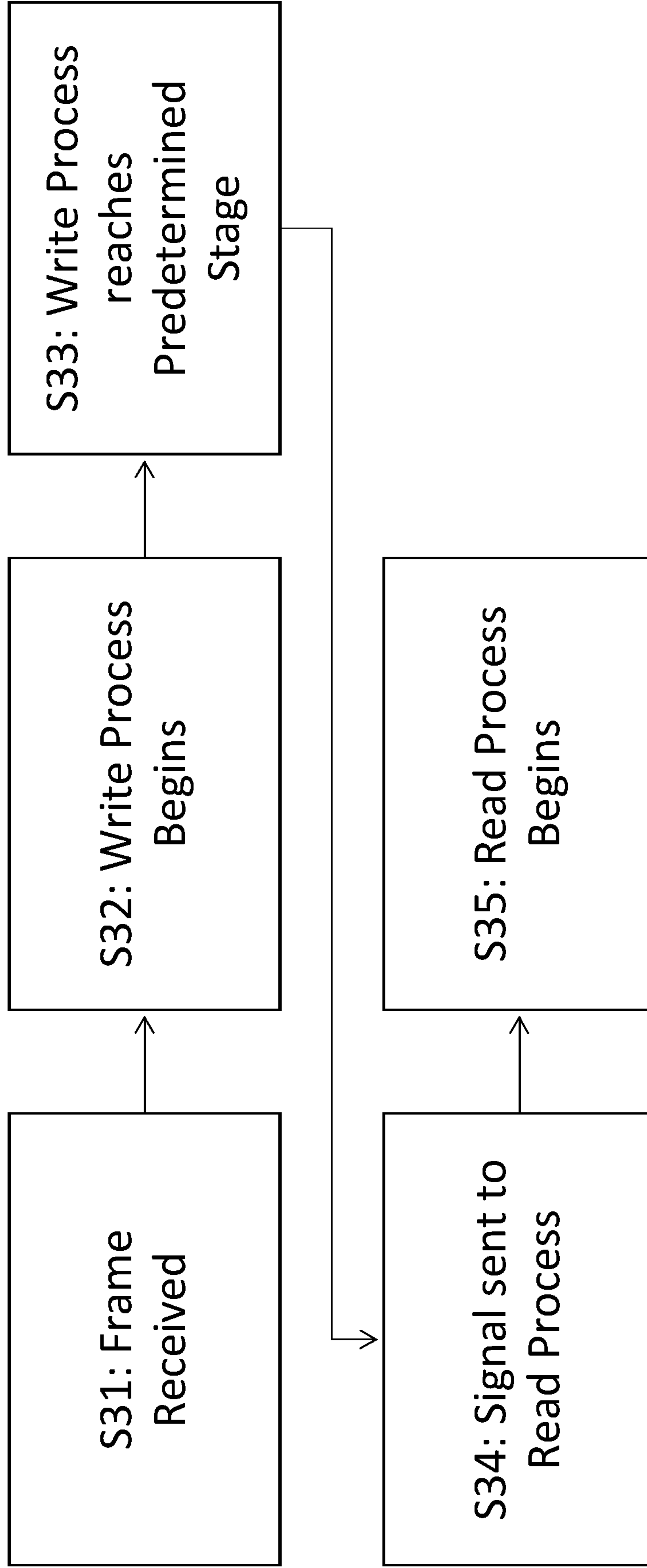
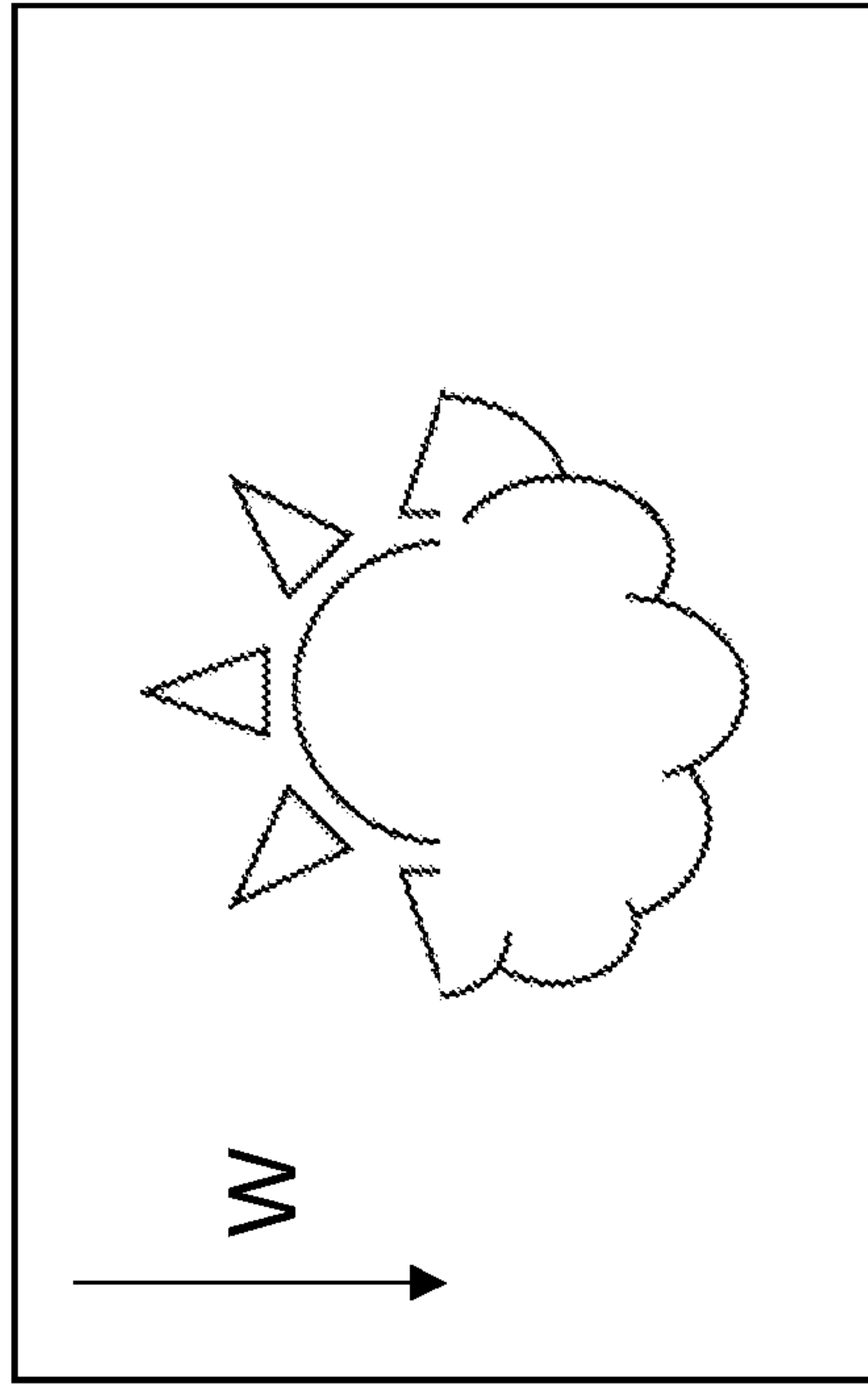




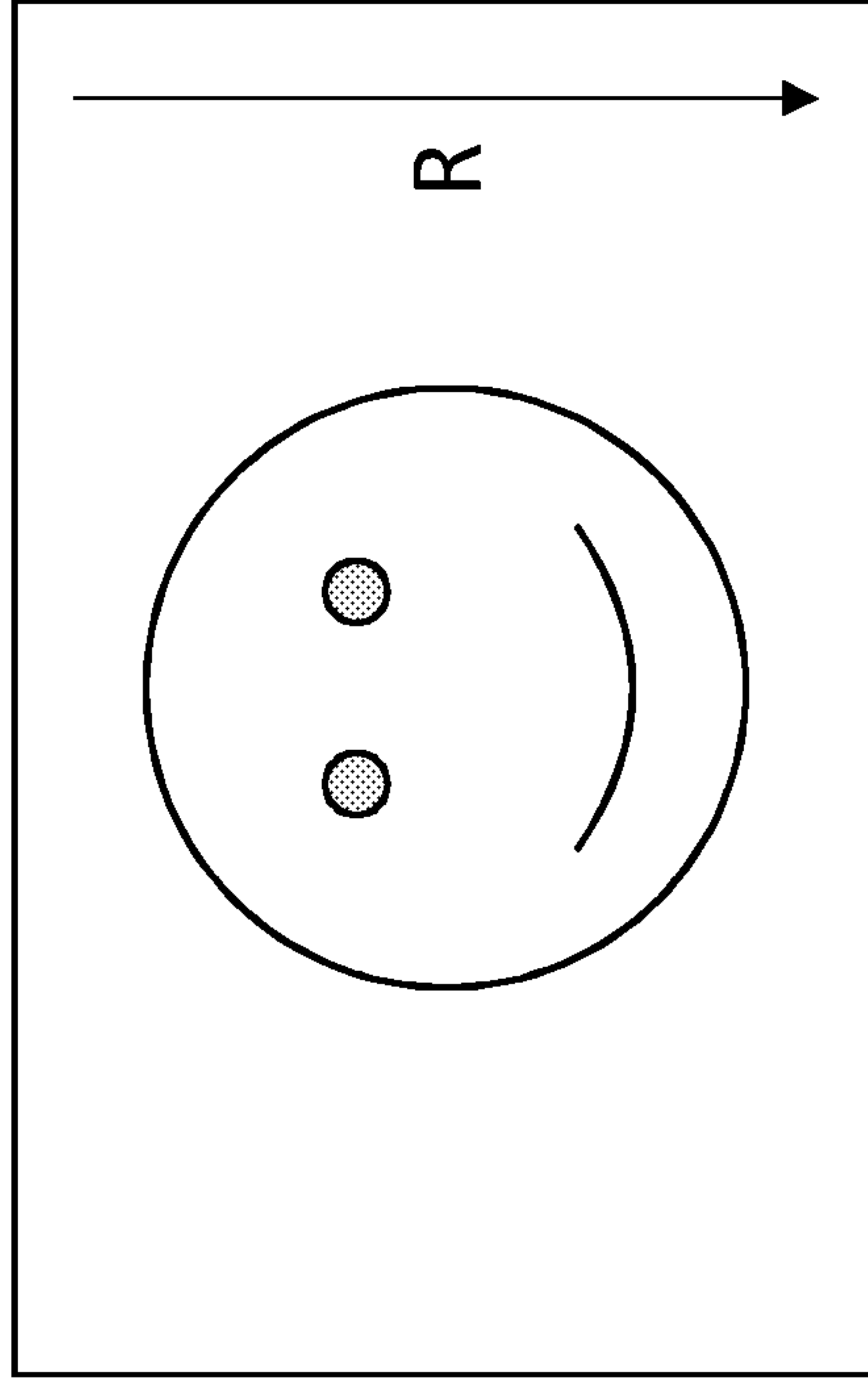
Figure 4a

FB1



S41

FB2



S42

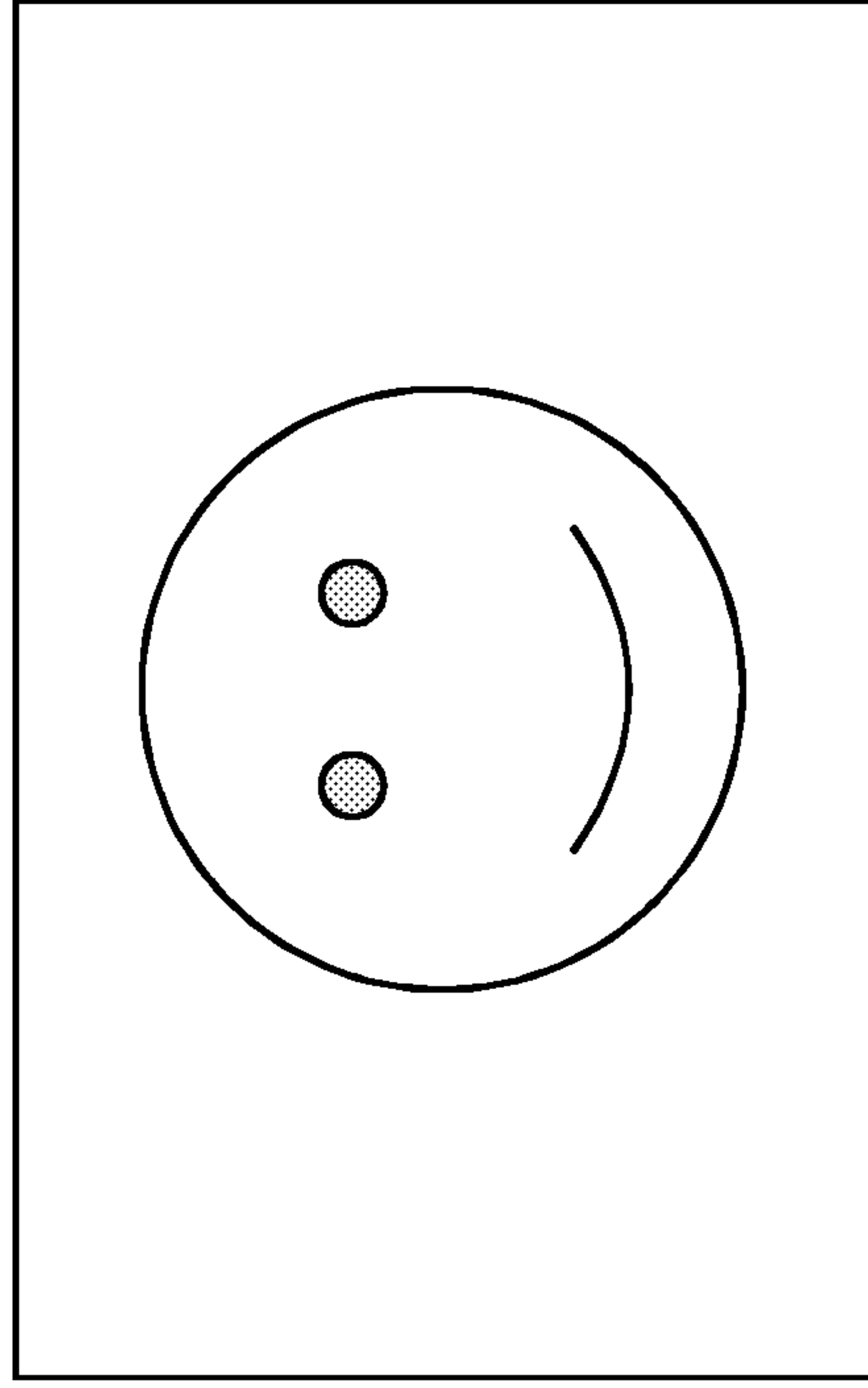
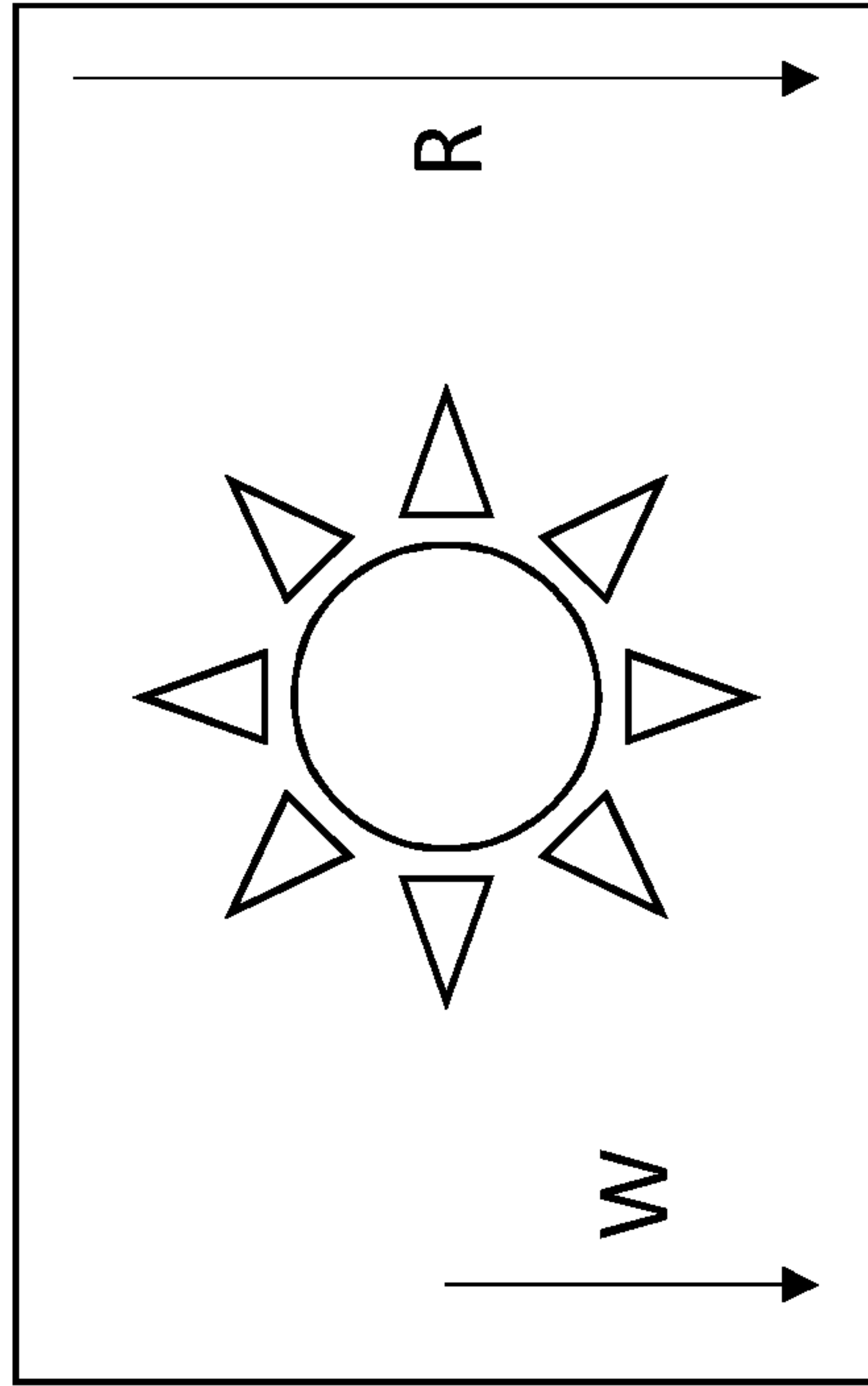
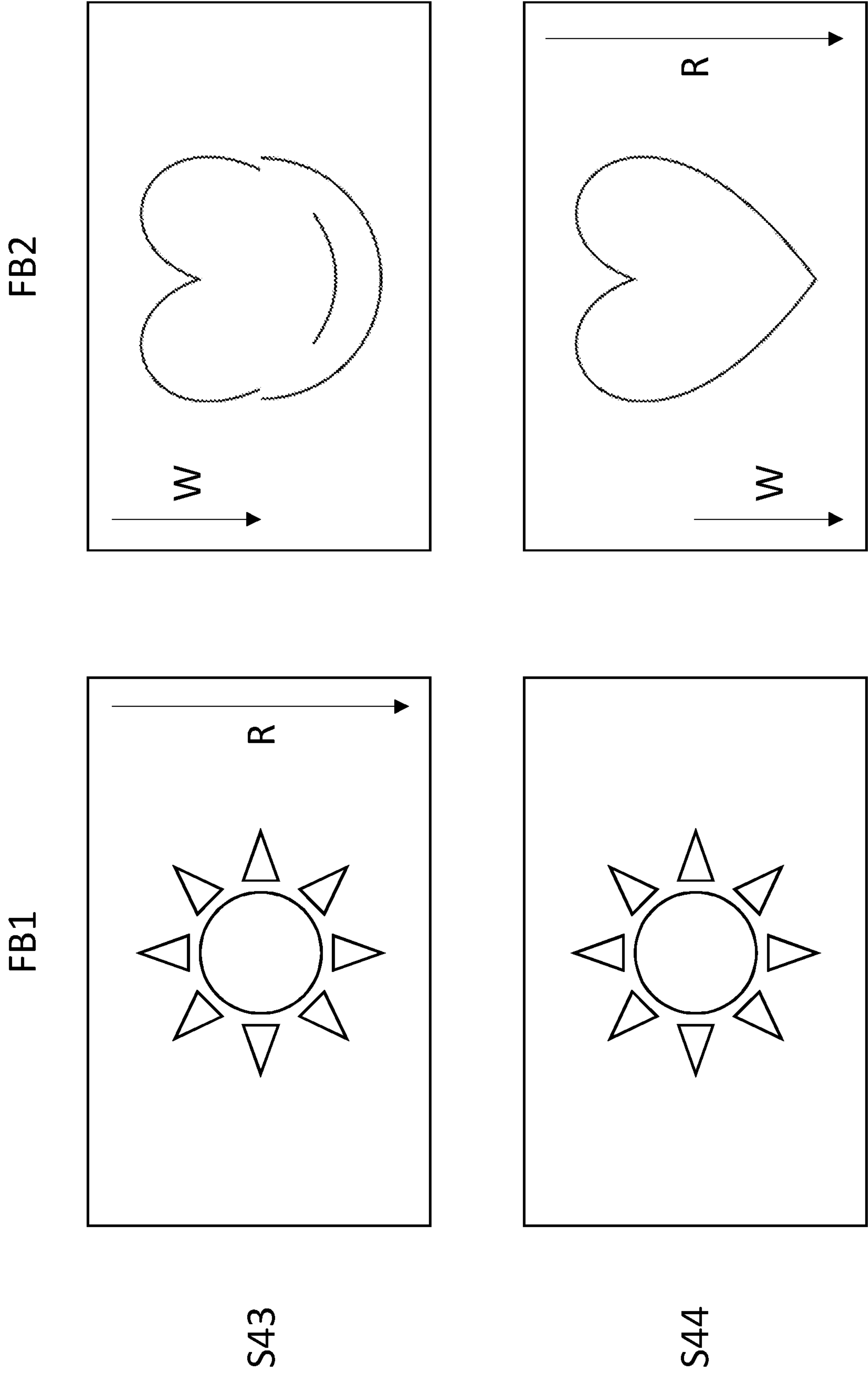


Figure 4b



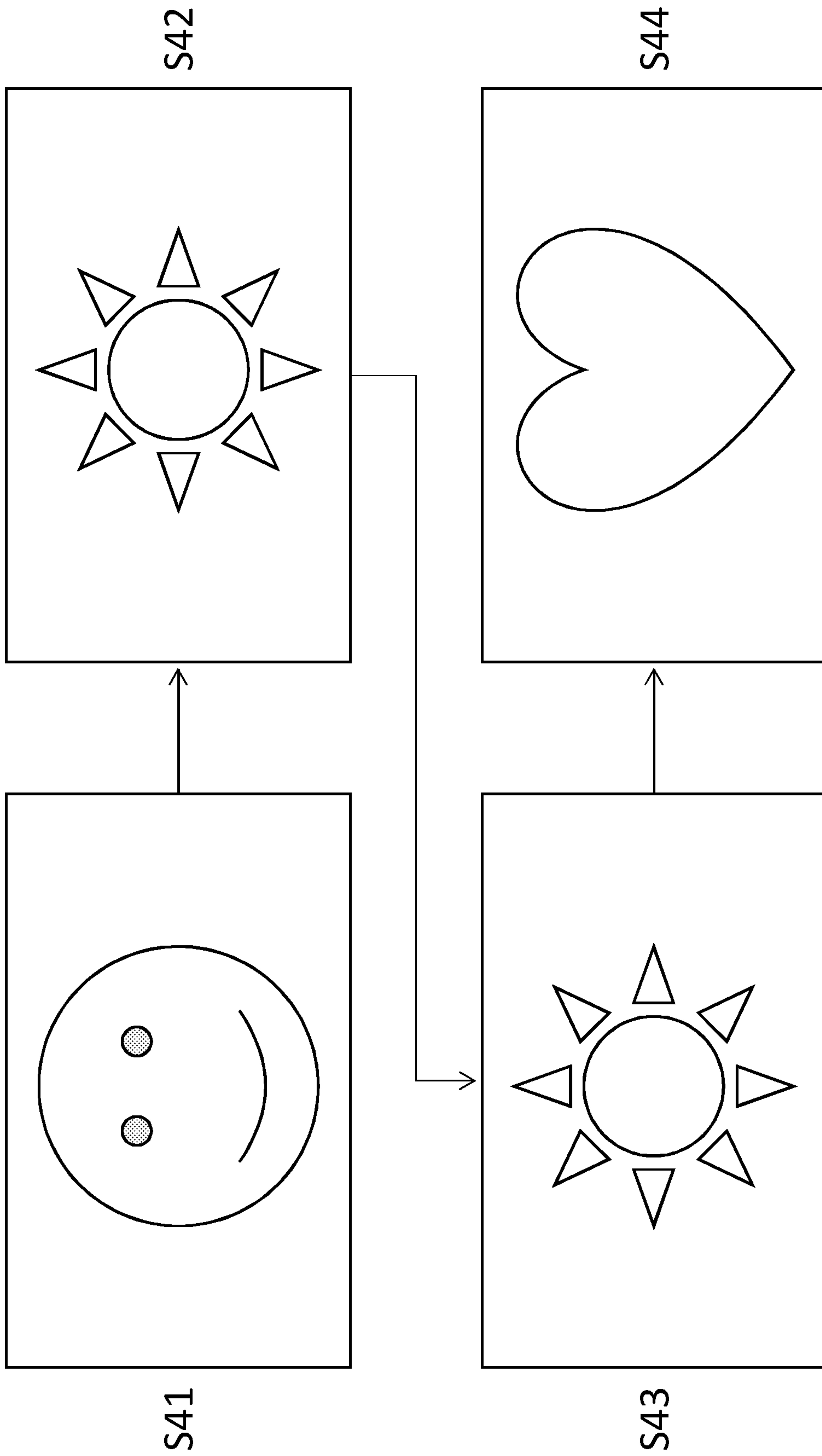
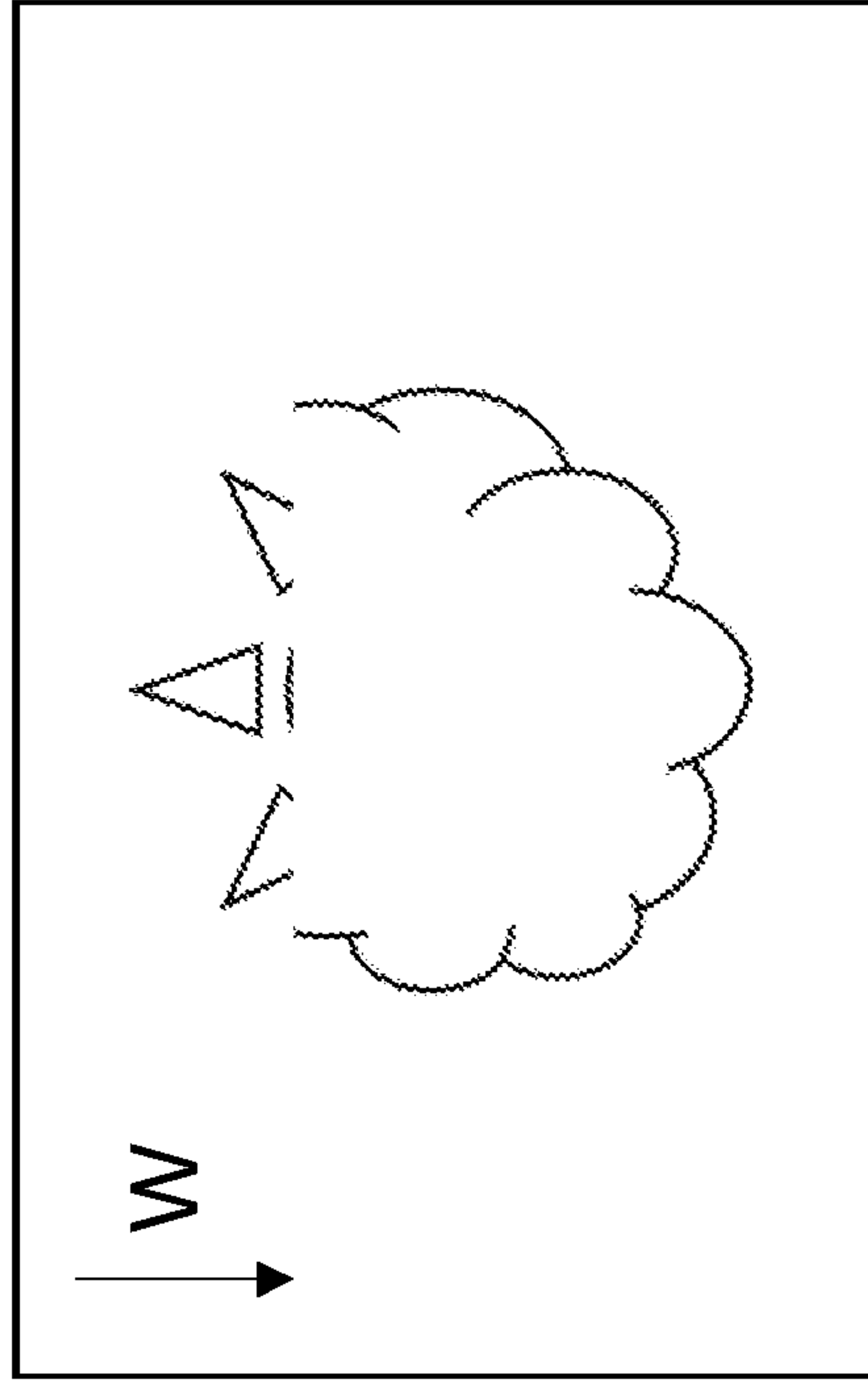


Figure 4c

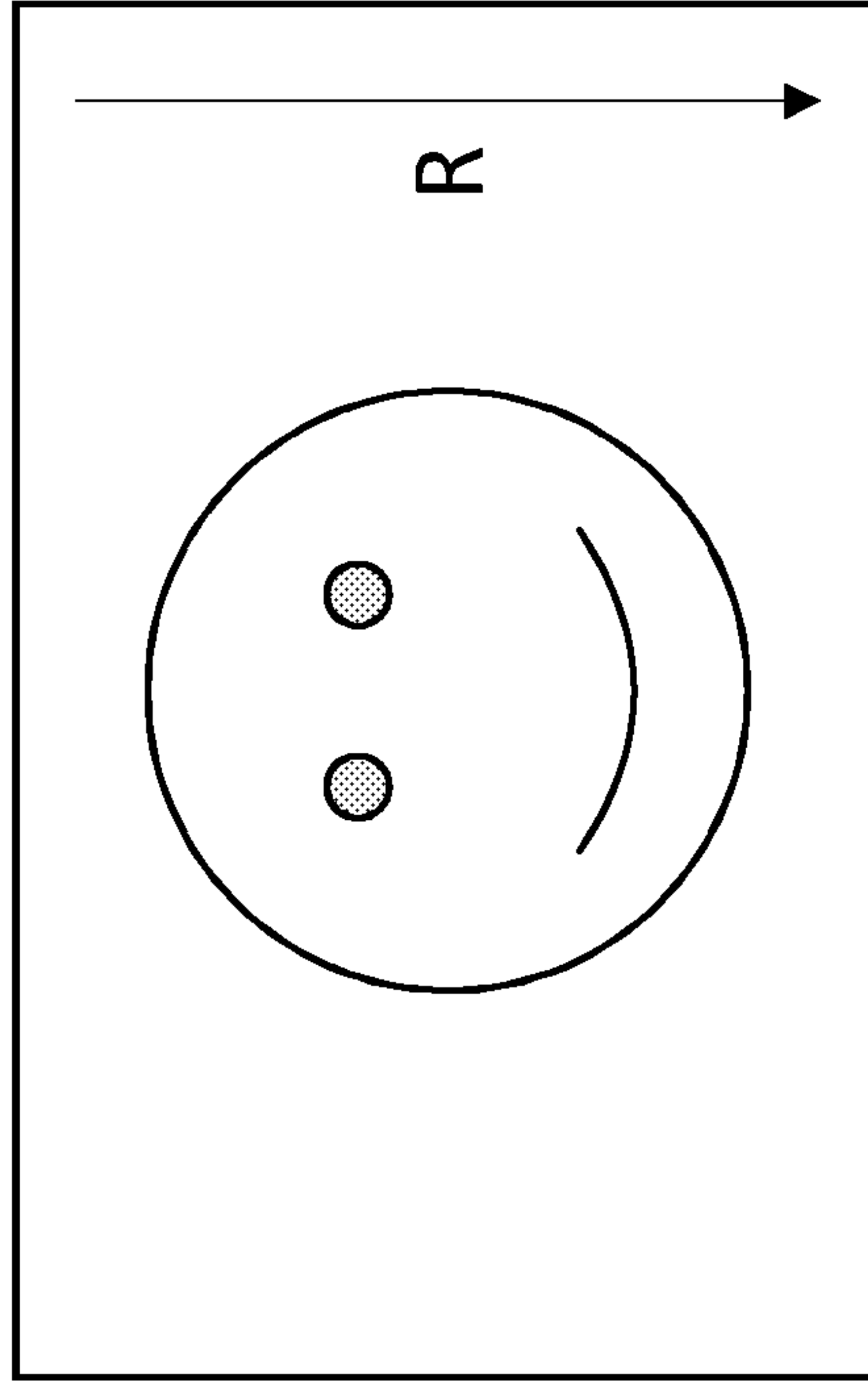
Figure 5a

FB1



S51

FB2



S52

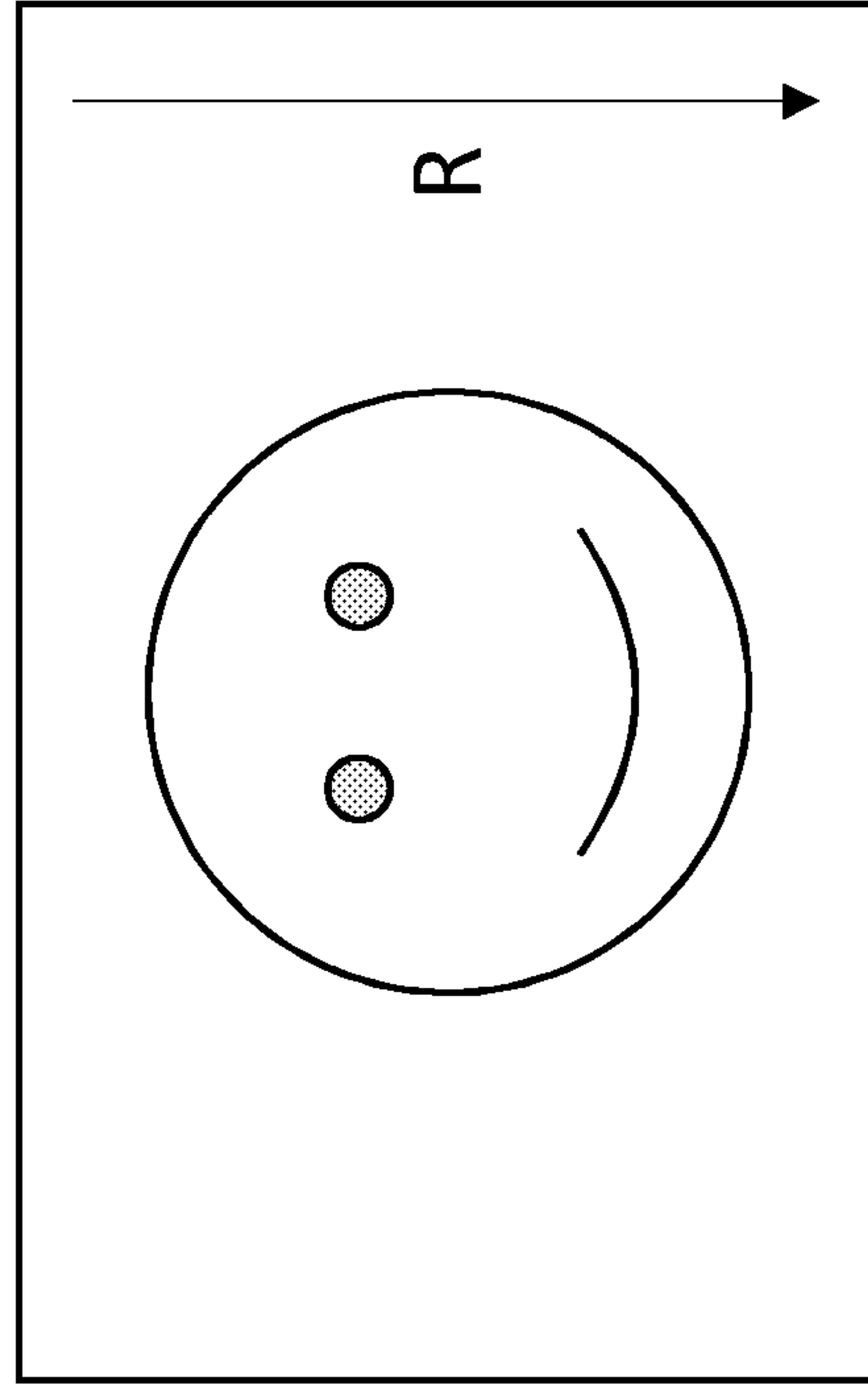
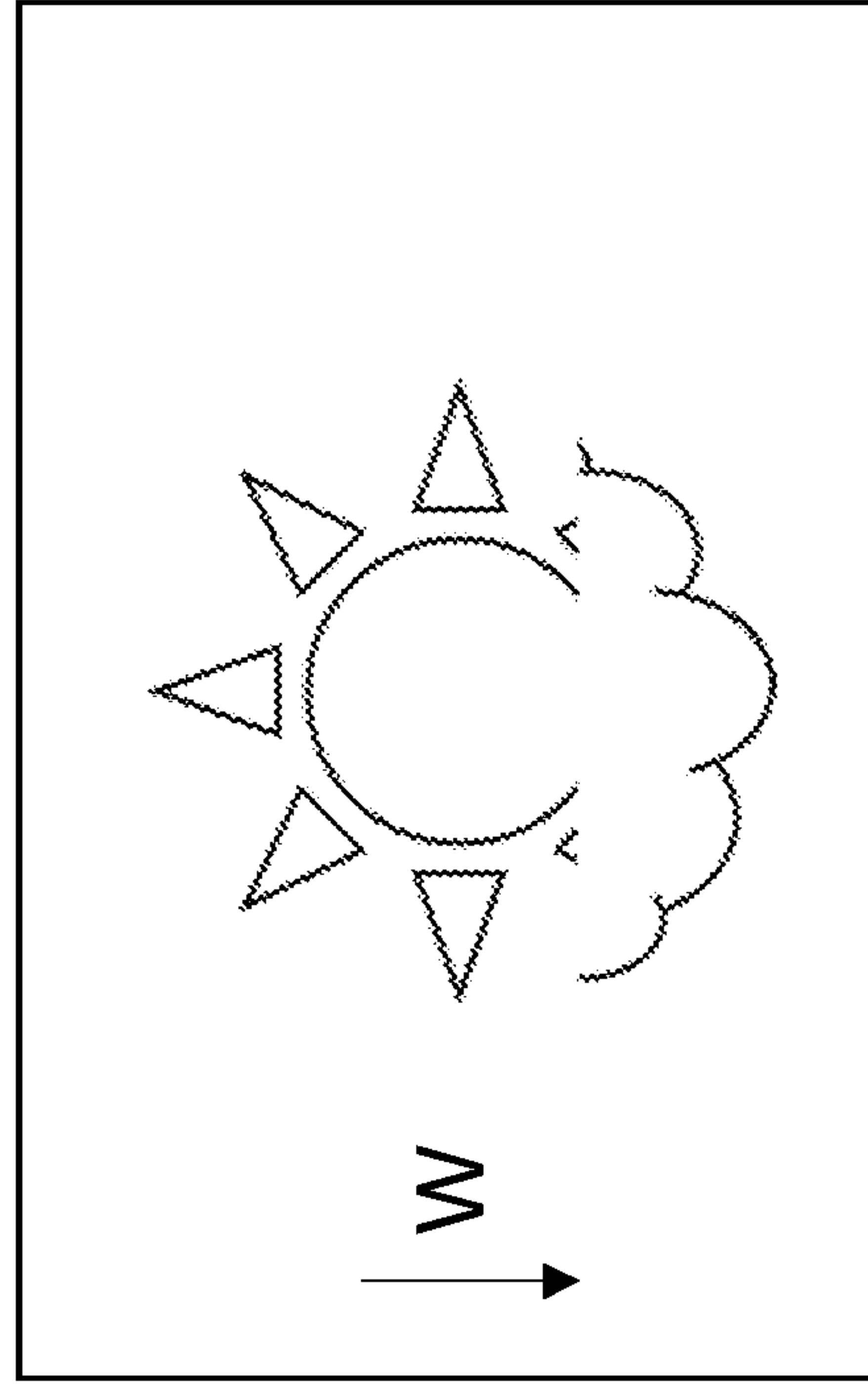
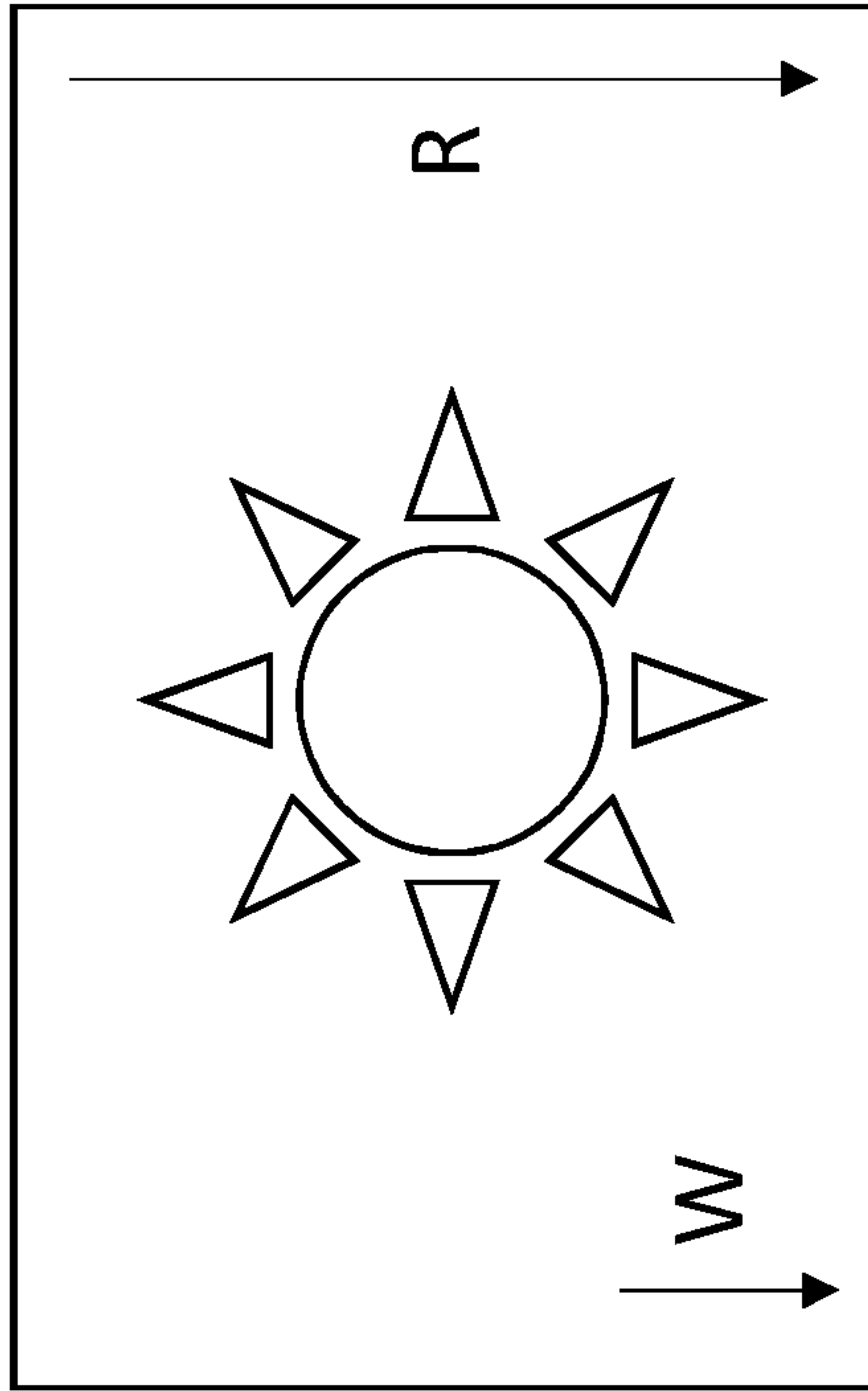


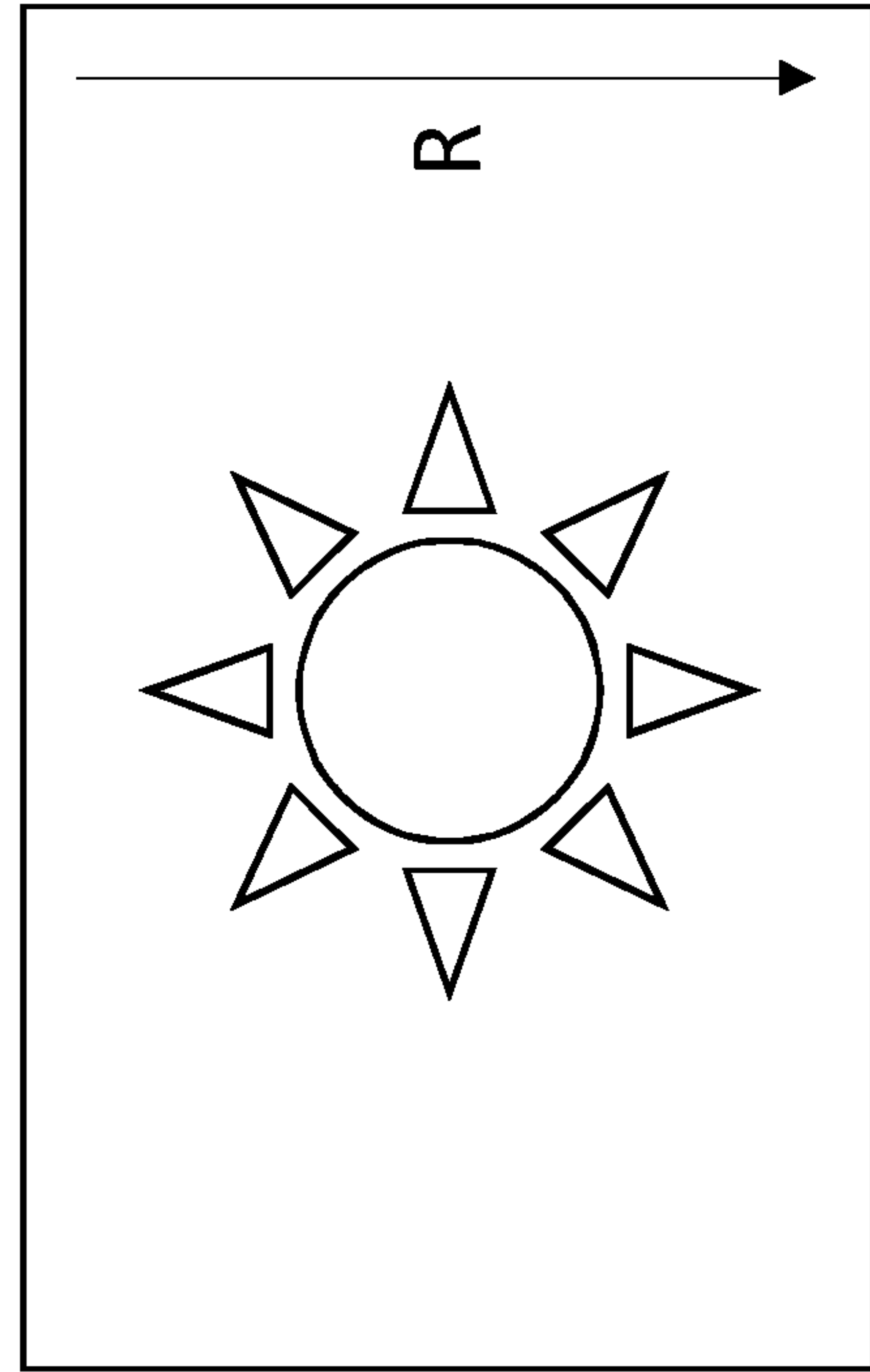
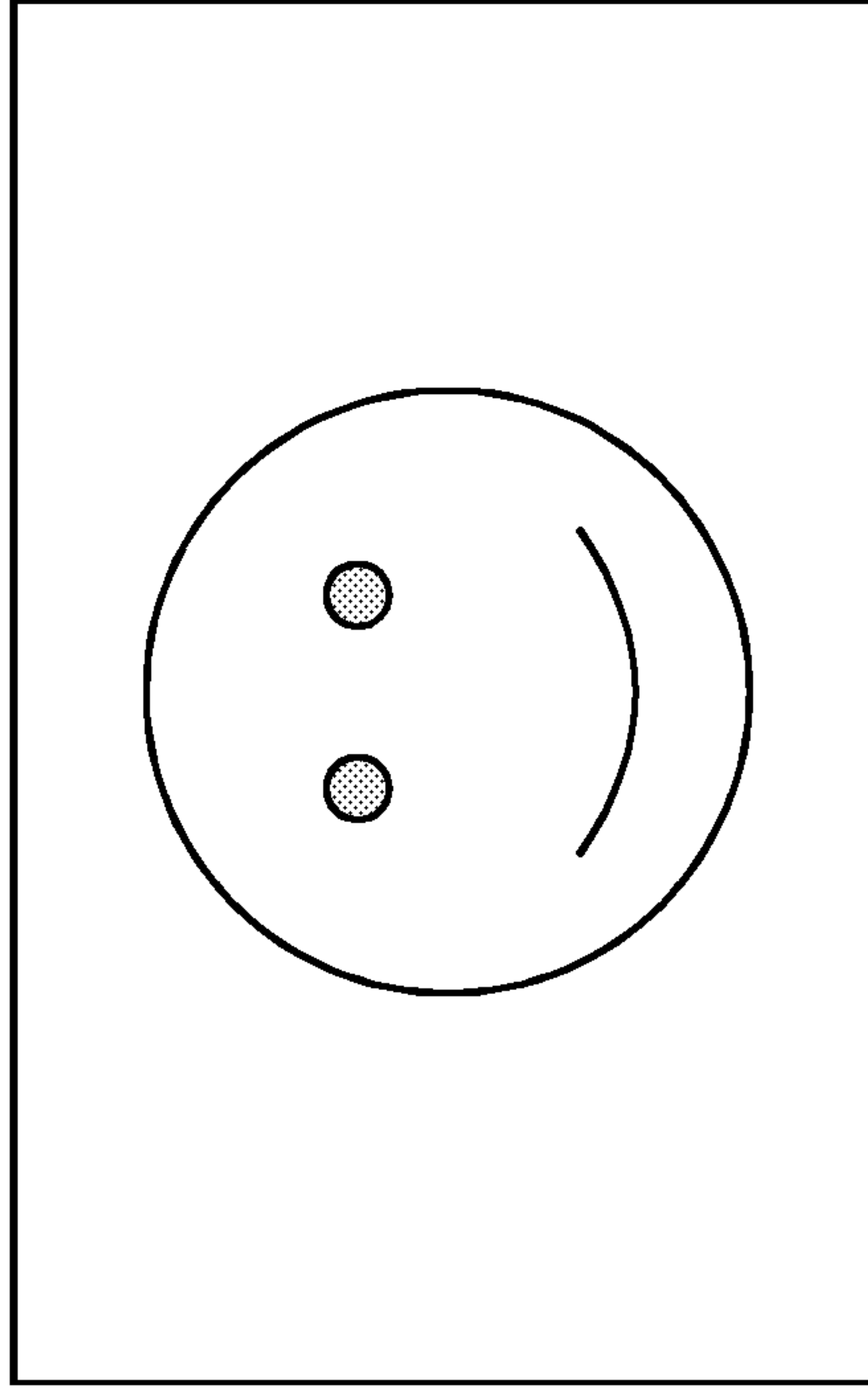
Figure 5b

FB1



S53

FB2



S54

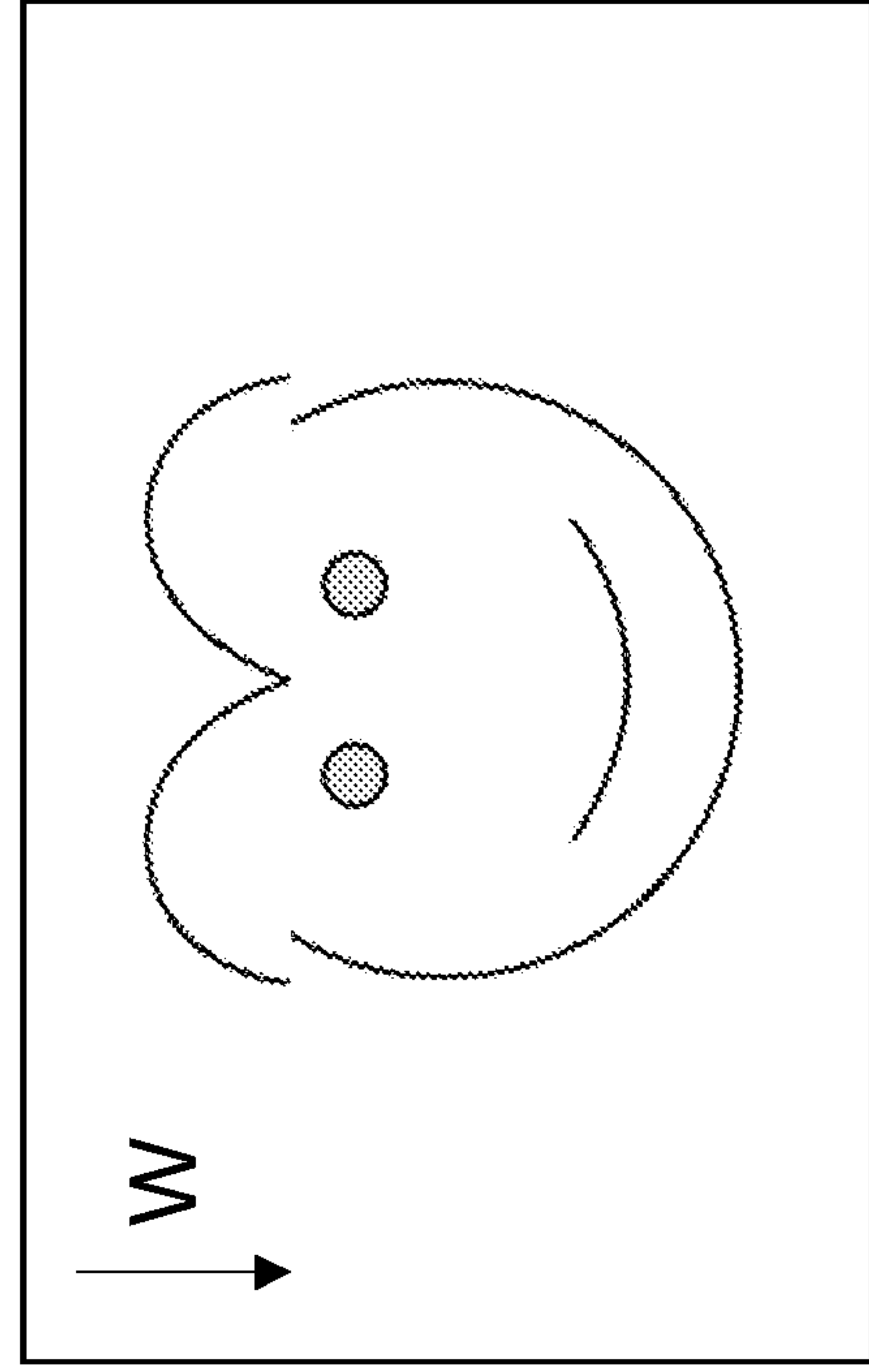
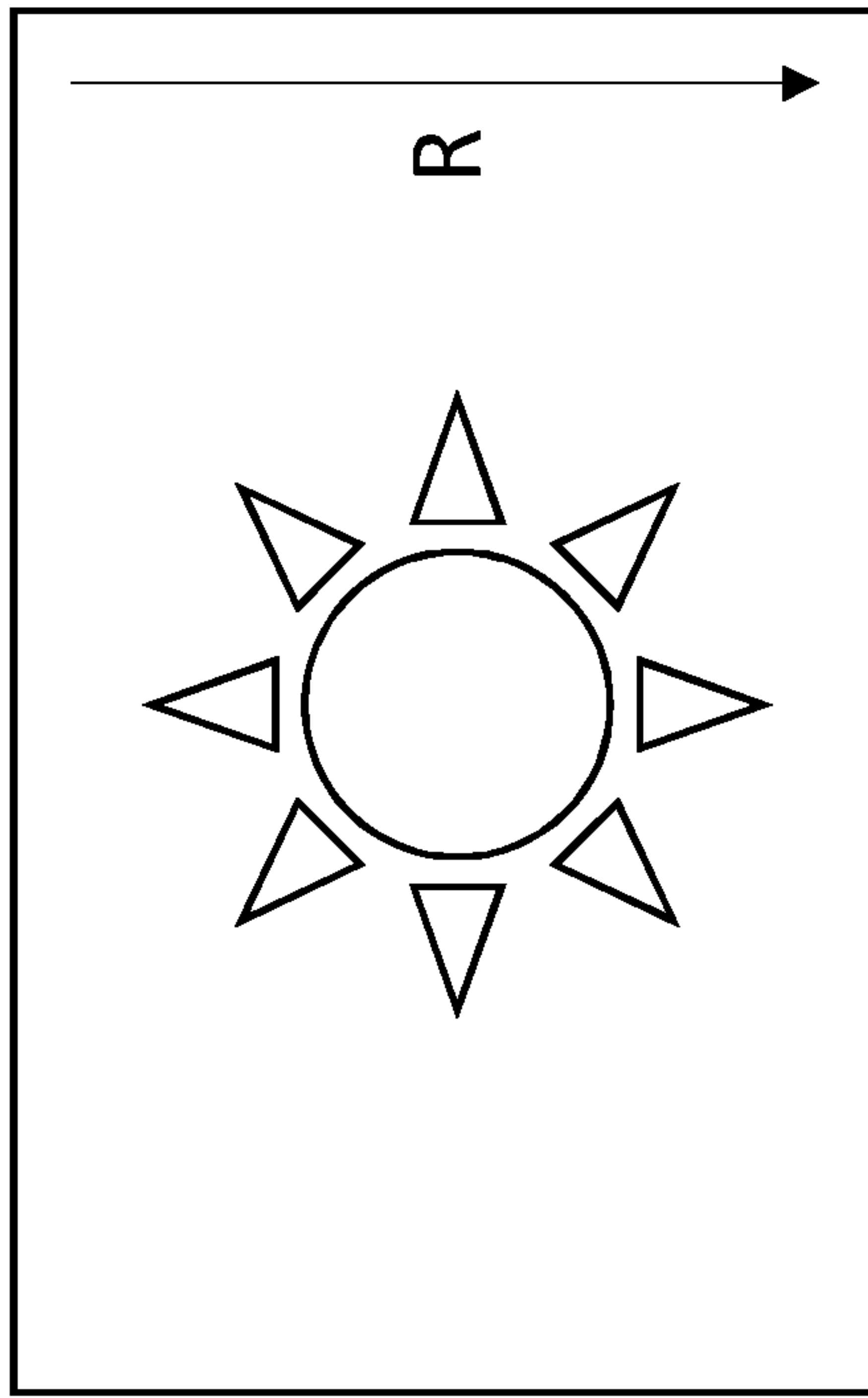


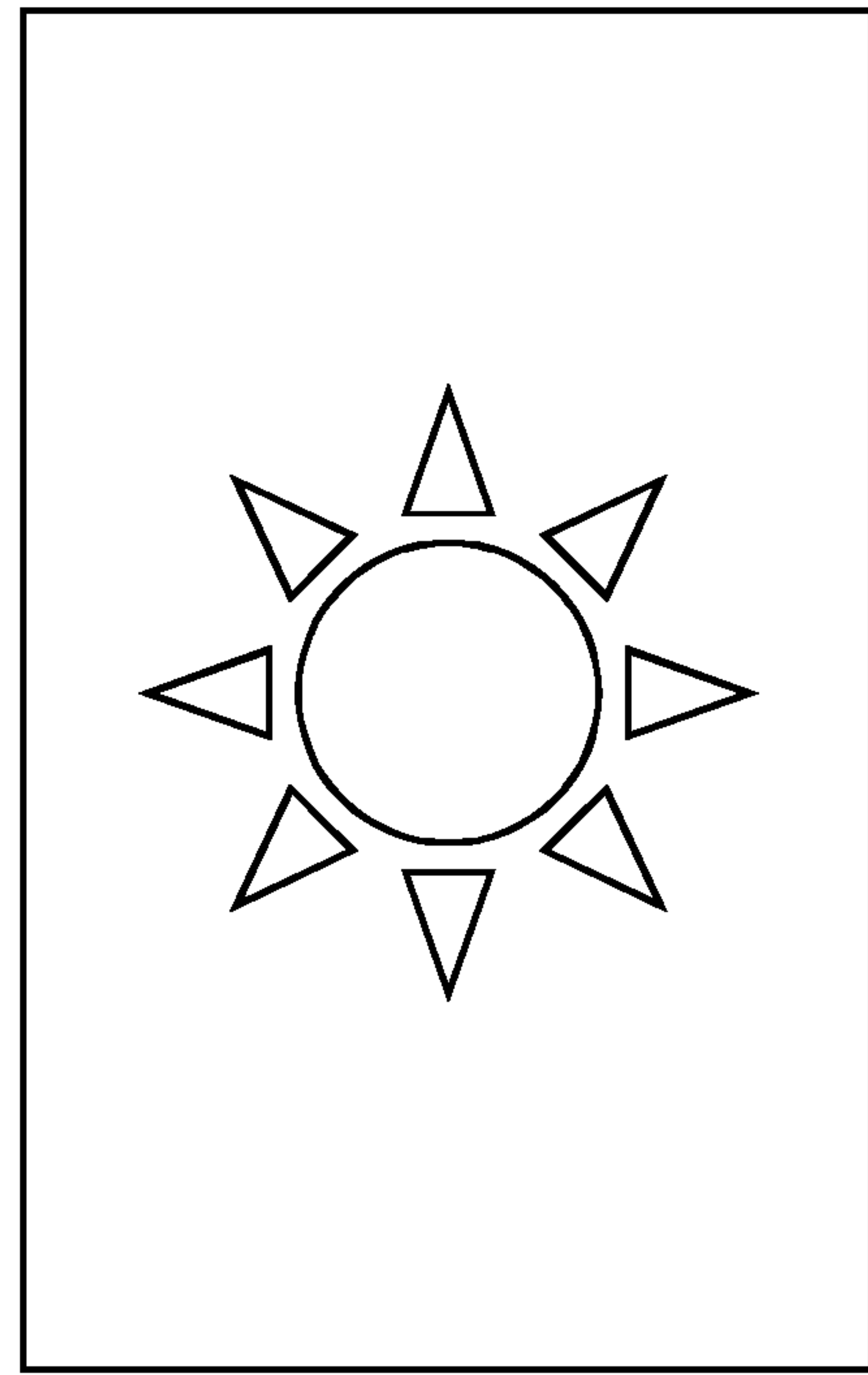
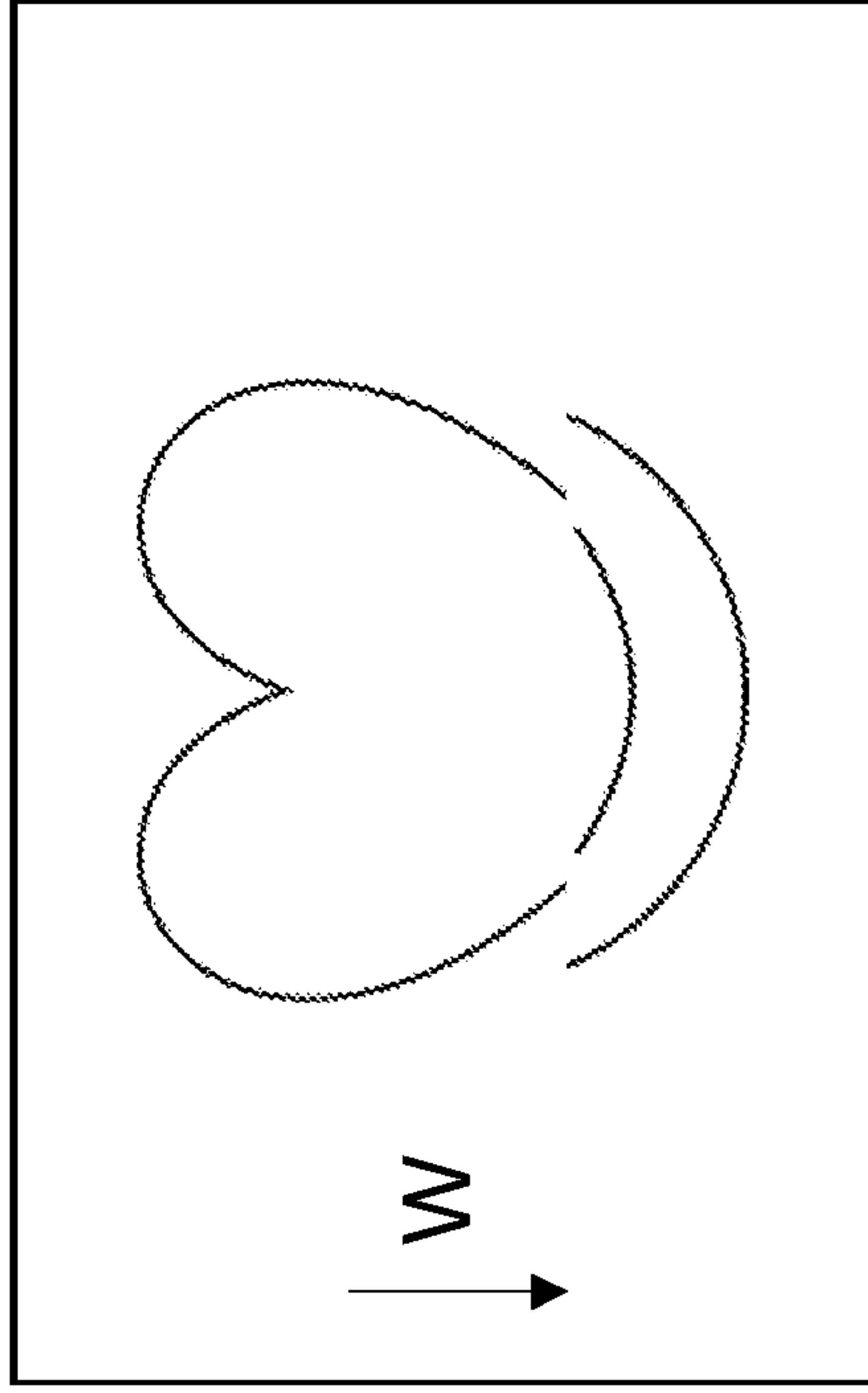
Figure 5c

FB1



S55

FB2



S56

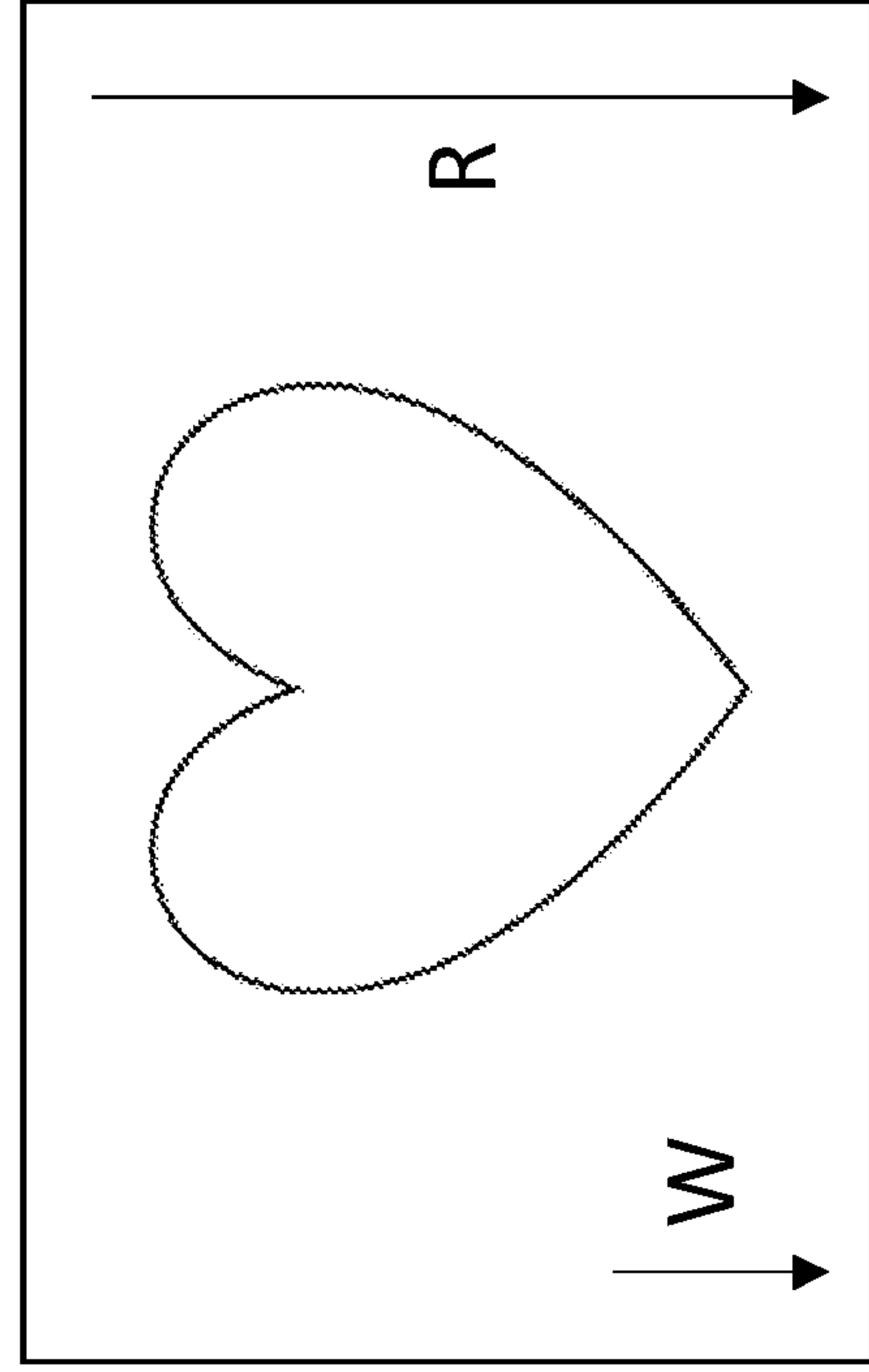


Figure 5d

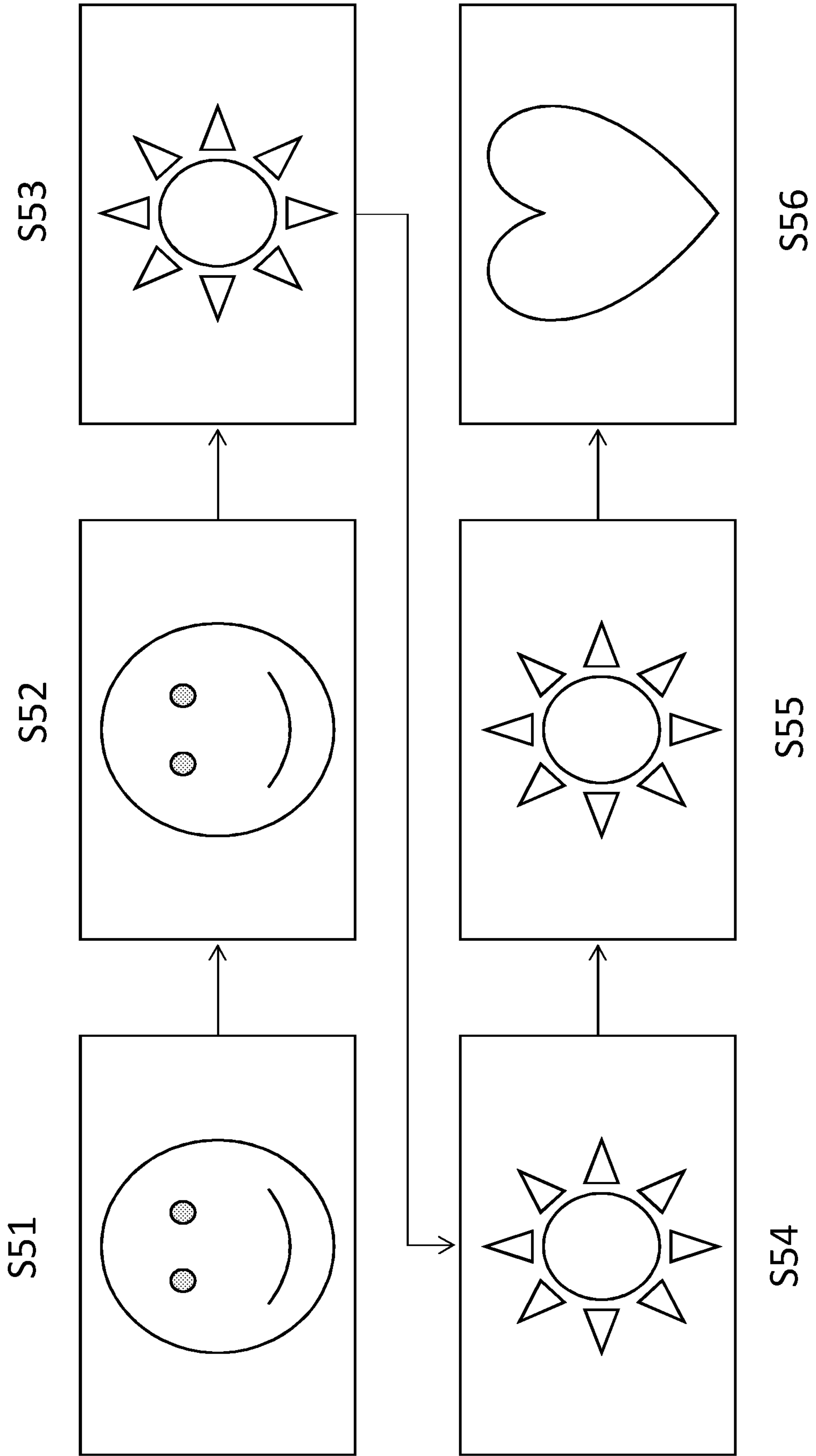
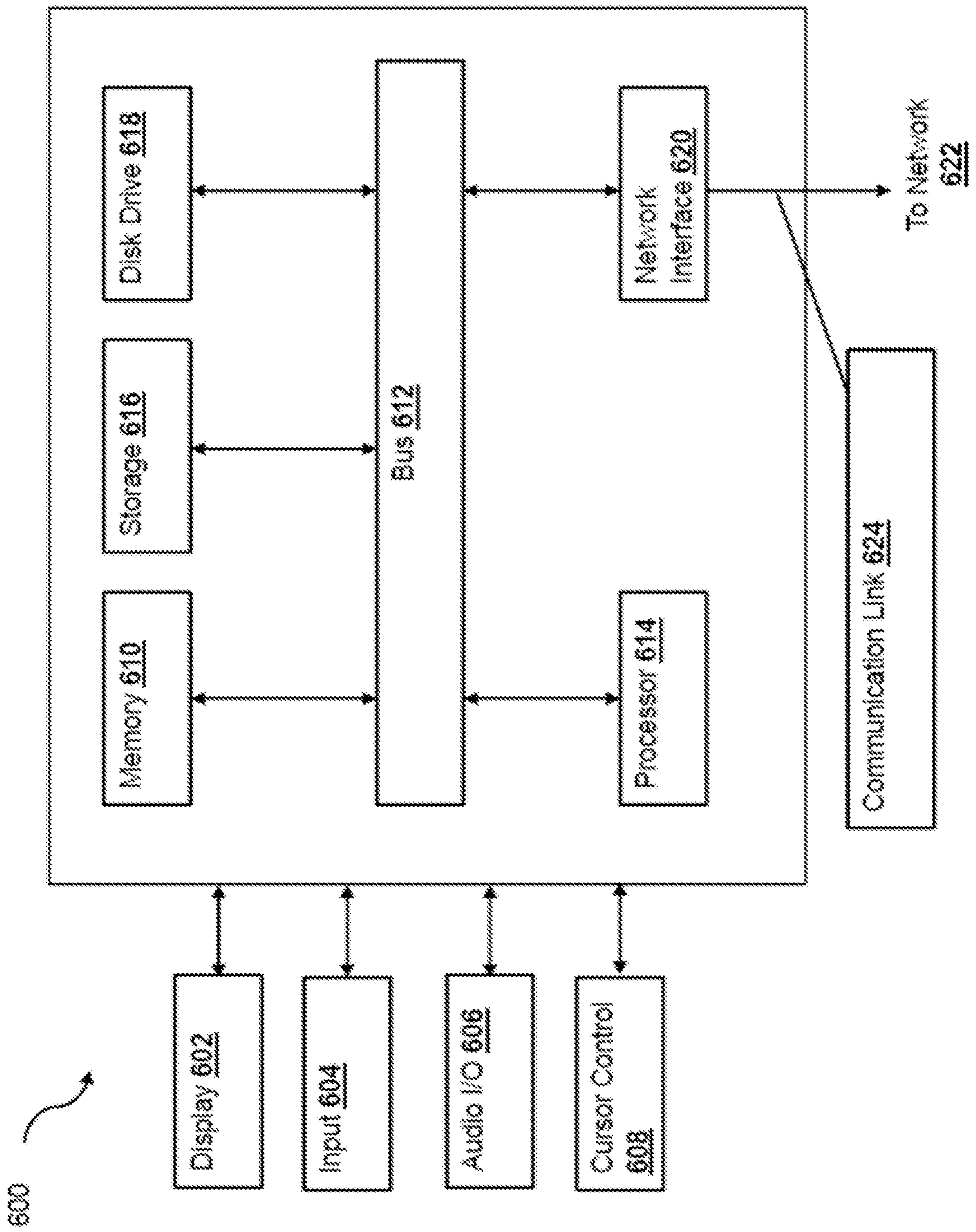


Figure 6





**INTERNATIONAL SEARCH REPORT**

International application No PCT/GB2021/050028
---

**A. CLASSIFICATION OF SUBJECT MATTER**  
 INV. G09G5/393      G09G5/395      G09G5/399  
 ADD.

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)  
 G09G

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)  
 EPO-Internal, WPI Data

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2004/135789 A1 (YU GYEONG-HO [KR]) 15 July 2004 (2004-07-15) paragraph [0003] - paragraph [0021] paragraph [0030] - paragraph [0051] figures 2-4	1-18
A	----- JP 2001 125548 A (MEGA CHIPS CORP) 11 May 2001 (2001-05-11) paragraph [0047] - paragraph [0061] figure 9	1,10,17, 18
A	----- WO 2006/113905 A2 (NOKIA CORP [FI]; NOKIA INC [US]) 26 October 2006 (2006-10-26) page 1, line 7 - page 2, line 31 -----	1,3,10, 17,18

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 application or patent 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  29 January 2021	Date of mailing of the international search report  09/02/2021
--	--

Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Authorized officer  Etienne, Christophe
--	---

# INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/GB2021/050028

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2004135789 A1	15-07-2004	KR 20040063207 A US 2004135789 A1	14-07-2004 15-07-2004
-----	-----	-----	-----
JP 2001125548 A	11-05-2001	JP 4446527 B2 JP 2001125548 A	07-04-2010 11-05-2001
-----	-----	-----	-----
WO 2006113905 A2	26-10-2006	CN 101176142 A EP 1878004 A2 US 2006238541 A1 WO 2006113905 A2	07-05-2008 16-01-2008 26-10-2006 26-10-2006
-----	-----	-----	-----