

***Arashi Vision (U.S.) LLC v.
GoPro, Inc.***

**IPR2024-00984
U.S. Patent No. 10,015,413**

Patent Owner's Demonstratives

Hearing: October 7, 2025

Summary: Grounds

Ground	Claim(s) challenged
1. Fleming	1, 6, 7
3. Obara + Liu	1, 6, 7
4. Meulen + Chao	1, 6, 7

- Remaining claims were disclaimed
- Ground 2 only challenges disclaimed claims

Grounds

- Institution Decision:

Ground	Reason
1. Fleming	No motivation to combine with camera
3. Obara + Liu	No motivation to combine with camera
4. Meulen + Chao	

Summary: Disputed Issues

- **Ground 4: Meulen-Chao disputed issues:**
 - **Transform “input image”** (1[f], 2[d]):
 - Issue: Whether Meulen discloses transforming an image from the embedded camera.
 - Meulen only discloses transforming “media source content,” which is from an external source, not the embedded camera.
 - **“determine a portion”** (1[e], 2[c]):
 - Issue: Whether Meulen discloses a step of “determin[ing] a portion” after obtaining an image
 - This term should be construed as a distinct step occurring after the “obtain an input image” step
 - Meulen only describes **pre-determined** portions
 - **“non-uniform” transformation** (1[g], 2[e]):
 - Issue: Whether “non-uniform” should be construed as “non-linear,” and whether Meulen’s piecewise linear transformation is “non-linear”
 - Specification and Petitioner’s expert agreed “non-uniform” requires “non-linear”
 - Piecewise **linear** transformation only approximates “non-linear”

Summary: Disputed Issues (Independent Claims)

- **Ground 4: Meulen-Chao disputed issues:**
 - ***“shifts the pixels”*** (1[g], 2[e]):
 - Issue: Whether Meulen or Chao disclose pixel shifting.
 - Meulen only discloses “polyphase filters,” which are not pixel shifting
 - Chao does not describe pixel shifting
 - No motivation to combine Meulen and Chao
 - ***Transformation along both axes*** (1[g], 2[e]):
 - Issue: Whether non-uniform transformation must occur along both axes, and whether Meulen discloses transformation along both axes.
 - Claim language is clear that transformation is based on the pixel’s position on both axes
 - Meulen only describes piecewise linear compression along a single axis

Summary: Disputed Issues

- **Objective evidence of non-obviousness:**
 - GoPro's SuperView embodies the '413 patent
 - GoPro's products were successful and had industry praise
 - Lema declaration and customer surveys show that SuperView contributed to GoPro's success
 - Evidence of copying by competitors
 - Long-felt need for immersive distortion
 - Unexpected result that distortion would lead to a more immersive feel for action cameras

Summary: Disputed Issues (Independent Claims)

- **Ground 1: Fleming disputed issues:**
 - **“image capture apparatus” or “camera”** (1[pre], cl. 6):
 - Issue: Whether Fleming renders obvious performing its transformation on a camera
 - Petition provided no evidence of cameras needing to perform aspect ratio conversion
 - Petition did not assert reasonable expectation of success
 - Reply’s new system art evidence is improper
 - Reply provides no motivation to use Fleming’s algorithm over system art’s letterboxing/cropping
 - **“determine a portion”** (1[e], 2[c]):
 - Issue: Whether Fleming performs a distinct step of “determine a portion”
 - Petition simply cited to Fleming’s transformation, where the portions are pre-determined
 - None of the new Reply citations indicate determination that occurs **after** obtaining an image

Summary: Disputed Issues (Independent Claims)

- **Ground 1: Obara-Liu disputed issues:**
 - ***“image capture apparatus” or “camera”*** (1[pre], cl. 6):
 - Issue: Whether Obara-Liu renders obvious performing its transformation on a camera
 - Petition provided no evidence of cameras needing to perform aspect ratio conversion
 - Petition did not assert reasonable expectation of success
 - Reply’s new system art evidence is improper
 - Reply provides no motivation to use Fleming’s algorithm over system art’s letterboxing/cropping
 - ***“determine a portion”*** (1[e], 2[c]):
 - Issue: Whether Obara performs a distinct step of “determine a portion”
 - Petition simply cited to Obara’s transformation, where the portions are pre-determined
 - Reply improperly relies on Fleming’s Figure 10, but that is piecewise linear

'413 Patent Overview

Patent Overview

- '413 Patent, Claim 1:

1. An image capture apparatus, comprising:
an image sensor configured to capture an input image having a source aspect ratio, the input image including pixels located at input positions defined along a first axis and a second axis, the first axis perpendicular to the second axis;
a display coupled to one or more physical processors; and
the one or more physical processors configured by machine-readable instructions to:
obtain the input image;
determine a portion within the input image, the portion including a subset of the pixels; and
apply a transformation to the input image to generate an output image, the output image having a target aspect ratio different than the source aspect ratio, the output image including the pixels located at output positions defined along the first axis and the second axis, wherein:
the transformation non-uniformly shifts the pixels from the input positions to the output positions based on (1) the input positions along the first axis, and (2) the input positions along the second axis; and
the transformation non-uniformly shifts the pixels such that differences between the input positions and the output positions of the subset of the pixels within the portion are less than differences

Summary of Prior Art

Summary of Prior Art: Meulen (Ex. 1007)

“The media source generally may comprise any source capable of delivering media source content to the wireless device 100.”

Ex. 1007 (Meulen) at 3:51-53; POR 13

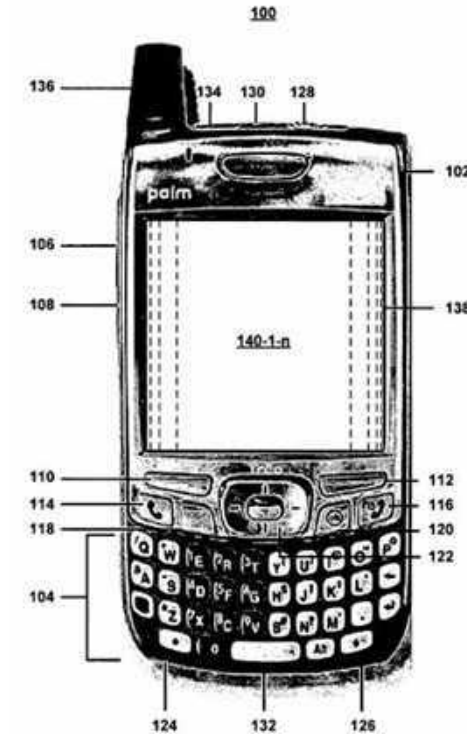


FIG. 1

Summary of Prior Art: Meulen (Ex. 1007)

“In various embodiments, the processor 510 may be arranged to receive information, such as media source content, through a communications interface 512. The communications interface 512 may comprises any suitable hardware, software, or combination of hardware and software that is capable of coupling the computing system 512 to one or more networks and/or network devices. The communications interface 512 may be arranged to operate with any suitable technique for controlling information signals using a desired set of communications protocols, services or operating procedures. The communications interface 512 may include the appropriate physical connectors to connect with a corresponding communications medium.”

Ex. 1007 (Meulen) at 13:5-17; Surreply 10

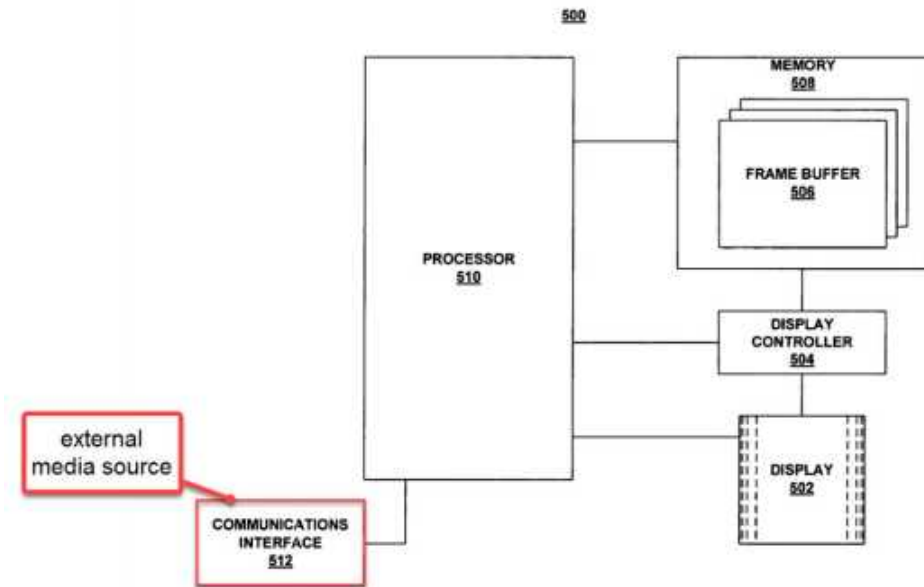


FIG. 5

Summary of Prior Art: Meulen (Ex. 1007)

TABLE 1

	Region					Total
	1	2	3	4	5	
Pixels/Region	10	50	200	50	10	320
Compression Ratio	4:1	2:1	none	2:1	4:1	
Source Pixels	40	100	200	100	40	480

Ex. 1007 (Meulen) at 5:54-60; POR 25

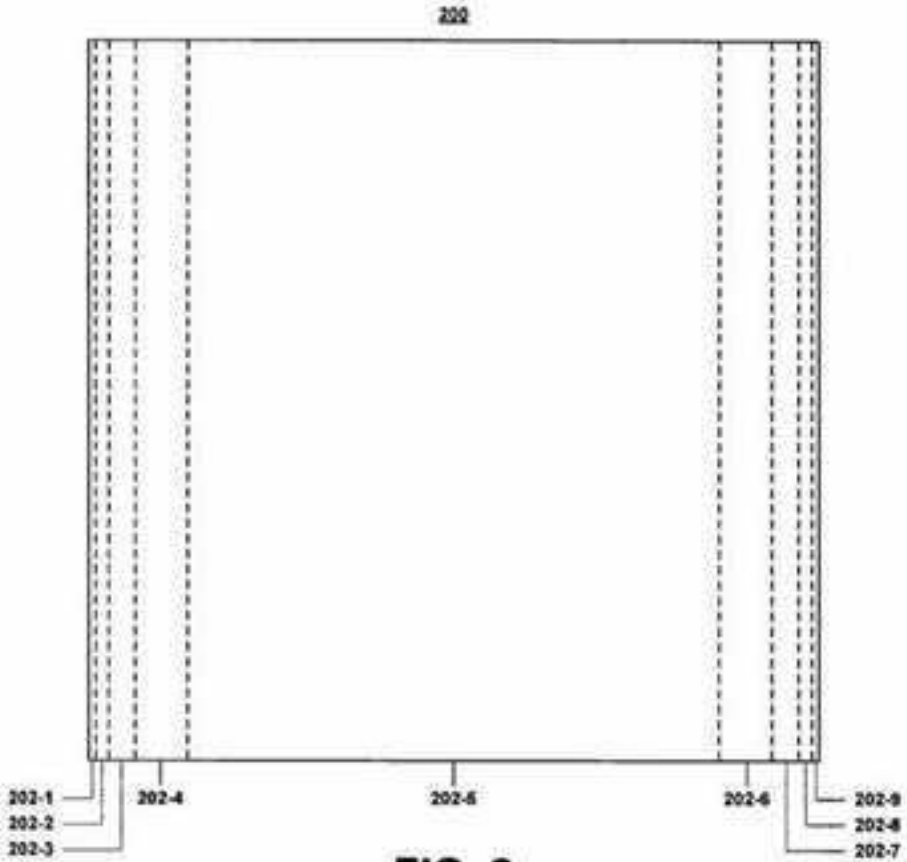


FIG. 2

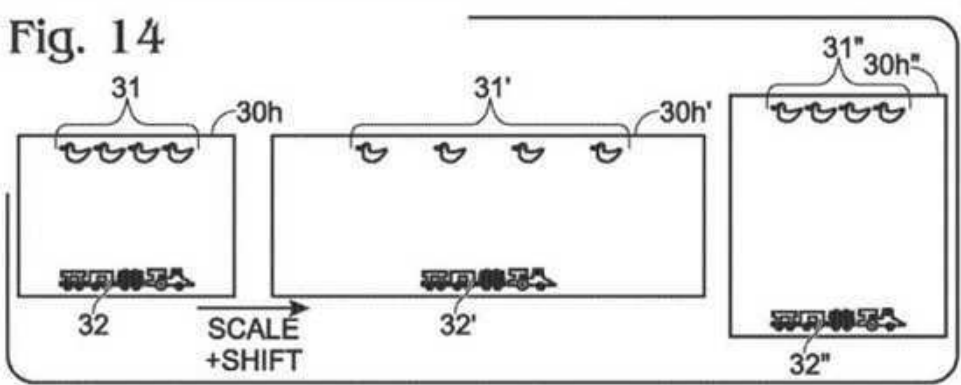
Summary of Prior Art: Chao (Ex. 1008)

“For example, people frequently like to incorporate professionally designed borders in their photos, newsletters, postcards, greeting cards and the like. As these images become more and more complex, they frequently incorporate elements that are designed for a specific page size. However, given the multitude of graphic design programs that allow people to resize their documents to any desired size, it would be extremely difficult, if not impossible for a graphic designer to create different versions of the same design for each possible page size.”

Ex. 1008 (Chao) at ¶1; POR 37

“The method comprises separating the foreground elements from the background elements, determining an original aspect ratio of the foreground elements and creating resized foreground elements by scaling the foreground elements while maintaining the original aspect ratio.”

Ex. 1008 (Chao) at Abstract; Pet. 83



Summary of Prior Art: Fleming (Ex. 1004)

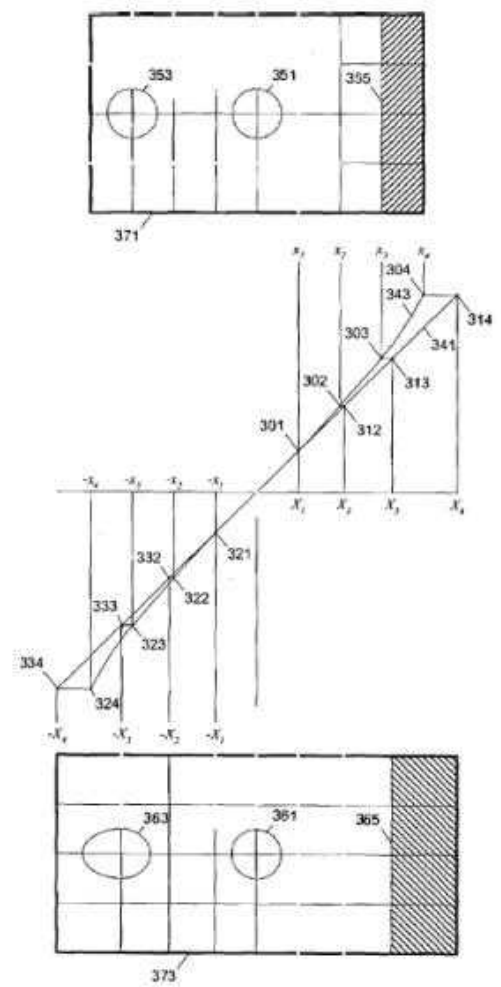


Fig. 3

Summary of Prior Art: Obara (Ex. 1006)

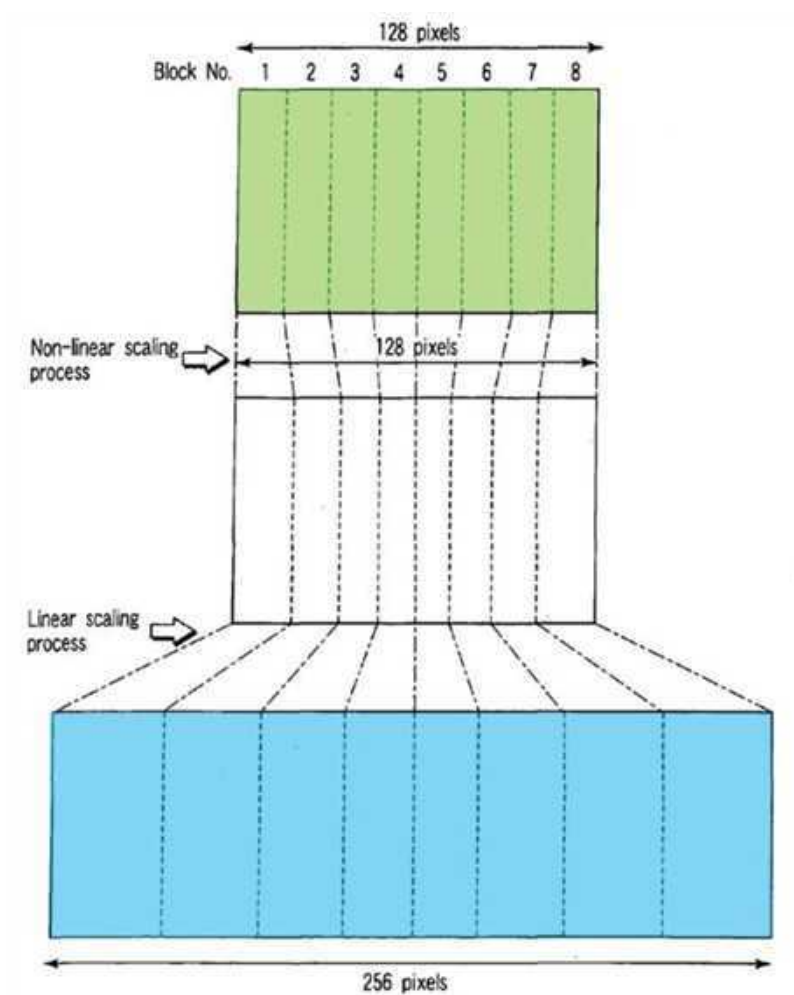


FIG. 6

Ground 4: Meulen Does Not Disclose Transforming An “Input Image” (All Claims)

Ground 4: Meulen Does Not Disclose an “Input Image” From the Camera

- ***Petitioner:*** Meulen’s “media source content” is an input image from the camera
- **Incorrect:**
 - Meulen’s “media source content” is received from external source via a “communications interface”
 - No motivation to modify Meulen – no disclosure of the aspect ratio of its camera sensor
 - Petitioner raises ***new evidence*** and ***argument*** on reply:
 - Palm Treo documentation
 - Unauthenticated documents (subject to motion to strike)
 - Petitioner’s new evidence supports Patent Owner’s position that Muelen has ***no need*** to apply its piecewise linear transformation on an image taken with the camera.

Ground 4: Meulen Does Not Disclose an “Input Image” From the Camera

- Claims:

1. An image capture apparatus, comprising:
an **image sensor configured to capture an input image** having a source aspect ratio, the input image including pixels located at input positions defined along a first axis and a second axis, the first axis perpendicular to the second axis;
a display coupled to one or more physical processors; and
the one or more physical processors configured by machine-readable instructions to:
obtain the input image;
determine a portion within the input image, the portion including a subset of the pixels; and
apply a transformation to the input image to generate an output image, the output image having a target aspect ratio different than the source aspect ratio, the output image including the pixels located at output positions defined along the first axis and the second axis,

2. A system that converts between aspect ratios, the system comprising:

one or more physical processors configured by machine-readable instructions to:

[REDACTED] the input image having a source aspect ratio, the input image including pixels located at input positions defined along a first axis and a second axis, the first axis perpendicular to the second axis;

determine a portion within the input image, the portion including a subset of the pixels; and

apply a transformation to the input image to generate an output image, the output image having a target aspect ratio different than the source aspect ratio, the output image including the pixels located at output positions defined along the first axis and the second axis, wherein:

6. The system of claim 2, wherein the one or more

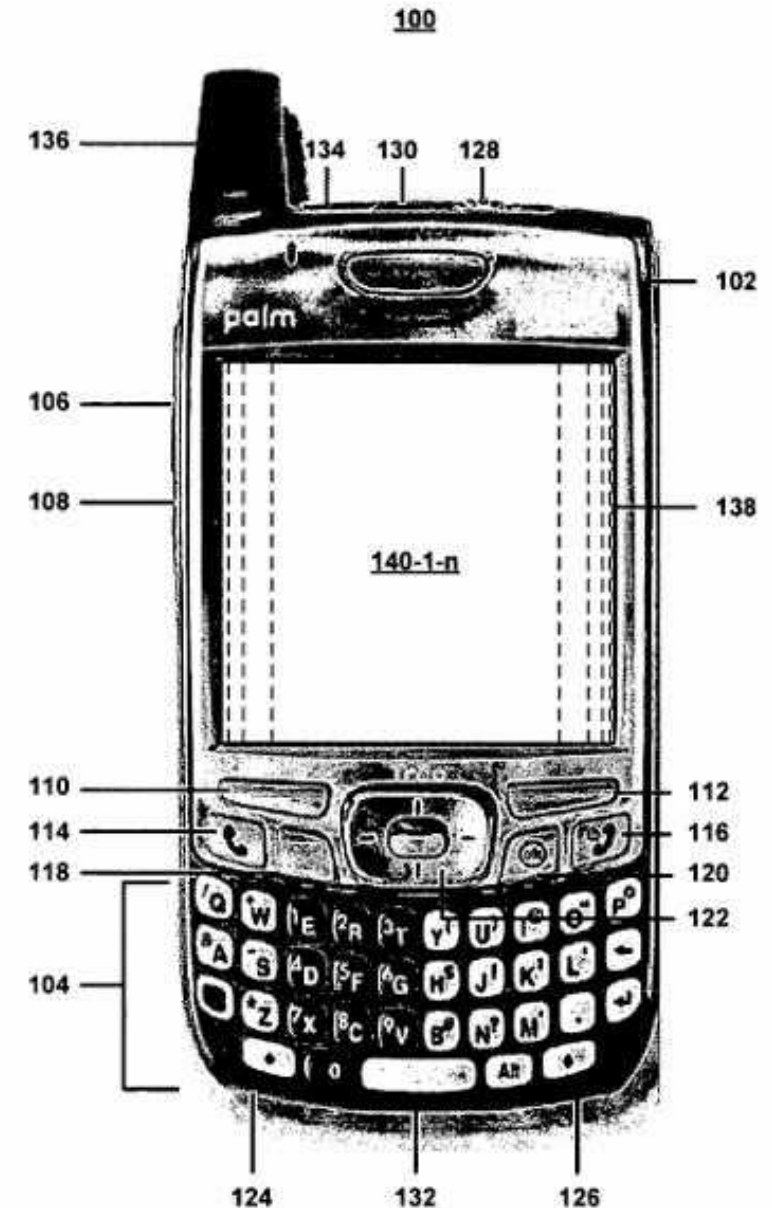
[REDACTED]

Ground 4: Meulen Does Not Disclose an “Input Image” From the Camera

- Meulen processes “media source content”

“In various embodiments, the image compression and/or scaling may be applied to the **media source content** using various physical or logical elements implemented as hardware, software, or any combination thereof, as desired for a given set of design parameters or performance constraints.”

Ex. 1007 (Meulen) at 4:50-53; Pet. 86



Ground 4: Meulen Does Not Disclose an “Input Image” From the Camera

- Meulen’s *media source content*:

“The display 138 may be arranged to display media source content received by the wireless device 100.”

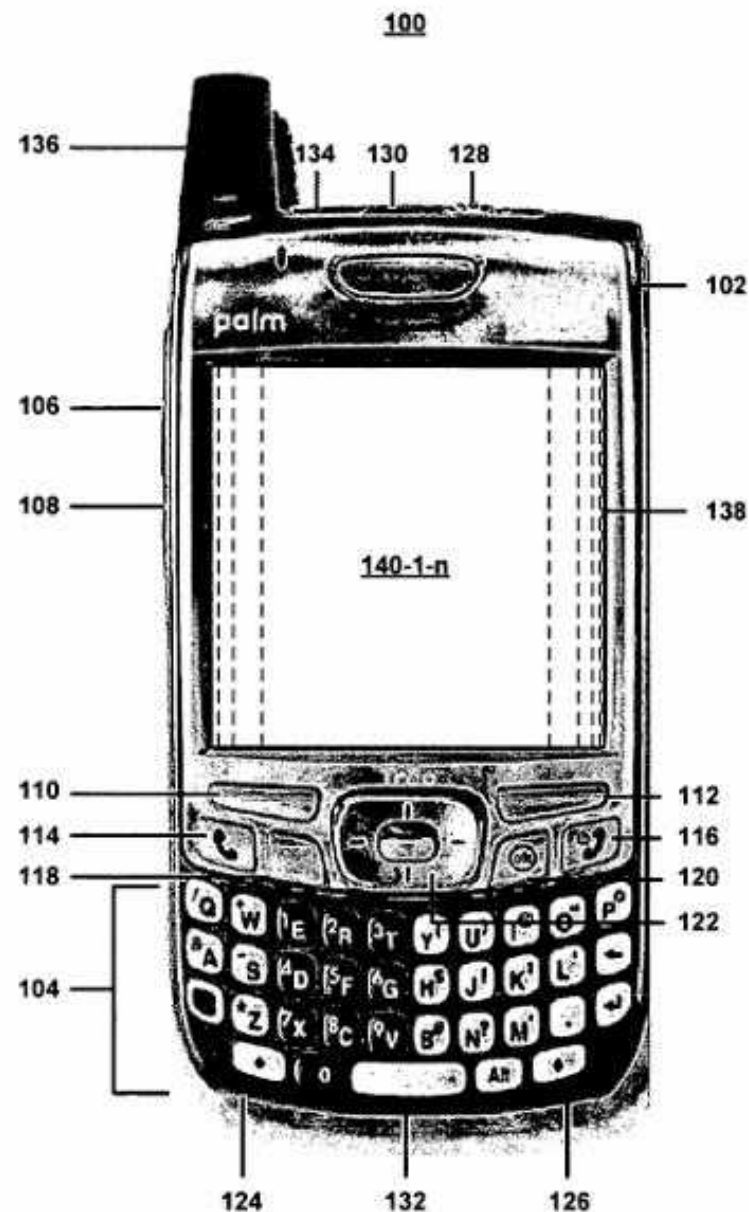
Ex. 1007 (Meulen) at 3:26-27; Pet. 87

“The media source generally may comprise any source capable of delivering media source content to the wireless device 100.”

Ex. 1007 (Meulen) at 3:51-53; POR 13

“In various embodiments, the wireless device 100 may be arranged to receive media source content through one or more wireless wide area network (WWAN) data communication services.”

Ex. 1007 (Meulen) at 4:1-12; POR 13-14



Ground 4: Meulen Does Not Disclose an “Input Image”

- Meulen’s ***media source content***:

“The media source generally may comprise any source capable of delivering media source content to the wireless device 100. An example of a media source may include a source for static or video image information, such as from a computer to a display. The media source may include a server, such as a web server arranged to deliver web content. Another example of a media source may include a source for A/V information such as television signals. The media source may be arranged to deliver standard analog television content, digital television content, HDTV content, and so forth. The media source may provide broadcast or streaming analog or digital video content. The media source may include a device arranged to deliver pre-recorded media content stored in various formats, such as a DVD device, a Video Home System (VHS) device, a digital VHS device, a digital camera, video camera, a portable media player, a gaming device, and so forth.

Ex. 1007 (Meulen) at 3:51-65; POR 13

Ground 4: Meulen Does Not Disclose an “Input Image” From the Camera

- *ID*:

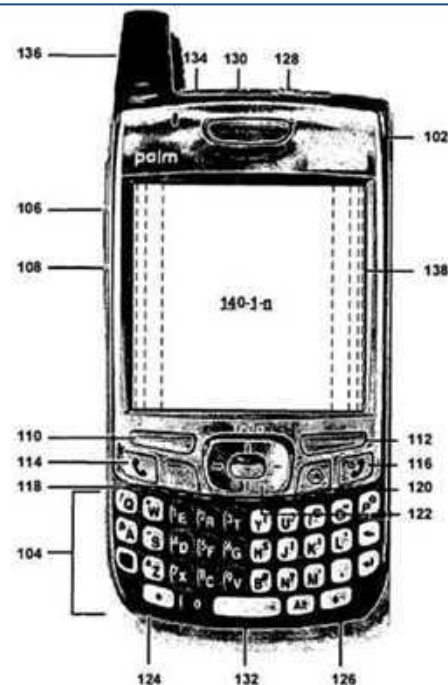


FIG. 1

Figure 1 of Meulen illustrates one embodiment of its wireless device. Ex. 1001, 2:8. The display 138 may display media source content received by the wireless device. *Id.* at 3:26–27. The media source content may have an aspect ratio that differs from the aspect ratio of the display. *Id.* at 4:13–15.

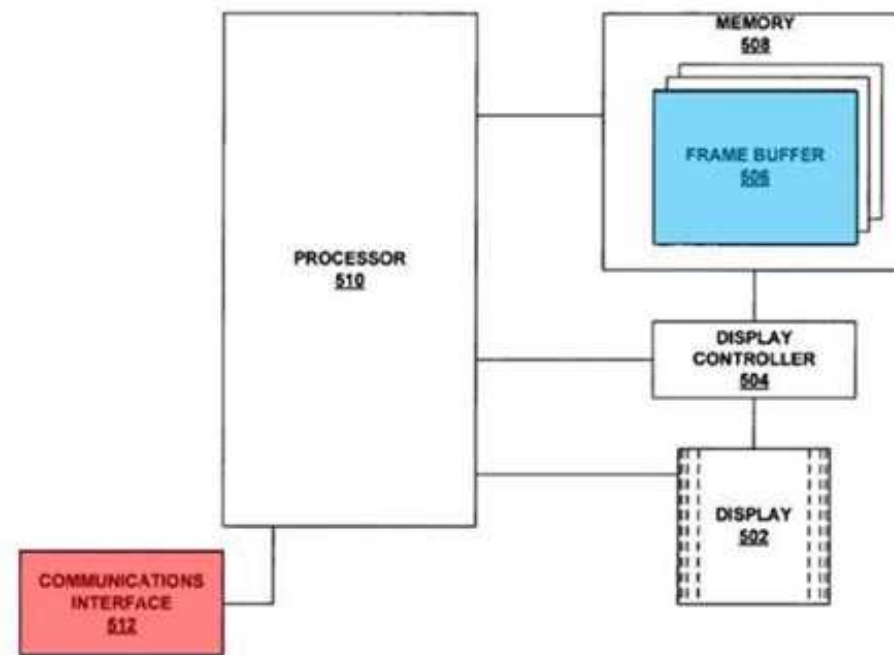
ID at 10

Ground 4: Meulen Does Not Disclose an “Input Image” From the Camera

- **New reply argument:** Media source content stored in a “frame buffer” must be internal images *Reply 4-5*

“Because the frame buffer 506 is not directly connected to the communications interface 512 (which Dr. Foroosh alleges is the interface to the external media source (EX2027, ¶57)), a POSITA would have understood that frame buffer 506 stores content from other sources such as internal media sources; otherwise, frame buffer 506 would be directly connected to the communications interface 512.”

Ex. 1014 (Katsaggelos Supp. Decl.) at ¶25; Reply 4



Ground 4: Meulen Does Not Disclose an “Input Image” From the Camera

- Meulen’s **communications interface** and **processor**:

“In various embodiments, the **processor 510** may be arranged to **receive information**, such as media source content, through a **communications interface 512**. The communications interface 512 may comprises any suitable hardware, software, or combination of hardware and software that is capable of coupling the computing system 512 to **one or more networks and/or network devices**. The communications interface 512 may be arranged to operate with any suitable technique for controlling information signals using a desired set of communications protocols, services or operating procedures. The communications interface 512 may include the appropriate physical connectors to connect with a corresponding communications medium.”

Ex. 1007 (Meulen) at 13:5-17; Surreply 10

“And the communications interface, that’s **always implying an external source**.”

Ex. 2026 (Katsaggelos Tr.) at 18:8-16; POR 15

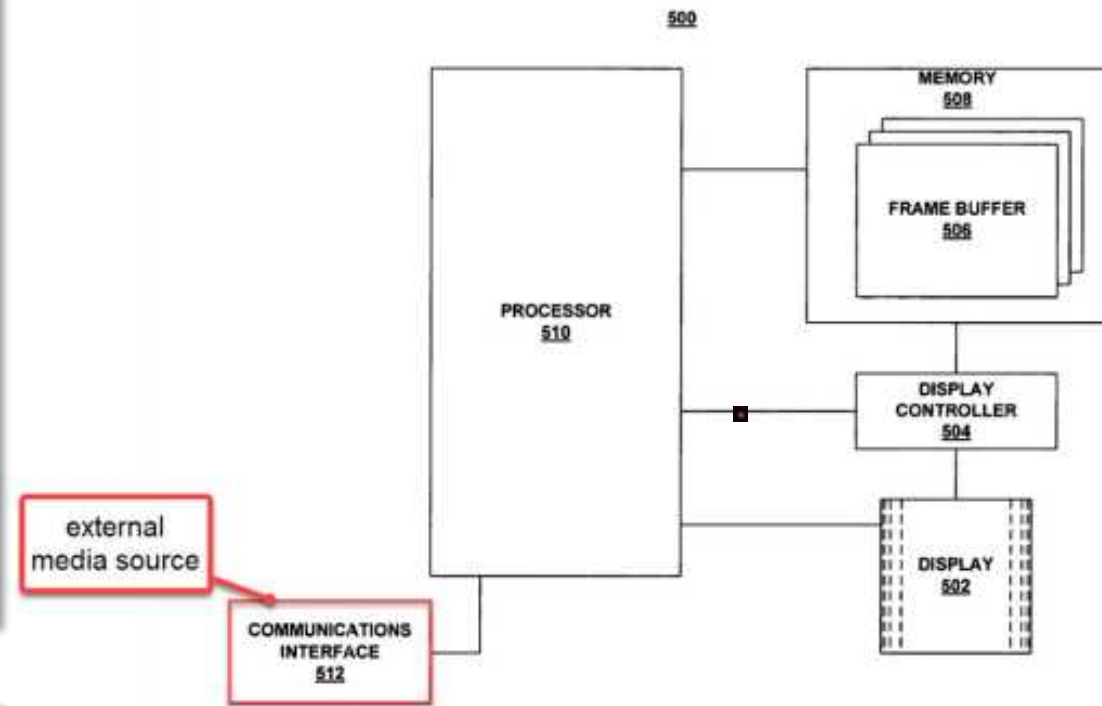
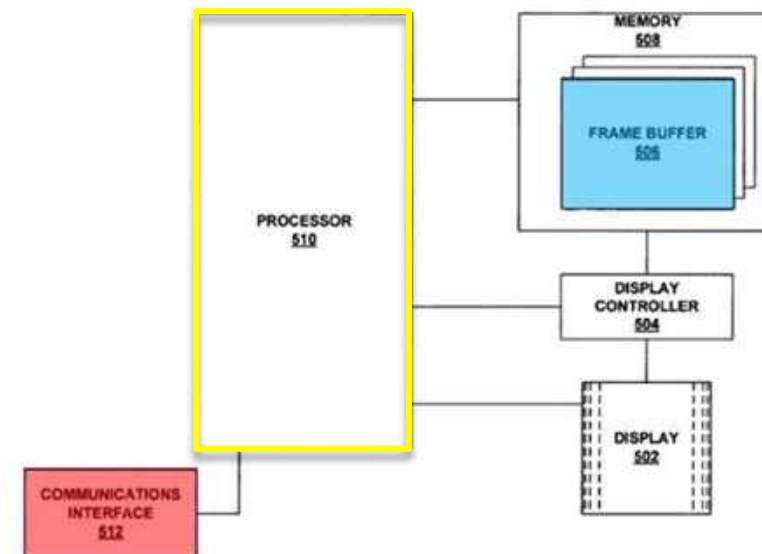


FIG. 5

Ground 4: Meulen Does Not Disclose an “Input Image” From the Camera

- Meulen’s ***communications interface*** and ***processor***:

“As shown, the frame buffer 506 may be implemented by a memory 508. In various embodiments, the memory 508 may comprise any machine-readable or computer-readable media capable of storing data, including both volatile and non-volatile memory. For example, memory may include read-only memory (ROM), random-access memory (RAM), dynamic RAM (DRAM), Double-Data-Rate DRAM (DDR-RAM), synchronous DRAM (SDRAM), static RAM (SRAM), programmable ROM (PROM), erasable programmable ROM (EPROM), electrically erasable programmable ROM (EEPROM), flash memory (e.g., NOR or NAND flash memory), content addressable memory (CAM), polymer memory (e.g., ferroelectric polymer memory), phase-change memory (e.g., ovonic memory), ferroelectric memory, silicon-oxide-nitride-oxide-silicon (SONOS) memory, disk memory (e.g., floppy disk, hard drive, optical disk, magnetic disk), or card (e.g., magnetic card, optical card), or any other type of media suitable for storing information.”



Ex. 1007 (Meulen) at 12:26-43; Surreply 10, Ex. 1014 (Katsaggelos Supp. Decl.) at ¶55

Ground 4: Meulen Does Not Disclose an “Input Image” From the Camera

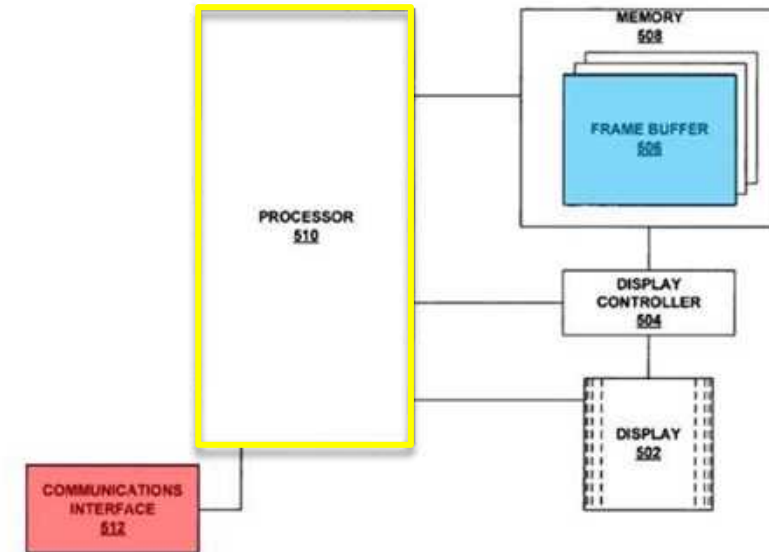
- Meulen’s *communications interface* and *processor*:

“The computing system 500 may comprise a processor 510 such as a central processing unit (CPU). In various embodiments, the processor 510 may be implemented as a **general purpose processor** ...”

Ex. 1007 (Meulen) at 12:44-47; Surreply 10

“In various embodiments, the processor 510 may be arranged to run an operating system (OS) and various mobile applications. Examples of an OS include, for example, a Microsoft® Windows OS, a Palm OS®, and any other proprietary or open source OS.”

Ex. 1007 (Meulen) at 12:62-66; Surreply 10

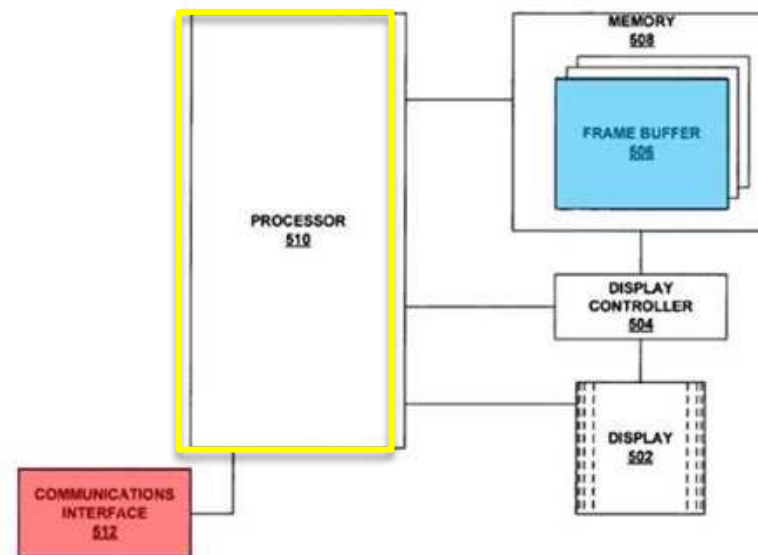


Ground 4: Meulen Does Not Disclose an “Input Image” From the Camera

- Meulen’s ***communications interface*** and ***processor***:

“In various embodiments, the **processor 510** may be arranged to receive information, such as media source content, through a **communications interface 512**. The communications interface 512 may comprises any suitable hardware, software, or combination of hardware and software that is capable of coupling the computing system 512 to one or more networks and/or network devices. The communications interface 512 may be arranged to operate with any suitable technique for controlling information signals using a desired set of communications protocols, services or operating procedures. The communications interface 512 may include the appropriate physical connectors to connect with a corresponding communications medium.”

Ex. 1007 (Meulen) at 13:5-17; Surreply 10



Ground 4: Meulen Does Not Disclose an “Input Image” From the Camera

- **New reply argument:** Palm Treo, *as evidenced by new, unauthenticated documents submitted on Reply*, establish motivation to modify Meulen Reply 6-7

“Also, Meulen’s device is a Palm® Treo™. (EX1007, 2:12.) Palm Treos *[as allegedly established by newly submitted documents]* at that time displayed images captured by its built-in camera.”

Reply 6

“[PO] argues that there is no need for Meulen to perform aspect ratio conversion. (POR, 16.) Again, that is not what Meulen discloses. ... Palm Treos’ aspect ratios also do not match *[as allegedly established by newly submitted documents]*.”

Reply 7

- **No dispute:** Meulen does not disclose the resolution of its camera sensor
- Petitioner **does not** contend that Meulen incorporates by reference any Palm Treo documentation
- Petitioner **does not** contend that Palm Treo was within the knowledge of a POSITA
- Petitioner **does not** argue its newly submitted, unauthenticated evidence was properly submitted with the petition

Ground 4: Meulen Does Not Disclose an “Input Image” From the Camera

- Meulen **does not** disclose the resolution of the embedded camera

“The wireless device 100 may comprise various input/output (I/O) devices such as a keyboard, keys, buttons, switches, a microphone, an audio headset, a camera, a touch-sensitive display screen, a stylus, and so forth.”

Ex. 1007 (Meulen) at 2:29-32; Pet. 97, Ex. 1002 (Katsasgelos Decl.) at ¶290

“A wireless device such as a handheld computer and/or mobile telephone may incorporate features such as a color display screen, a digital camera, Web access, and multimedia capability.”

Ex. 1007 (Meulen) at 1:6-9; POR 15

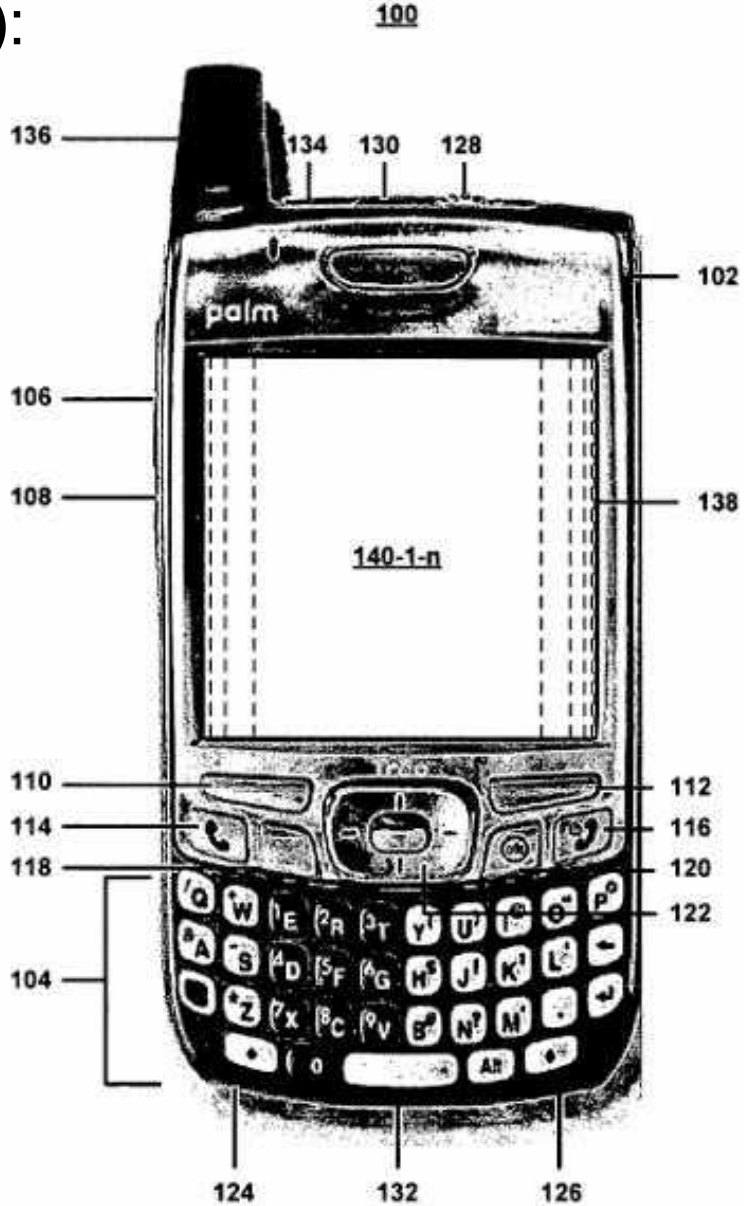
- Meulen discloses the resolution of external devices

“Most media sources, however, have a non-square aspect ratio such as 4:3, 3:2, 16:9, or even 2:1. For example, the aspect ratio for standard television as well as for many web sites and multimedia products is 4:3. High-definition television (HDTV) and many digital video cameras use an aspect ratio of 16:9. Digital video disc (DVD) players may be designed to use an aspect ratio of either 4:3 or 16:9.”

Ex. 1007 (Meulen) at 1:18-25; POR 15

Ground 4: Meulen Does Not Disclose an “Input Image” From the Camera

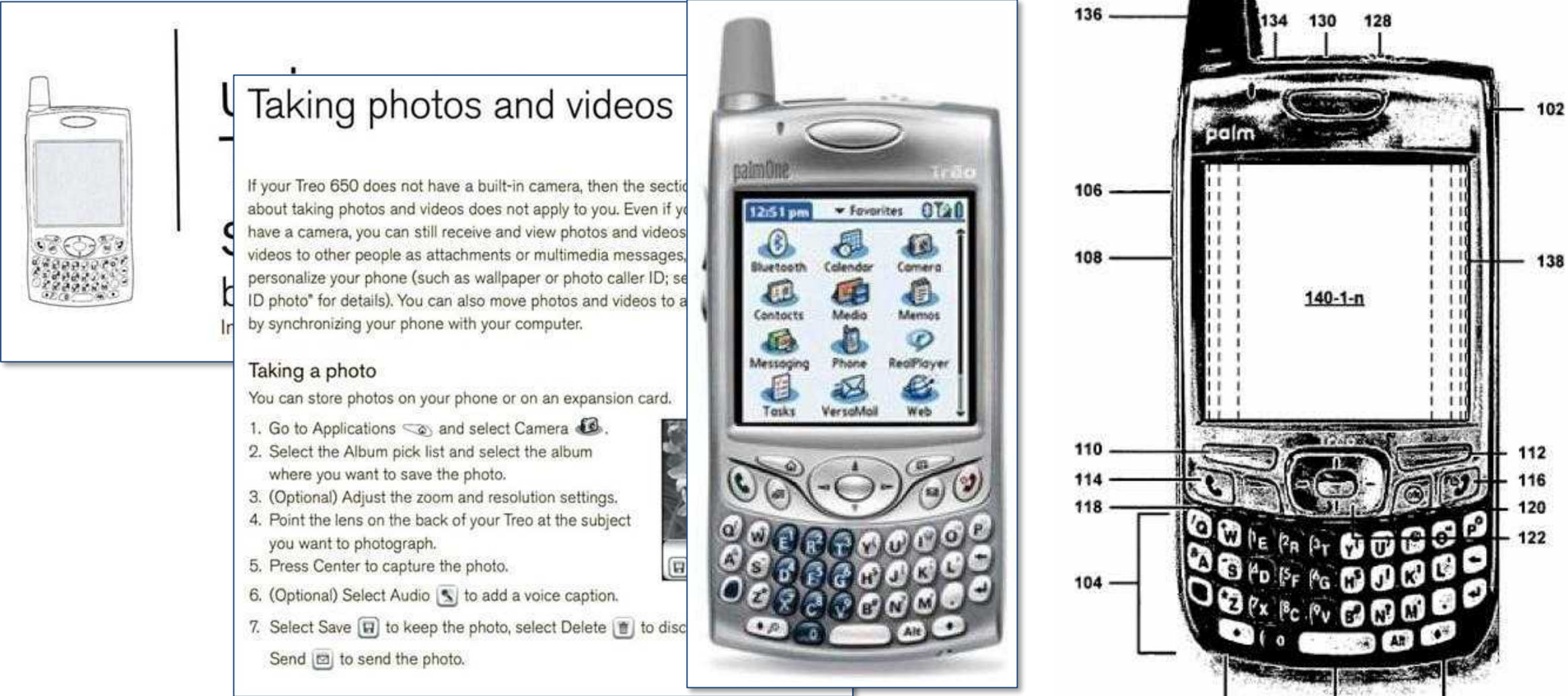
- Petitioner’s new evidence (subject to motion to strike):



Ex. 1018 (Palm Treo 600 Uesr Manual); Ex. 1022; Reply 7, Ex. 1014 (Katsaggelos Supp. Decl.) at ¶37

Ground 4: Meulen Does Not Disclose an “Input Image” From the Camera

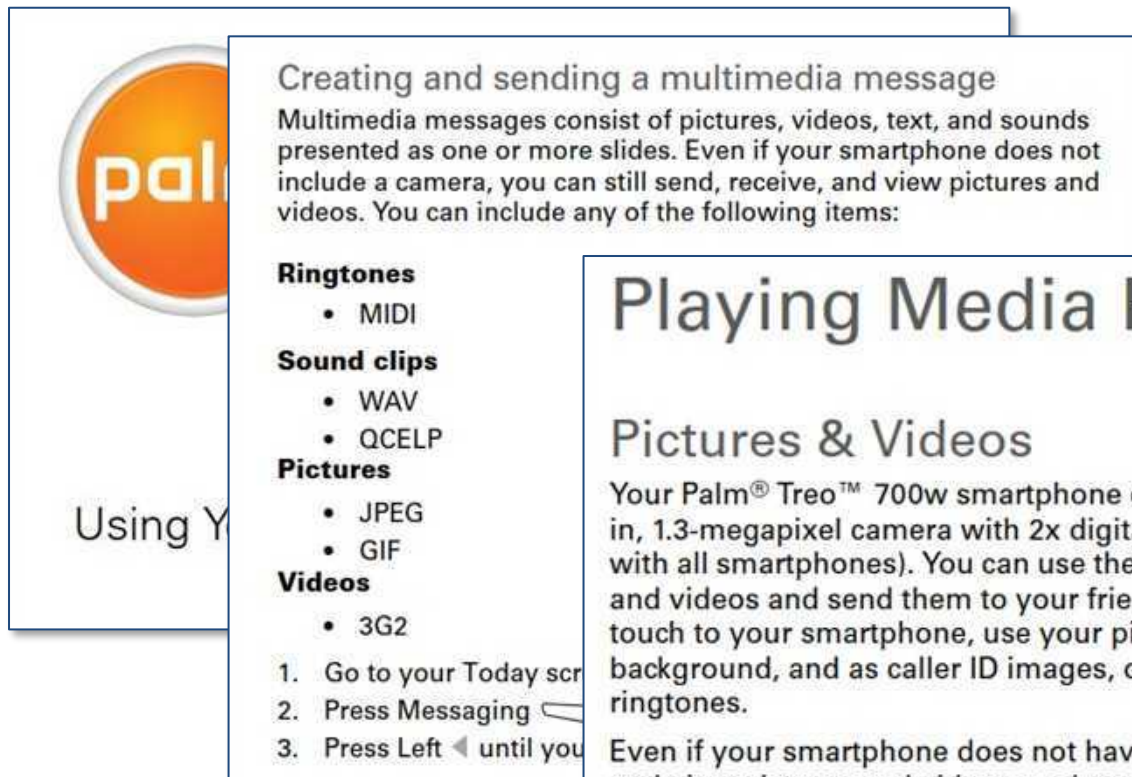
- Petitioner’s new evidence (subject to motion to strike):



Ex. 1020 (Palm Treo 650 User Manual); Ex. 1023; Reply 7, Ex. 1014 (Katsaggelos Supp. Decl.) at ¶¶32, 37

Ground 4: Meulen Does Not Disclose an “Input Image” From the Camera

- Petitioner’s new evidence (subject to motion to strike):

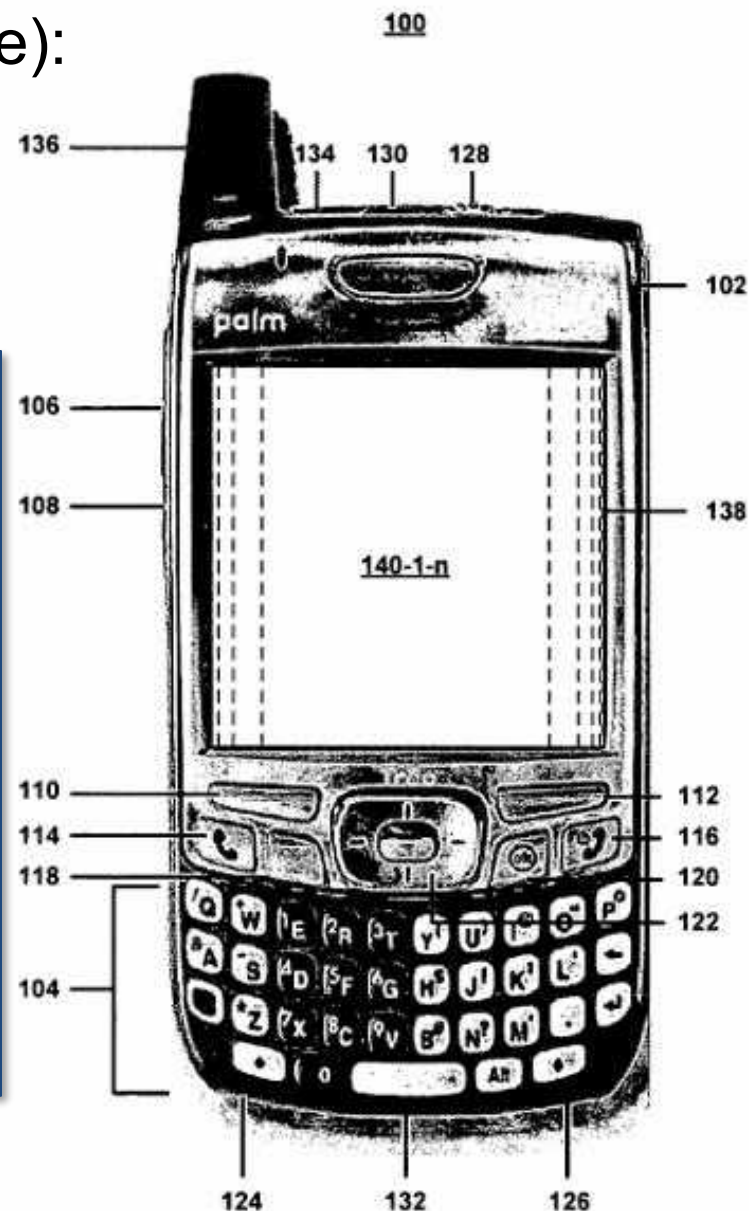


Playing Media Files

Pictures & Videos

Your Palm® Treo™ 700w smartphone comes with an easy-to-use, built-in, 1.3-megapixel camera with 2x digital zoom (camera not included with all smartphones). You can use the camera to take and view pictures and videos and send them to your friends and family. To add a personal touch to your smartphone, use your pictures as your Today screen background, and as caller ID images, or use your videos to create video ringtones.

Even if your smartphone does not have a camera, you can still receive and view pictures and videos, and send pictures and videos as attachments or multimedia messages. You can also personalize your smartphone by using a picture as a background or by adding a picture to a contact; see “Assigning a picture and ringtone ID to a contact” on page 176 for details). You can also move pictures and videos to a computer by synchronizing your smartphone with your computer.



Ex. 1021 (Palm Treo 700w User Manual); Reply 7

Ground 4: Meulen Does Not Disclose an “Input Image” From the Camera

- Palm Treo documentation:

CHAPTER 6 Playing Media Files

*** Tip**
If you see a camcorder icon below the preview image, video mode is on. To turn on the still camera, press Menu and select Still Mode.

*** Tip**
You can also adjust the zoom and brightness by pressing Menu and selecting Zoom or Brightness.

+ Did You Know?
Videos are captured and stored in 3G2 format. Video resolution settings range from 352 x 288 to 176 x 144 pixels.

+ Did You Know?
After three minutes of inactivity when previewing an image, the camera goes into standby mode. Press any key or tap the screen to return to the image preview.

1. Press Start  and select Pictures & Videos.
2. Select Camera .
3. Adjust the position of your smartphone until you see the subject you want to photograph on the screen.
4. (Optional) Adjust any of the following:
 - **Zoom:** Press Up , to zoom in or Down  to zoom out.
 - **Brightness:** Press Right  to increase the brightness or Left  to decrease the brightness.
 - **Resolution:** Press Menu  (right action key) and select Resolution to change the image quality.
5. (Optional) Set a 5-second timer: Press Menu  (right action key), select Mode, and then select Timer.
6. Press Center  to capture the picture or start the timer.
7. Hold your smartphone still until the picture renders, and then do one of the following:
 - Press Camera  (left action key) to take another picture.
 - Press OK  to return to the Thumbnails View.



Taking pictures in burst mode
Burst mode takes five pictures in quick succession, with one button press.

1. Press Start  and select Pictures & Videos.
2. Press Camera  (left action key).

118

Using Your Treo 700w Smartphone

- **Resolution:** Press Menu  and select Quality to change the video resolution.

6. Press Center  to start recording.
7. When you're finished recording, press Stop  (left action key) or Center  to stop recording.
8. (Optional) To review the video in Windows Media Player Mobile, press Thumbnails  (left action key), highlight the video, and then press Center . Press OK  to return to the Thumbnail View.



Viewing a picture or video

1. Press Start  and select Pictures & Videos.
2. Select the picture or video you want to view. For more info on viewing videos, see "Playing media files on your smartphone" on page 128.
3. Press OK  to return to the Thumbnail View.

Ground 4: Meulen Does Not Disclose “Determine a Portion” (All Claims)

Ground 4: Claim Construction of “Determine a Portion”

- **Claim Language:** “determine a portion within the input image”
 - The transformation performs less pixel shifting in this “portion”

Proposed Construction	
Patent Owner	Petitioner
“Determin[ing]” step must be distinct and performed after “obtain[ing]” input image but before “transformation” step	“Determin[ing]” step can be performed before “obtain[ing]” input image

- Petitioner is **incorrect:**
 - Claim requires “determin[ing] a portion within **the input image**” that was obtained
 - “Determin[ing]” step was added to claims to overcome art that merely applied a transformation

Ground 4: Claim Construction of “Determine a Portion”

- Claim 1 requires a specific order:

1. An image capture apparatus, comprising:
an image sensor configured to capture an input image having a source aspect ratio, the input image including pixels located at input positions defined along a first axis and a second axis, the first axis perpendicular to the second axis;
a display coupled to one or more physical processors; and
the one or more physical processors configured by machine-readable instructions to:
obtain the input image;
determine a portion within the input image, the portion including a subset of the pixels; and
apply a transformation to the input image to generate an output image, the output image having a target aspect ratio different than the source aspect ratio, the output image including the pixels located at output positions defined along the first axis and the second axis, wherein:
the transformation non-uniformly shifts the pixels from the input positions to the output positions based on (1) the input positions along the first axis, and (2) the input positions along the second axis; and
the transformation non-uniformly shifts the pixels such that differences between the input positions and the output positions of the subset of the pixels within the portion are less than differences

Ground 4: Claim Construction of “Determine a Portion”

- Claim 2 requires a specific order:

obtain an input image, the input image having a source aspect ratio, the input image including pixels located at input positions defined along a first axis and a second axis, the first axis perpendicular to the second axis;

determine a portion within the input image, the portion including a subset of the pixels; and

apply a transformation to the input image to generate an output image, the output image having a target aspect ratio different than the source aspect ratio, the output image including the pixels located at output positions defined along the first axis and the second axis, wherein:

...

the transformation non-uniformly shifts the pixels such that differences between the input positions and the output positions of the subset of the pixels within the portion are less than differences between the input positions and the output positions of others of the pixels.

Ground 4: Claim Construction of “Determine a Portion”

- The specification:

“FIG. 2A illustrates an example embodiment of a process for converting from a source aspect ratio to a target aspect ratio in a camera system. In the process of FIG. 2A, **an input image is obtained** 202 in the source aspect ratio (e.g., a 4:3 aspect ratio). ...

In the illustrated example, the center of the image is chosen to be the region of interest that is most fully corrected and the distortion increases in each direction as the frame boundary is approached. ...

In other embodiments, **a different part of the image may be selected** as the area most fully corrected, or in the case, of video, **the area may dynamically change between images** (e.g., to follow a moving object).”

Ex. 1001 (413 patent) at 3:61-4:36; POR 19

“Furthermore, in other alternative embodiments, the equations may change dynamically in different frames of a video. For example, **warping may be applied in a manner that reduces distortion applied to a particular object in motion across different frames** of a video since the viewer is likely to focus attention on the object in motion.

Ex. 1001 (413 patent) at 7:64-8:2; Surreply 8

Ground 4: Claim Construction of “Determine a Portion”

- ***Petitioner:*** '413 specification discloses “vertical warp distortion” with pre-determined portions *Reply 3*
- Petitioner is **incorrect:**
 - Specification describes changing the portion to track a moving object to have less distortion

“Furthermore, in other alternative embodiments, the equations may change dynamically in different frames of a video. For example, **warping may be applied in a manner that reduces distortion** applied to a particular object in motion across different frames of a video since the viewer is likely to focus attention on the object in motion.

Ex. 1001 (413 patent) at 7:64-8:2; Surreply 8

Ground 4: Claim Construction of “Determine a Portion”

- Applicant amended claims to add that the “portion” is “determine[d]” after obtaining image but before transformation

the one or more physical processors configured by machine-readable instructions to:

obtain the input image;

 within the input image, the portion including a subset of the pixels; and

apply a transformation to the input image to generate an output image, the output image having a target aspect ratio different than the source aspect ratio, the output image including the pixels located at output positions defined along the first axis and the second axis, wherein:

the transformation non-uniformly shifts the pixels from the input positions to the output positions based on (1) the input positions along the first axis, and (2) the input positions along the second axis; and

~~the input image includes  including a subset of the pixels~~

the transformation non-uniformly shifts the pixels such that differences between the input positions and the output positions of the subset of the pixels within the portion are less than differences between the input positions and the output positions of others of the pixels; and

present the output image on the display.

Ex. 1003 (413 Prosecution History) at 32; POR 8-9

Ground 4: Claim Construction of “Determine a Portion”

- Applicant added “determine” step to distinguish art that merely performed a transformation

At best, Oosawa describes facilitating comparison of multiple (e.g., two) images of an identical object by performing a global matching operation between the images and determining the shifting amount of corresponding pixels between the images.

(Oosawa, Abstract, col. 6:6-47) To the extent Oosawa shows the amount of shifting to be applied to the center of the image is smaller than the peripheral of the image, this is merely the result of the visuals captured within two images being compared and is **not result of a transformation that creates less differences between input positions and output positions of the subset of the pixels than the differences between input positions and output positions of other pixels in the input image**. Thus,

Oosawa fails to disclose determining a portion (including a subset of pixels) within an input image and non-uniformly shifting the pixels of the input image such that differences between input positions and output positions are less for the subset of the pixels than other pixels of the input image.

Ex. 1003 (413 Prosecution History) at 38-39; POR 8-9

Ground 4: Claim Construction of “Determine a Portion”

- Dr. Katsaggelos agreed the “determine” step is separate from the “transformation” step

“Q. Those are two different operations that the claim is contemplating, correct?”

A. Yes, there are two different limitations here.”

Ex. 2026 (Katsaggelos Tr.) at 26:10-23; Surreply 7

Ground 4: Meulen Does Not Disclose “Determine a Portion”

- **Petition:**
 - Meulen’s disclosure of a central region with no compression meets the claimed “determine a portion”
- **Reply:**
 - Meulen discloses controlling “center” of compression ***after*** image is obtained
 - Meulen discloses “expand[ing]” regions ***after*** image is obtained
- Petitioner is **incorrect:**
 - Meulen’s portions are ***pre-determined*** before the image is obtained
 - Petitioner’s new Reply arguments mischaracterize Meulen, which discloses that the regions are still ***pre-determined***

Ground 4: Meulen Does Not Disclose “Determine a Portion”

- Meulen does not disclose *determining* the less distorted region *after* obtaining the image
- Meulen’s transformations use *pre-determined* portions

TABLE 1

	Region					Total
	1	2	3	4	5	
Pixels/Region	10	50	200	50	10	320
Compression Ratio	4:1	2:1	none	2:1	4:1	
Source Pixels	40	100	200	100	40	480

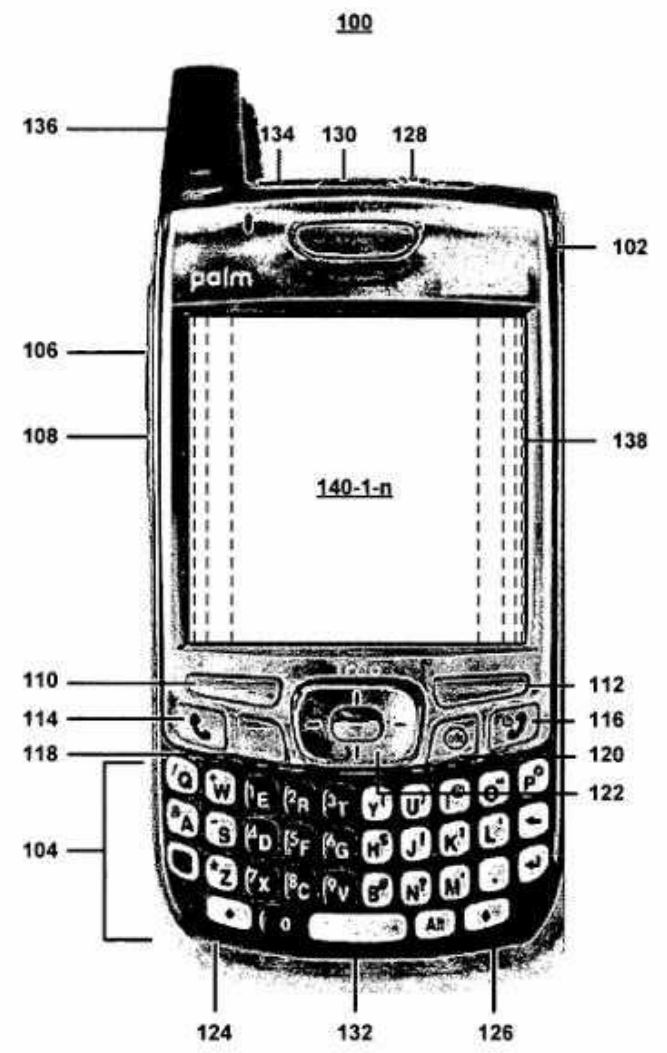
“Table 1 illustrates an implementation for one embodiment of a **display 138** comprising display regions 140-1-5.”

Ex. 1007 (Meulen) at 5:50-60; Reply 11

“Q. You agree that in table 1 region 3 is **allocated in advance** 200 pixels of the display and a compression ratio of zero, correct?

A. **Yeah**, this, again, is an example where this portion that needs to be determined of the input image is represented by this 200 pixels.”

Ex. 2026 (Katsaggelos Tr.) at 21:1-18; POR 21



Ground 4: Meulen Does Not Disclose “Determine a Portion”

- Meulen does not disclose *determining* the less distorted region *after* obtaining the image
- Meulen’s transformations use *pre-determined* portions

TABLE 1

	Region					Total
	1	2	3	4	5	
Pixels/Region	10	50	200	50	10	320
Compression Ratio	4:1	2:1	none	2:1	4:1	
Source Pixels	40	100	200	100	40	480

TABLE 2

	Region									Total
	1	2	3	4	5	6	7	8	9	
Pixels/Region	4	8	16	32	200	32	16	8	4	320
Compression Ratio	7:1	4:1	2:1	3:2	none	3:2	2:1	4:1	7:1	
Source Pixels	28	32	32	48	200	48	32	32	28	480

TABLE 3

	Region									Total
	1	2	3	4	5	6	7	8	9	
Pixels/Region	2	4	8	16	260	16	8	4	2	320
Compression Ratio	14:1	8:1	4:1	3:1	none	3:1	4:1	8:1	14:1	
Source Pixels	28	32	32	48	260	48	32	32	28	480

TABLE 4

	Region									Total
	1	2	3	4	5	6	7	8	9	
Pixels/Region	2	4	8	16	230	32	16	8	4	320
Compression Ratio	14:1	8:1	4:1	3:1	none	3:2	2:1	4:1	7:1	
Source Pixels	28	32	32	48	230	48	32	32	28	480

Ground 4: Meulen Does Not Disclose “Determine a Portion”

- Petition:

d. 2[c]: “determine a portion within the input image, the portion including a subset of the pixels; and”

Meulen’s media source content is an “input image.” (*See* IV.D.3.c.)

“[C]entral display region 140-3 may comprise 200 pixels and require the media source content to undergo no image compression (e.g., 200 source pixels to 200 display pixels).” (EX1007, 6:27-31.) A POSITA would have understood that the central region within the media source content has a corresponding central display region, which includes a subset of the source pixels. (EX1002, ¶¶252-253.)

Pet. 88

Ground 4: Meulen Does Not Disclose “Determine a Portion”

- **Petitioner:** Different compression ratios in different regions requires determination *Reply 3*
- Petitioner is **incorrect:**
 - Meulen describes different **embodiments**
 - Each of Meulen’s embodiments has **pre-determined** regions before the image is obtained

“Q. Those different examples in Meulen that you cited to me, those are all **alternative implementations** of the wireless device, correct?”

A. **Yeah**, examples of embodiments.”

Ex. 2026 (Katsaggelos Tr.) at 23:3-6; POR 21

“Q. You agree that in table 1 region 3 is **allocated in advance** 200 pixels of the display and a compression ratio of zero, correct?”

A. **Yeah**, this, again, is an example where this portion that needs to be determined of the input image is represented by this 200 pixels.”

Ex. 2026 (Katsaggelos Tr.) at 21:1-18; POR 21

Ground 4: Meulen Does Not Disclose “Determine a Portion”

- Petitioner:** Meulen can shift the center region via user control *Reply 8*

“In some embodiments, for example, the center of linearity and/or center of compression for the panoramic display 200 may be shifted to the left and/or to the right under user-control (e.g., navigation button 122) to provide a shifted or magnified effect.”

Ex. 1007 (Meulen) at 9:1-10:6; Reply 8

- Meulen’s compression shifting does not alter the regions, only the compression ratios:

TABLE 2										
	Region									Total
	1	2	3	4	5	6	7	8	9	
Pixels/Region	4	8	16	32	200	32	16	8	4	320
Compression Ratio	7:1	4:1	2:1	3:2	none	3:2	2:1	4:1	7:1	
Source Pixels	28	32	32	48	200	48	32	32	28	480

TABLE 4										
	Region									Total
	1	2	3	4	5	6	7	8	9	
Pixels/Region	2	4	8	16	230	32	16	8	4	320
Compression Ratio	14:1	8:1	4:1	3:1	none	3:2	2:1	4:1	7:1	
Source Pixels	28	32	32	48	230	48	32	32	28	480

“[I]mage compression and/or scaling may be applied to the left lateral display regions 202-1-5 and the right lateral display regions 202-9-6 **asymmetrically**.”

Ex. 1007 (Meulen) at 9:1-3; Surreply 15

Ground 4: Meulen Does Not Disclose “Determine a Portion”

- **Petitioner:** Meulen can “expand” one or more regions *Reply 9*

“In other implementations, the display controller 504 may be arranged to expand one or more display regions to create a magnified effect.”

Ex. 1007 (Meulen) at 12:23-25; Reply 9

- Meulen does not disclose that the regions can be altered prior to transformation.
- The “magnified effect” merely changes the compression/scaling ratio, not the regions

“In some embodiments, for example, the center of linearity and/or center of compression for the panoramic display 200 may be shifted to the left and/or to the right under user-control (e.g., navigation button 122) to provide a shifted or magnified effect.”

Ex. 1007 (Meulen) at 9:1-10:6; Reply 8

Ground 4: Meulen Does Not Disclose “Non-Uniform” Transformation (All Claims)

Ground 4: Claim Construction of “Non-Uniform” Transformation

- Claim Language: “the transformation non-uniformly shifts the pixels”

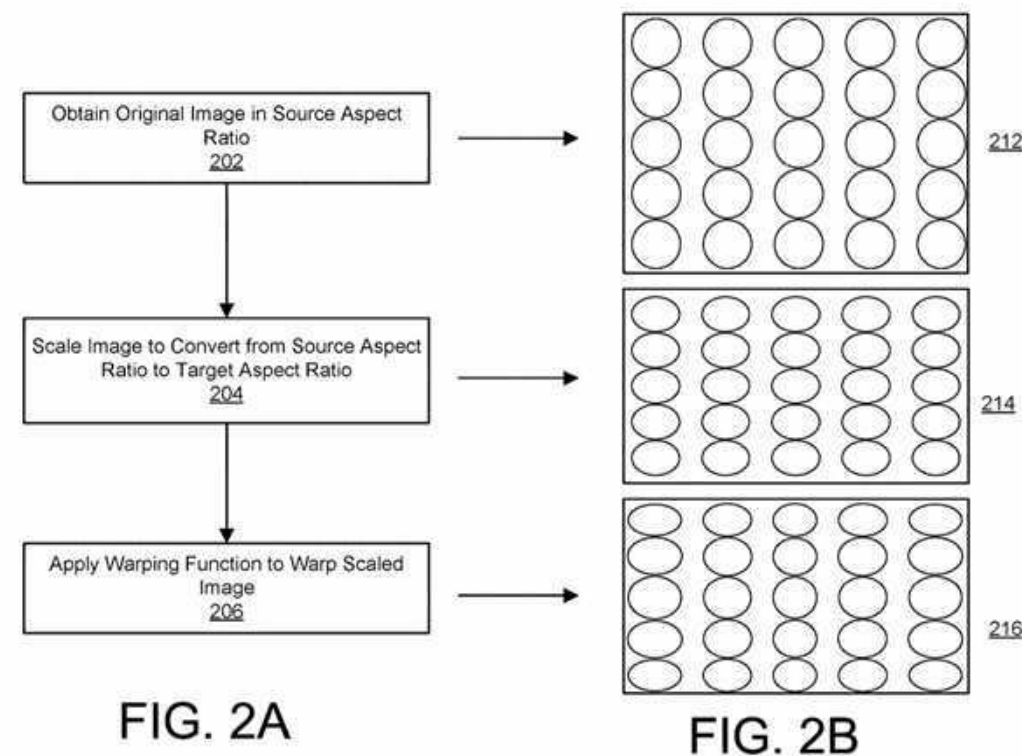
Proposed Construction of “non-uniform”	
Patent Owner	Petitioner
Plain meaning: “non-linear” transformation	“not the same in all parts or areas”

- Petitioner is incorrect:
 - Specification consistently requires “non-linear” transformation
 - Petitioner’s construction includes linear scaling, which the patent distinguished
 - Petitioner’s expert consistently described “non-uniform” transformation as “non-linear”

Ground 4: Claim Construction of “Non-Uniform” Transformation

- **'413 Patent:**

“FIG. 2A illustrates an example embodiment of a process for converting from a source aspect ratio to a target aspect ratio in a camera system. In the process of FIG. 2A, an input image is obtained 202 in the source aspect ratio (e.g., a 4:3 aspect ratio). The image is then scaled 204 from the source aspect ratio (e.g., the 4:3 aspect ratio) to the target aspect ratio (e.g., a 16:9 aspect ratio) using a linear scaling function. A non-linear warping is then applied 206 to the scaled image. FIG. 2B illustrates the process of FIG. 2A as applied to an example image 212 containing an array of circles. As can be seen, the scaling function applied in step 204 causes the circles to be stretched horizontally and compressed vertically as shown in example image 214. Because the scaling function applied in step 204 is linear, each circle is stretched/compressed equally independent of horizontal and vertical position. In the warping step 206, a non-linear warping is applied which results in the circles being stretched/compressed by different amounts depending on their horizontal and vertical positions resulting in example image 216.”



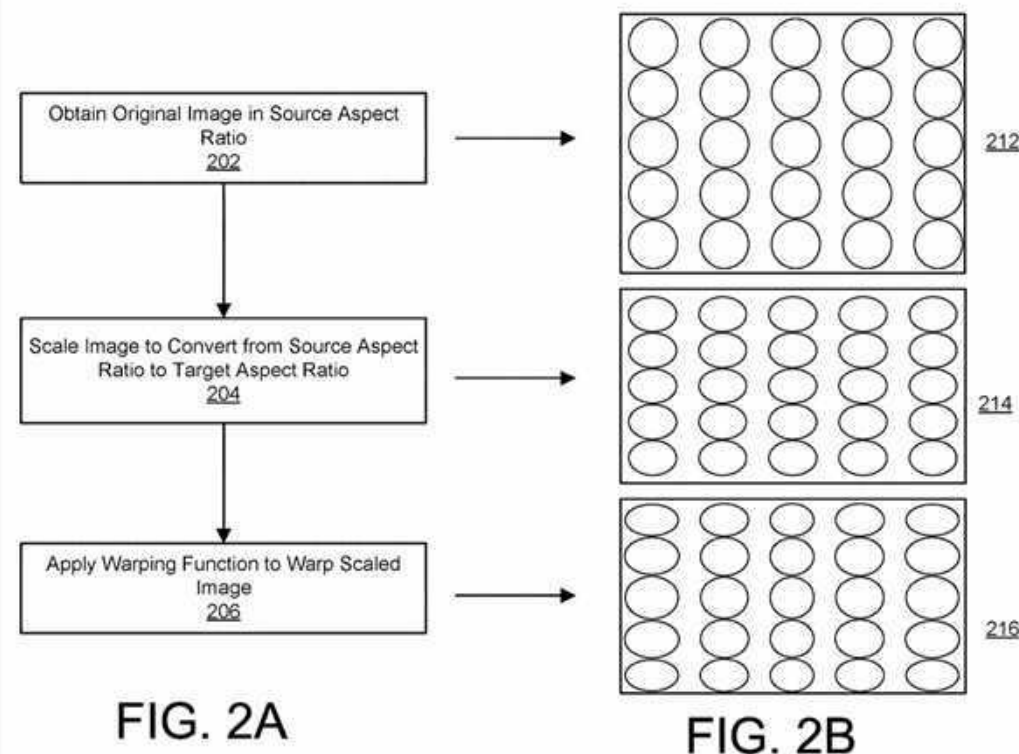
Ex. 1001 (413 patent) at 3:61-4:13; POR 24

Ground 4: Claim Construction of “Non-Uniform” Transformation

- **'413 Patent:**

“As can also be observed from FIG. 2B, the scaling and warping steps 204, 206 preserves the full field of view in the resulting image 216 relative to the original image 212. To achieve this, the warp module is configured so that some areas of the image are fully corrected geometrically, while some other areas are distorted. In the illustrated example, the center of the image is chosen to be the region of interest that is most fully corrected and the distortion increases in each direction as the frame boundary is approached. Thus, in image 216, the center circle is fully corrected so that it maintains the original circular shape of the corresponding circle in the input image 212. In the vertical direction, the circles become more distorted (e.g., more vertically compressed) in the as they approach the top and bottom edges. In the horizontal direction, the circles become more distorted (e.g., more horizontally stretched) as they approach the left and right edges. As a result, the circles in the corners exhibit the most distortion, because they have the highest amount of both vertical compression and horizontal stretching. In other embodiments, a different part of the image may be selected as the area most fully corrected, or in the case, of video, the area may dynamically change between images (e.g., to follow a moving object).”

Ex. 1001 (413 patent) at 4:14-4:36; Pet. 4



Ground 4: Claim Construction of “Non-Uniform” Transformation

“Non-uniform” means “non-linear”

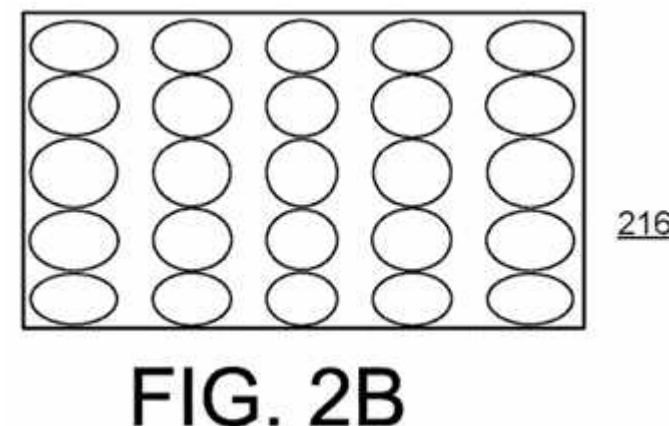
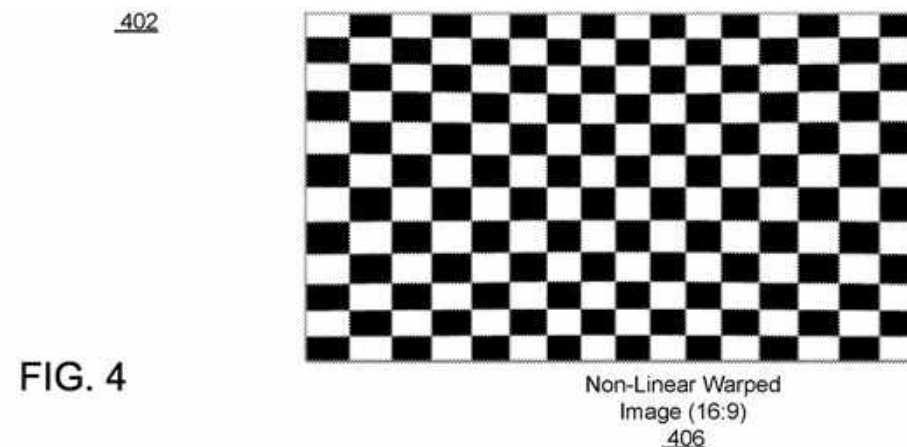
- Specification **only** uses the term “non-linear”, not “non-uniform”

“In one embodiment, the output image is **non-linearly warped** relative to the input image such that a distortion in the output image relative to the input image is greater in a corner region of the output image than a center region of the output image.”

Ex. 1001 (413 patent) at Abstract; Pet. 1

“In one embodiment, the conversion is achieved using a resolution remapping that applies a **non-linear function** across the image so as to minimize or reduce undesirable key feature aspect ratio distortions.”

Ex. 1001 (413 patent) at 3:42-45; POR 71



Ground 4: Claim Construction of “Non-Uniform” Transformation

“Non-uniform” means “non-linear”

- Specification **only** uses the term “non-linear”, not “non-uniform”

“The image is then scaled 204 from the source aspect ratio (e.g., the 4:3 aspect ratio) to the target aspect ratio (e.g., a 16:9 aspect ratio) using a linear scaling function. A **non-linear warping** is then applied 206 to the scaled image.”

Ex. 1001 (413 patent) at 3:65-4:2; POR 24

“In the warping step 206, **a non-linear warping** is applied which results in the circles being stretched/compressed by different amounts depending on their horizontal and vertical positions resulting in example image 216.”

Ex. 1001 (413 patent) at 4:6-13; POR 24

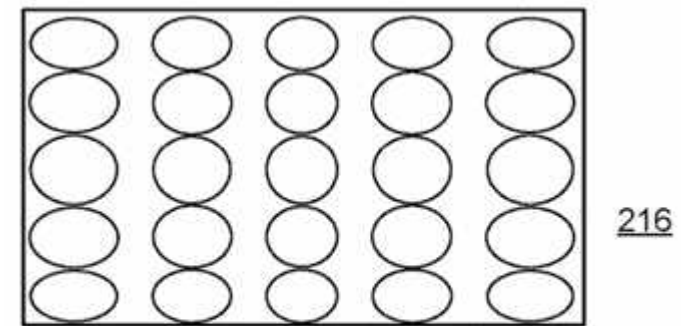
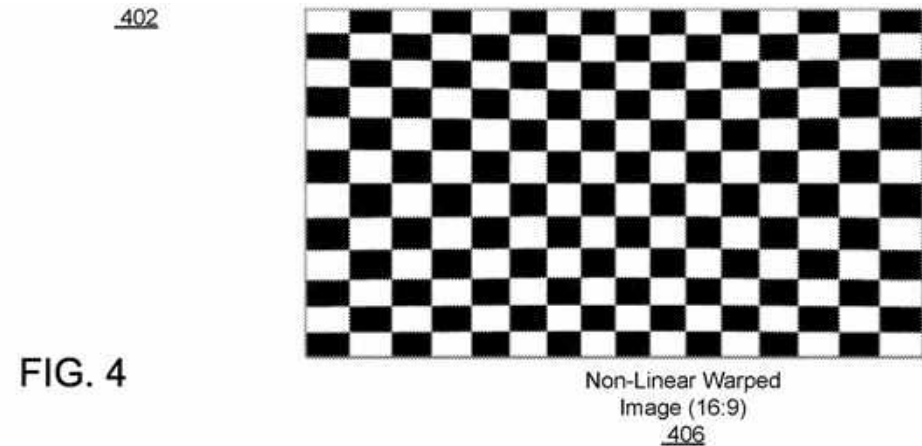


FIG. 2B

Ground 4: Claim Construction of “Non-Uniform” Transformation

“Non-uniform” means “non-linear”

- Claim language and specification equates non-uniform with non-linear:

3. The system of claim 2, wherein the transformation includes:

a scaling function to uniformly stretch or compress the input image along the first axis; and

a warping function to non-uniformly warp the input image along the second axis.

“FIG. 3 is a first set of example images showing an original image in a source aspect ratio, a cropped image converted to a target aspect ratio using a cropping technique, and a non-linear warped image converted to the target aspect ratio using a non-linear warping technique.”

Ex. 1001 (413 patent) at 1:36-40; Surreply 1-2

Ground 4: Claim Construction of “Non-Uniform” Transformation

“Non-uniform” means “non-linear”

- Petitioner’s expert, Dr. Katsaggelos:

“Q. Turning back to the limitation about the non-uniform shifting of pixels, would you agree that in order to accomplish a non-uniform shifting, **you need to apply a non-linear transformation**?”

A. **Yeah**, the application of a non-linear transformation results in the non-vertical shifting of the pixels.”

Ex. 2026 (Katsaggelos Tr.) at 34:7-13; POR 23

“Q. Is it technically possible to have a system that **non-uniformly shifts** the pixels, as were cited in the ‘413 patent, without determining a portion?”

A. It is possible to determine a **non-linear transformation** that determines which parts of the image are shifted which way ...”

Ex. 2026 (Katsaggelos Tr.) at 29:9-17; Surreply 2

“Well, the transformation shifts non-uniformly. The transformation – It is not the transformation that is non-uniform. It’s **the shifts that are non-uniform**. The transformations – The **non-linear transformation** results in non-vertical shifts.”

Ex. 2026 (Katsaggelos Tr.) at 29:9-17; Surreply 2

“It says **non-uniformly warp** the input image. So it seems to **imply a non-linear transformation** applied to the image.”

Ex. 2026 (Katsaggelos Tr.) at 83:19-84:15; Surreply 2

Ground 4: Claim Construction of “Non-Uniform” Transformation

- Reply Declaration:

“I did not agree in my deposition that ‘non-uniform[] shifting’ requires a non-linear transformation, or that the two terms have the same meaning. ... In other words, a non-linear transformation is a type of, but is not the same as, non-uniform shifting.”

Ex. 1014 (Katsaggelos Supp. Decl.) at ¶8

- First Deposition:

“Q. Turning back to the limitation about the non-uniform shifting of pixels, would you agree that in order to accomplish a non-uniform shifting, you need to apply a non-linear transformation?”

A. Yeah, the application of a non-linear transformation results in the non-vertical shifting of the pixels.”

- Second Deposition:

“Q. And you agree that Figure 2B is an embodiment of a non-uniform transformation?”

A. I’m sorry. Non-uniform transformation?

Q. Yes, sir.

A. I’m not sure if I know exactly what a non-uniform transformation is.”

Ex. 2042 (2nd Katsaggelos Tr.) at 11:22-12:2; Surreply 6

Ground 4: Claim Construction of “Non-Uniform” Transformation

- **Petitioner:** Dictionary supports our construction

non-uniform

adjective (also nonuniform)

UK  /ˌnɒnˈjuːnɪfɔːm/ US  /ˌnɑːnˈjuːnəfɔːrm/

non-uniform adjective (NOT SAME)

not the same in all parts or areas:

Ex. 1016 (M-W Dictionary); Reply 1

- Dictionary evidence not relevant when intrinsic evidence aligns with expert testimony



“In most situations, an analysis of the intrinsic evidence alone will resolve any ambiguity in a disputed claim term. In such circumstances, it is improper to rely on extrinsic evidence.”

Vitronics Corp. v. Conceptor, Inc., 90 F.3d 1576, 1583 (Fed. Cir. 1996); Surreply 4

Ground 4: Claim Construction of “Non-Uniform” Transformation

- **Petitioner:** ‘413 specification discloses “nonuniform” transformation can be “linear.” *Reply 1*

PO argues that “non-uniform” should be limited to “non-linear”—*i.e.*, “[n]o two pixels will have the same amount of shift.” (POR, 6, 28.) But PO’s definition is inconsistent with the specification. *Phillips v. AWH Corp.*, 415 F.3d 1303, 1315 (Fed. Cir. 2005) (en banc). The specification discloses non-uniform shifting as “using a linear scaling function” and then applying “non-linear warping.” (EX1001, 3:61-4:2; POR, 24 (PO agreeing that the ‘413 claims encompass this embodiment).) Thus, “non-uniform” *cannot* be “non-linear” because, it may *also* involve a *linear* scaling. (EX1014, ¶¶8-11.)

Reply 1

“In the process of FIG. 2A, an input image is obtained 202 in the source aspect ratio (e.g., a 4:3 aspect ratio). The image is then scaled 204 from the source aspect ratio (e.g., the 4:3 aspect ratio) to the target aspect ratio (e.g., a 16:9 aspect ratio) using a **linear scaling function**. A **non-linear warping is then applied** 206 to the scaled image.”

Ex. 1001 (413 Patent) at 3:61-4:2; Reply 1

“As can be seen, the scaling function applied in step 204 causes the circles to be stretched horizontally and compressed vertically as shown in example image 214. Because the scaling function applied in step 204 is linear, each circle is stretched/compressed equally independent of horizontal and vertical position. **In the warping step 206, a non-linear warping is applied which results in the circles being stretched/compressed by different amounts** depending on their horizontal and vertical positions resulting in example image 216.”

Ex. 1001 (413 Patent) at 4:3-13; POR 40

Ground 4: Claim Construction of “Non-Uniform” Transformation

- **Petitioner:** ‘413 specification discloses “nonuniform” transformation can be “linear.” *Reply 1*
- **Incorrect:**

“In the process of FIG. 2A, an input image is obtained 202 in the source aspect ratio (e.g., a 4:3 aspect ratio). The image is then scaled 204 from the source aspect ratio (e.g., the 4:3 aspect ratio) to the target aspect ratio (e.g., a 16:9 aspect ratio) using a linear scaling function. A non-linear warping is then applied 206 to the scaled image.”

Ex. 1001 (413 Patent) at 3:61-4:2; Reply 1

3. The system of claim 2, wherein the transformation includes:

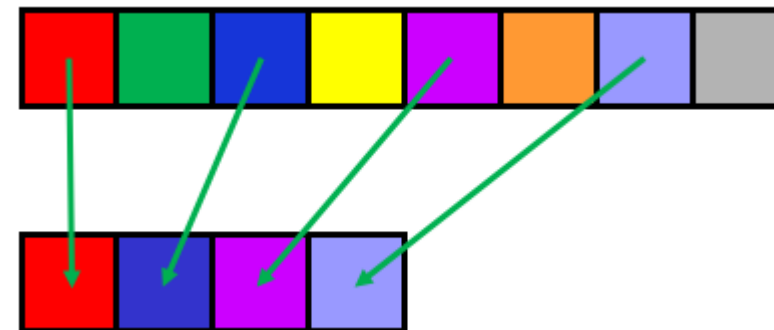
- a scaling function to uniformly stretch or compress the input image along the first axis; and
- a warping function to non-uniformly warp the input image along the second axis.

Ground 4: Claim Construction of “Non-Uniform” Transformation

- **Petitioner:** “non-uniform” shifting of pixels includes linear scaling

“I note that even the shift is not the same for pixels that have the same compression, e.g., the shift of the third pixel (blue) is different from the shift of the seventh pixel (purple), both pixels have a 2:1 compression.”

Ex. 1014 (Katsaggelos Supp. Decl.) at ¶¶56-57; Reply 13



- Petitioner is **incorrect**:

“Conventional conversion techniques such as cropping, linearly scaling, and padding each result in a perceivable reduction in the quality of the image or video. ... Linear scaling maintains the full field of view but introduces perceivable distortion into the image. For example, if an image is vertically compressed and/or horizontally stretched to convert from a 4:3 to a 16:9 aspect ratio, features in scene will appear “short and fat” relative to the original image.

Ex. 1001 (413 Patent) at 1:12-21; Surreply 6

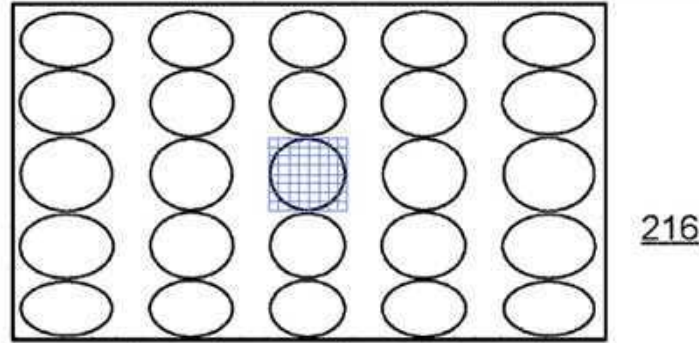
“In some instances, a POSITA would not have preferred cropping, padding or linear scaling over non-uniform shifting.”

Ex. 1014 (Katsaggelos Supp. Decl.) at ¶39; Surreply 6

Ground 4: Claim Construction of “Non-Uniform” Transformation

- **Petitioner:** Pixels in center of Figure 2B have zero shift, lacking non-linearity.

Reply 1



- **Incorrect:**

determine a portion within the input image, the portion including a subset of the pixels; and

the transformation non-uniformly shifts the pixels such that differences between the input positions and the output positions of the subset of the pixels within the portion are less than differences between the input positions and the output positions of others of the pixels; and

Ground 4: Meulen Does Not Disclose “Non-Uniform” Transformation

- **Petitioner:** Meulen’s piecewise linear transformation is a non-linear transformation
- Petitioner is **incorrect:**
 - Piecewise linear is only an approximation of a non-linear transformation
 - Petitioner’s expert agreed piecewise linear transformation is used to avoid the complexities of non-linear transformation

Ground 4: Meulen Does Not Disclose “Non-Uniform” Transformation

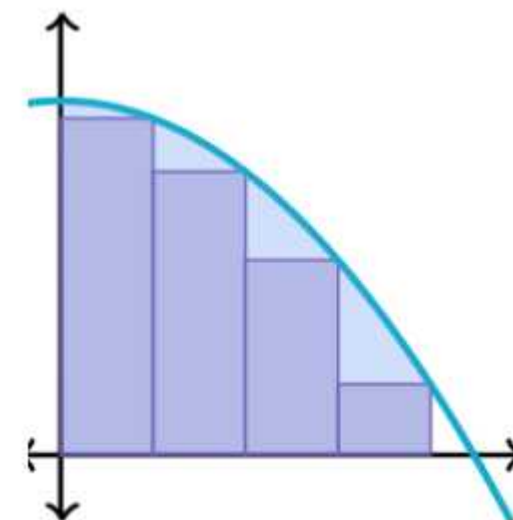
- Piece-wise linear transformation is only an ***approximation*** of a non-linear transformation:

“A piece-wise linear transformation is different from a non-linear transformation. It involves **estimating a non-linear transformation** by simplifying it to several linear transformations. It can be analogized to estimating a curve by using several straight lines. A classic example is using multiple rectangles to find the area under a curve. Instead of performing a non-linear calculation, the area is divided into multiple rectangles whose areas are significantly easier to calculate using a linear calculation (*i.e.* height * width). Since there are multiple rectangles, this constitutes a piece-wise linear transformation.”

Ex. 2027 (Foroosh Decl.) at ¶71; POR 24

“[F]or an approximation to be considered an approximation, it would be defined as an approximation **which has to have an error**. And if I want to approximate a highly non-linear function with this linear pieces, then I have to have a fine linear approximation to approximate a non-linear function.”

Ex. 2026 (Katsaggelos Tr.) at 42:16-43:3; POR 26



Ground 4: Meulen Does Not Disclose “Non-Uniform” Transformation

- Petitioner’s Expert (Dr. Katsaggelos): Piece-wise linear is “**computationally indistinguishable**” from linear transformation:

“I can have a non-linear solution with a piecewise linear approach, right? So its computation is **indistinguishable from a linear approximation**, but it doesn’t prevent a non-linear function.”

Ex. 2026 (Katsaggelos Tr.) at 40:19-41:5; POR 26

“Q. Did you say piecewise linear is computationally indistinguishable from a linear transformation?

A. It’s very similar. It’s very similar.”

Ex. 2026 (Katsaggelos Tr.) at 41:6-41:8; POR 26

- Dr. Katsaggelos: piece-wise linear avoids complexity of non-linear transformation:

“Q. [A]re you familiar with the technique of applying a **linear approximation** in order to **avoid the complexities of a non-linear calculation**? Are you familiar with that notion?

A. Yeah, that notion exists.”

Ex. 2026 (Katsaggelos Tr.) at 40:1-7; POR 27-28

Ground 4: Meulen Does Not Disclose “Non-Uniform” Transformation

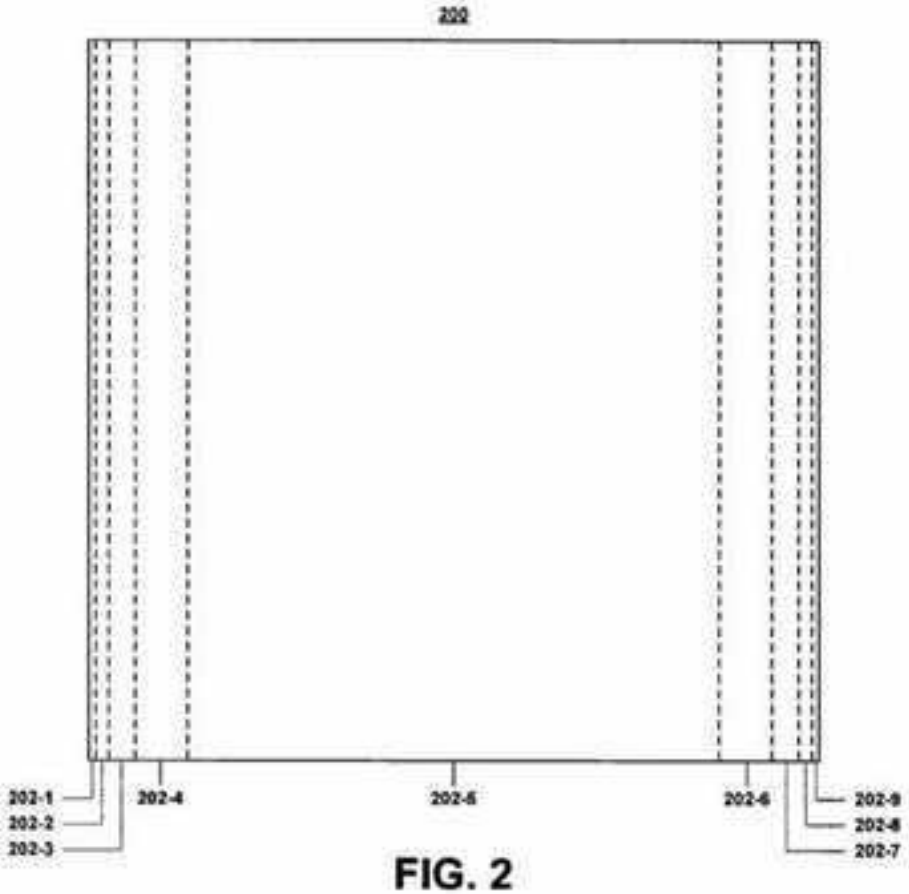
- Undisputed: Meulen performs a piece-wise *linear* transformation *POR 24, Reply 11*
- Each region in Meulen uses linear compression:

TABLE 1

	Region					Total
	1	2	3	4	5	
Pixels/Region	10	50	200	50	10	320
Compression Ratio	4:1	2:1	none	2:1	4:1	
Source Pixels	40	100	200	100	40	480

TABLE 2

	Region									Total
	1	2	3	4	5	6	7	8	9	
Pixels/Region	4	8	16	32	200	32	16	8	4	320
Source Pixels	40	80	160	320	2000	320	160	80	40	4800



Ground 4: Meulen Does Not Disclose “Non-Uniform” Transformation

- Linear piecewise avoids complexity of non-linear transformation:
- ‘413 Patent’s non-linear transformations:

$$\Delta h/H = 0.1107h^6 + 0.534h^5 - 1.7813h^4 + 1.3167h^3 - 0.0722h^2 - 0.1073h$$

$$\Delta v/V_0 = -0.0749v^6 - 0.3167v^5 + 1.0939v^4 - 0.8113v^3 + 0.0408v^2 + 0.0678v$$

$$\Delta v/V_{10} = -0.0658v^6 - 0.2785v^5 + 0.9617v^4 - 0.7132v^3 + 0.0359v^2 + 0.0596v$$

$$\Delta v/V_{20} = -0.0568v^6 - 0.2402v^5 + 0.8294v^4 - 0.6152v^3 + 0.0309v^2 + 0.0514v$$

Ex. 1001 (413 Patent) at 7:3-42; POR 26

- Dr. Katsaggelos:

“Q. [A]re you familiar with the technique of applying a linear approximation in order to avoid the complexities of a non-linear calculation? Are you familiar with that notion?

A. Yeah, that notion exists.”

Ex. 2026 (Katsaggelos Tr.) at 40:1-7; POR 27-28

“[F]or an approximation to be considered an approximation, it would be defined as an approximation which has to have an error. And if I want to approximate a highly non-linear function with this linear pieces, then I have to have a fine linear approximation to approximate a non-linear function.”

Ex. 2026 (Katsaggelos Tr.) at 42:16-43:3; POR 26

Ground 4: Meulen Does Not Disclose “Non-Uniform” Transformation

- **Petitioner:** Meulen’s Figures 4A-4C show non-linear curves

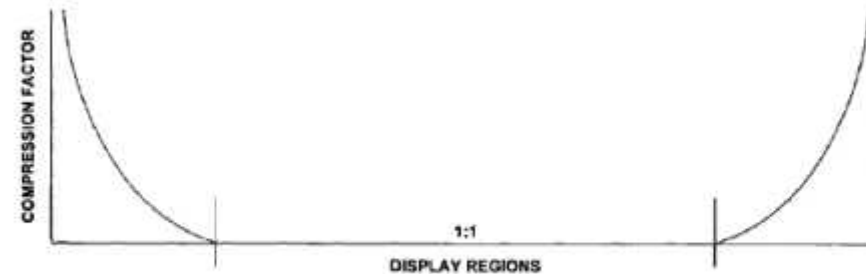


FIG. 4A

- These are still piecewise linear with a higher number of regions, but still an approximation:

“In this embodiment, a panoramic display may be implemented by employing significantly more display regions (e.g., >30)”

Ex. 1007 (Meulen) at 10:44-48; Surreply 17

Ground 4: Meulen and Chao Do Not Disclose “Shift[ing] The Pixels” (All Claims)

Ground 4: Meulen and Chao Do Not Disclose Pixel “Shifting”

- **Petitioner:**
 - Meulen’s compression can be performed using pixel shifting
 - Chao describes object shifting
- Petitioner is **incorrect:**
 - Meulen only discloses compression using “polyphase filters,” which is indisputably not pixel shifting
 - Petitioner’s example of “sub-sampling” is not found in Meulen
 - Chao does not describe pixel shifting, much less non-uniform shifting
 - Petitioner provides no reason to apply Chao’s object shifting to Meulen’s compression

Ground 4: Meulen Does Not Disclose Pixel “Shifting”

- Meulen’s compression uses “polyphase filtering”

“[T]he wireless device 100 may comprise sample-rate converters implemented in hardware by **programmable polyphase filters** arranged to sample a line of source pixels and to **generate a compressed and/or scaled** line of display pixels based on a compression factor.

Ex. 1007 (Meulen) at 5:19-23; POR 31

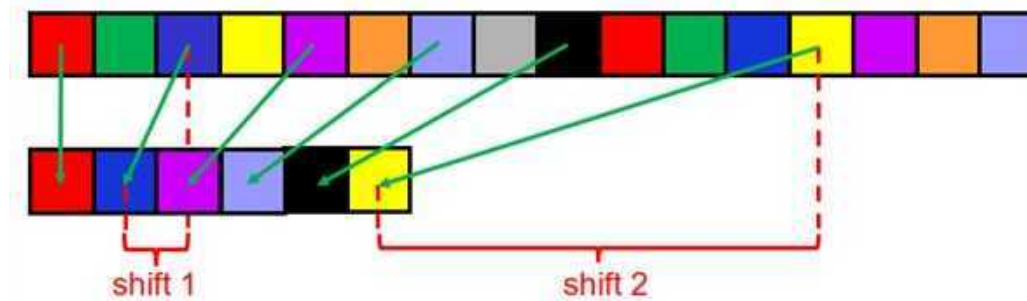
- **Undisputed:** polyphase filtering is ***not*** pixel shifting

“A polyphase filter involves combining a number of pixels into one pixel using multiple phases that can be reused for subsequent pixels. Thus, rather than shift pixels, Meulen’s polyphase filters is better characterized as blending multiple pixels into one pixel.”

Ex. 2027 (Foroosh Decl.) at ¶81; POR 31

Ground 4: Meulen Does Not Disclose Pixel “Shifting”

- **Petitioner:** Meulen’s compression can be performed using “sub-sampling” *Reply*
13-14



Reply 14

- Petitioner admits Meulen does not describe sub-sampling

“Meulen mentions scaling, but does not disclose the details of how to perform scaling.”

Pet. 91

Ground 4: Meulen Does Not Disclose Pixel “Shifting”

- **Petitioner:** ‘413 patent describes compression using pixel shifting *Reply 14*
- Using the ‘413 patent is hindsight-driven
- Meulen’s use of the broader term “compression” does not disclose the narrower term of non-linear pixel-shifting



“It is well established that the disclosure of a genus in the prior art is not necessarily a disclosure of every species that is a member of that genus.”

Sanofi-Synthelabo v. Apotex, Inc., 550 F.3d 1075, 1084 (Fed. Cir. 2008); Surreply 19

Ground 4: Chao Does Not Disclose Pixel “Shifting”

- ***Petition:*** Chao’s background “stretching” or “shrinking” means pixel shifting

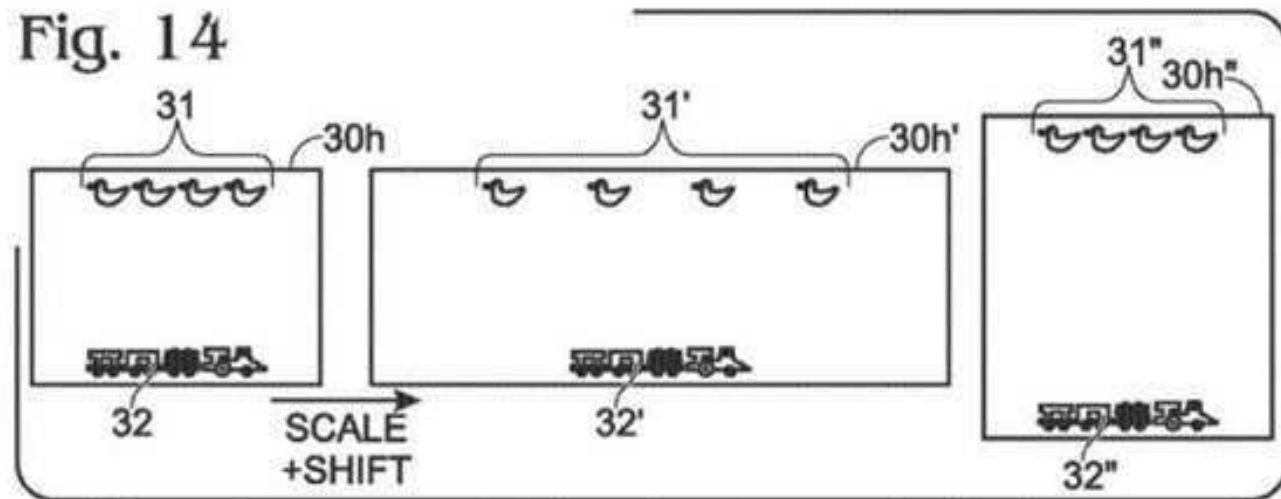
“[T]he **background** element is re-sized, by **stretching or shrinking**, as necessary, to fit the new page size.” (EX1008, [0028].) Chao explains: “the terms ‘**stretch**’ and ‘**shrink**’ are intended to mean the resizing of an element without regard to the original aspect ratio of the image. Accordingly, a ‘stretched’ element may be elongated in only one direction, i.e. may be resized such that it is wider than the original element in the horizontal direction.” (EX1008, [0063].) A POSITA would understand that **stretching/shrinking means shifting the pixels’ positions**. (EX1002, ¶263.)”

Pet. 91-92

- Chao ***never*** describes that its background compression involves pixel shifting

Ground 4: Chao Does Not Disclose Pixel “Shifting”

- **Reply:** Chao’s foreground object shifting is pixel shifting *Reply 14-15*



“According to one embodiment, the aspect ratio of all the elements in the foreground layer is preserved by selecting a **uniform scaling factor** along both the horizontal and vertical directions.”

Ex. 1008 (Chao) at ¶93; POR 34

- Object shifting does not require pixel shifting:

“Chao never describes that stretching/shrinking involves shifting pixels, i.e. moving pixels from one position to a new position as opposed to adding/subtracting pixels through an **interpolation or compression process**.”

Ex. 2027 (Foroosh Decl.) at ¶85; POR 32

- Chao’s object shifting is uniform

Ground 4: No Motivation to Combine Meulen and Chao

- Petitioner never explains how to combine Chao's object shifting with Meulen's compression

“A POSITA would understand that stretching/shrinking means shifting the pixels' positions. (EX1002, ¶263.)

A POSITA would have understood that Meulen's rows are along a horizontal direction. Meulen-Chao involves combining prior art elements and known techniques, resulting in shifting the pixels based on input positions along the horizontal axis. (EX1002, ¶263.)”

Pet. 92

- Petition provided no motivation to combine Meulen and Chao

- Reply:**

“Instead, the Petition explained that Meulen mentions scaling, but does not disclose the details of how to perform scaling. (Petition, 91.) Because Meulen lacks such details, a POSITA would have been motivated to look to another reference such as Chao for such details.”

Reply 17

- Meulen already describes polyphase filtering, and Petitioner provides no reason to use anything else

Ground 4: No Motivation to Combine Meulen and Chao

No motivation to combine as Meulen and Chao are directed to *different fields*

- Meulen directed to modifying media content for display on a screen:

“Various embodiments may be directed to systems and techniques that may potentially improve the performance of a **wireless device to display content from a media source**.”

Ex. 1007 (Meulen) at 1:44-47; Pet. 98

- Chao directed to changing aspect ratio of a graphic design for printing:

“For example, people frequently like to incorporate **professionally designed borders** in their **photos, newsletters, postcards, greeting cards** and the like. As these images become more and more complex, they frequently incorporate elements that are designed for a specific page size. However, given the multitude of graphic design programs that allow people to resize their documents to any desired size, it would be extremely difficult, if not impossible for a graphic designer to create different versions of the same design for each possible page size.”

Ex. 1008 (Chao) at ¶1; POR 37

Ground 4: No Motivation to Combine Meulen and Chao

No motivation to combine as Meulen and Chao have *different goals*

- Meulen wants to avoid black bars on a screen:

“Conventional best fit algorithms such as those defined by the Synchronized Multimedia Integration Language (SMIL) specification may be suitable for fitting content to non-square displays, but **produce a black bar on square displays**. Consequently, there may be a need for improved display systems and techniques for a wireless device.”

Ex. 1007 (Meulen) at 1:26-31; POR 37

- Chao wants to avoid distortion of stylized graphic borders:

“For example, FIG. 1 depicts an original image 10 that has been resized from an original 4×5 template size to a 4×10 template size (at 12). As shown, the ducks that are used as upper border elements 31 have been stretched to fit the width of the image size, distorting the original proportions and overall look of the ducks.”

Ex. 1008 (Chao) at ¶12; POR 37

Ground 4: No Motivation to Combine Meulen and Chao

No motivation to combine as Meulen and Chao have *different approaches*

- Meulen divides an image into regions and performs compression to avoid central distortion:

“For example, more image compression and/or scaling may be applied to the media source content that is to be displayed closer to edges of the display 138, than to the media source content that is to be displayed closer to the center of the display 138.”

Ex. 1007 (Meulen) at 4:45-49; POR 37-38

- Chao shifts foreground elements while stretching/shrinking background:

“The method comprises separating the foreground elements from the background elements, determining an original aspect ratio of the foreground elements and creating resized foreground elements by scaling the foreground elements while maintaining the original aspect ratio.”

Ex. 1008 (Chao) at Abstract; Pet. 83

- Petitioner provides no explanation for incorporating Chao’s foreground / background transformation to Meulen

Ground 4: Meulen Does Not Disclose Non-Uniform Shifting on Both Axes (All Claims)

Ground 4: Meulen Does Not Disclose “Non-Uniform” Shifting Along Both Axes

- **Claim language:** “the transformation non-uniformly shifts the pixels from the input positions to the output positions **based on** (1) the input positions along the **first axis**, and (2) the input positions along the **second axis**”
- **Petitioner:**
 - Claims do not require non-uniform shifting along both axes
 - Meulen describes non-uniform shifting along both axes
- Petitioner is **incorrect:**
 - The transformation must shift non-uniformly along **both** axes
 - Meulen only discloses shifting along a single axis

Ground 4: Meulen Does Not Disclose “Non-Uniform” Shifting Along Both Axes

- Claims:

the transformation non-uniformly shifts the pixels from the input positions to the output positions based on (1) the input positions along the first axis, and (2) the input positions along the second axis; and

- Spec:

“In one embodiment, the warping function involves both a horizontal and vertical warping. For convenience of description, the horizontal and vertical warping is described as two steps, e.g., first horizontal then vertical.”

Ex. 1001 (413 Patent) at 5:29-32; Surreply 23

“An example of a **horizontal warping** function is illustrated in FIG. 5. In this graph, the horizontal coordinate of a pixel is expressed as a percentage of the horizontal pixel position relative to the horizontal width of the image, where the percentage increases in a direction away from a reference edge. ... In this embodiment, the horizontal warp is function of the horizontal coordinate only and is independent of vertical position.”

“In one embodiment, a similarly shaped function is used for **vertical warping**. In another embodiment, the vertical warping function is modified so that the amount of compensation in the vertical direction also depends on horizontal position.”

Ex. 1001 (413 Patent) at 6:13-17; Surreply 23, Reply 3, Ex. 1014 at ¶13

“The vertical warp distortion, in one embodiment, is a function of both the horizontal and vertical positions.”

Ex. 1001 (413 Patent) at 5:38-51; Reply 19, Ex. 1014 (Katsaggelos Supp. Decl.) at ¶71

Ex. 1001 (413 Patent) at 7:5-6; POR 7

Ground 4: Meulen Does Not Disclose “Non-Uniform” Shifting Along Both Axes

- Petitioner:

“This interpretation is consistent with the ’413 embodiments that shift along the horizontal axis only. (EX1001, 5:49-51 (“the horizontal warp is [a] function of the horizontal coordinate only and is independent of [the] vertical position”), 6:65-66 (“the horizontal warp distortion ... is a function of the horizontal position only”))”

Reply 19

- ‘415 Specification:

“In one embodiment, the warping function involves both a horizontal and vertical warping. For convenience of description, the horizontal and vertical warping is described as two steps, e.g., first horizontal then vertical.”

Ex. 1001 (413 Patent) at 5:29-32; Surreply 23

“In an alternative embodiment, the scaling and warping functions applied in steps 204, 206 could be performed in reverse order, such that the warping function in step 206 is applied first followed by scaling in step 204. In another embodiment, the scaling and warping are combined into a single function such that only a single transformation is applied to the image that achieves both the scaling and warping.”

Ex. 1001 (413 Patent) at 4:47-54; Surreply 23

“In another embodiment, the vertical warping function is modified so that the amount of compensation in the vertical direction also depends on horizontal position.”

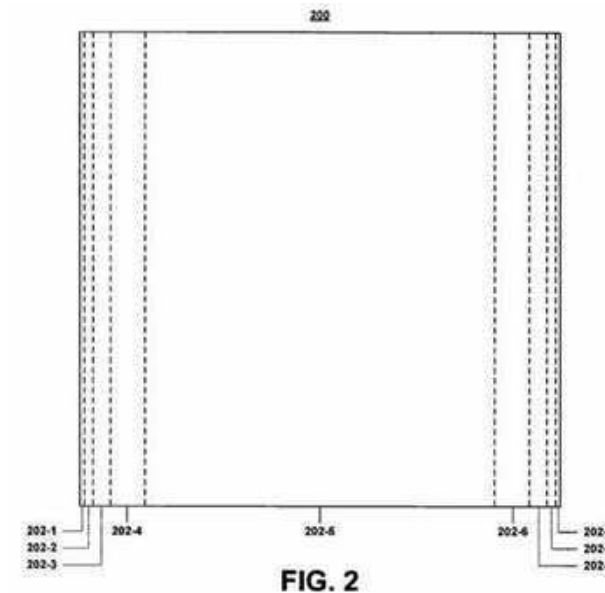
Ex. 1001 (413 Patent) at 6:14-17; Surreply 23

Ground 4: Meulen Does Not Disclose “Non-Uniform” Shifting Along Both Axes

- Meulen:

“The display regions 140-1-5 may be defined by the 320 horizontal pixels of the display 138. The display regions 140-1-5 may be defined by the 320 horizontal pixels of the display 138. Because the media source content has a different aspect ratio and/or resolution than the display 138, image compression and/or scaling may be applied to the media source that is to be displayed in one or more of the display regions 140-1-5.”

Ex. 1007 (Meulen) at 5:65-67; POR 39



Ground 4: Meulen Does Not Disclose “Non-Uniform” Shifting Along Both Axes

- **Petitioner:** Meulen describes “image compression and/or scaling” along both axes

“In some embodiments, image compression and/or scaling may be applied in a horizontal direction and/or in a vertical direction. When applied in both the horizontal and vertical direction, the image compression and/or scaling may allow expansion of the lateral display regions 202-1-4 and 202-6-9 to create a magnified effect or to simulate a video graphics array (VGA) having a resolution of 640×480 with a display having a resolution of 320×320.”

Ex. 1007 (Meulen) at 10:19-26; Pet. 91

- Meulen:

“In such embodiments, image compression and/or scaling may be applied to the media source content that is to be displayed in one or more of the display regions 140-1-n. The image compression and/or scaling may comprise visual scaling with a factor smaller than one in order to reduce size.”

Ex. 1007 (Meulen) at 4:37-41; Surreply 24

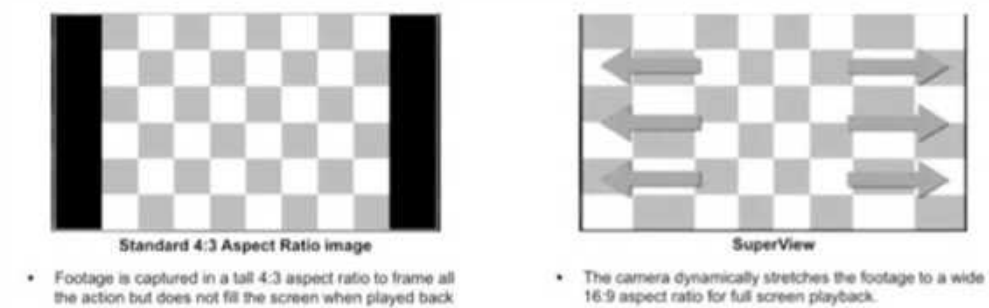
Objective Indicia of Non-Obviousness

Objective Indicia of Non-Obviousness

- There is strong objective indicia of non-obviousness:
- **Practice**: Undisputed that GoPro's SuperView practices the '413 patent
- **Nexus**: Customer surveys show that SuperView contributed to at least some of GoPro's success
- **Commercial Success and Industry Praise**: GoPro's products were successful and had industry praise
- **Copying**: Evidence of copying by competitors
- **Long-felt Need and Unexpected Results**: '413 Patent introduced distortion, leading to unexpected result of increased immersiveness

Objective Indicia of Non-Obviousness: Practice

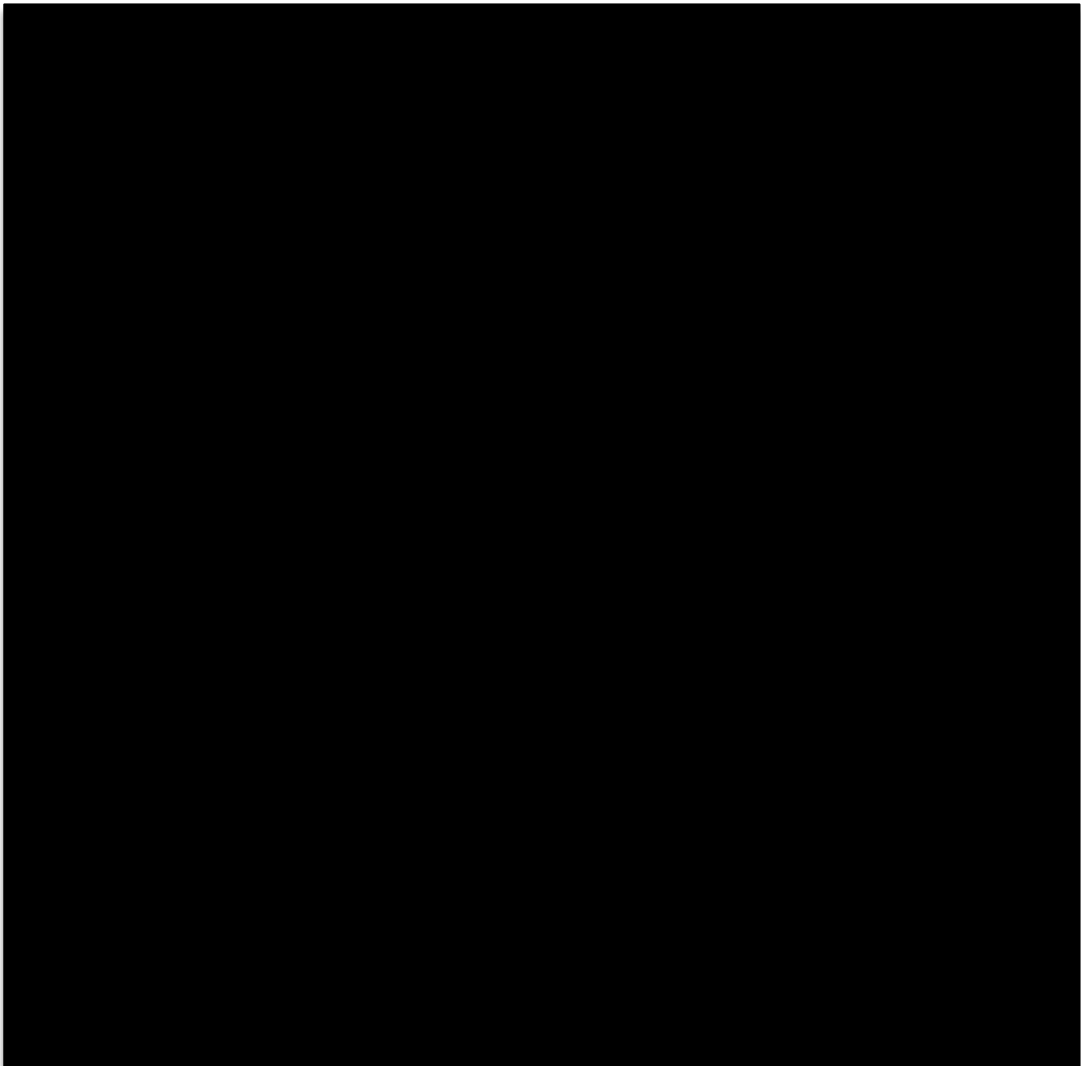
- Undisputed: GoPro's HERO7 Black with SuperView practices the '413 patent



Ex. 2028 (Hero 7 Black Manual) at 47; POR 46



Ex. 2021 (SuperView YouTube video); POR 59



Ex. 2027 (Foroosh Decl.) at ¶144; POR 46

Objective Indicia of Non-Obviousness: Nexus

- Mr. Nema testified about SuperView's effect on HERO7's success

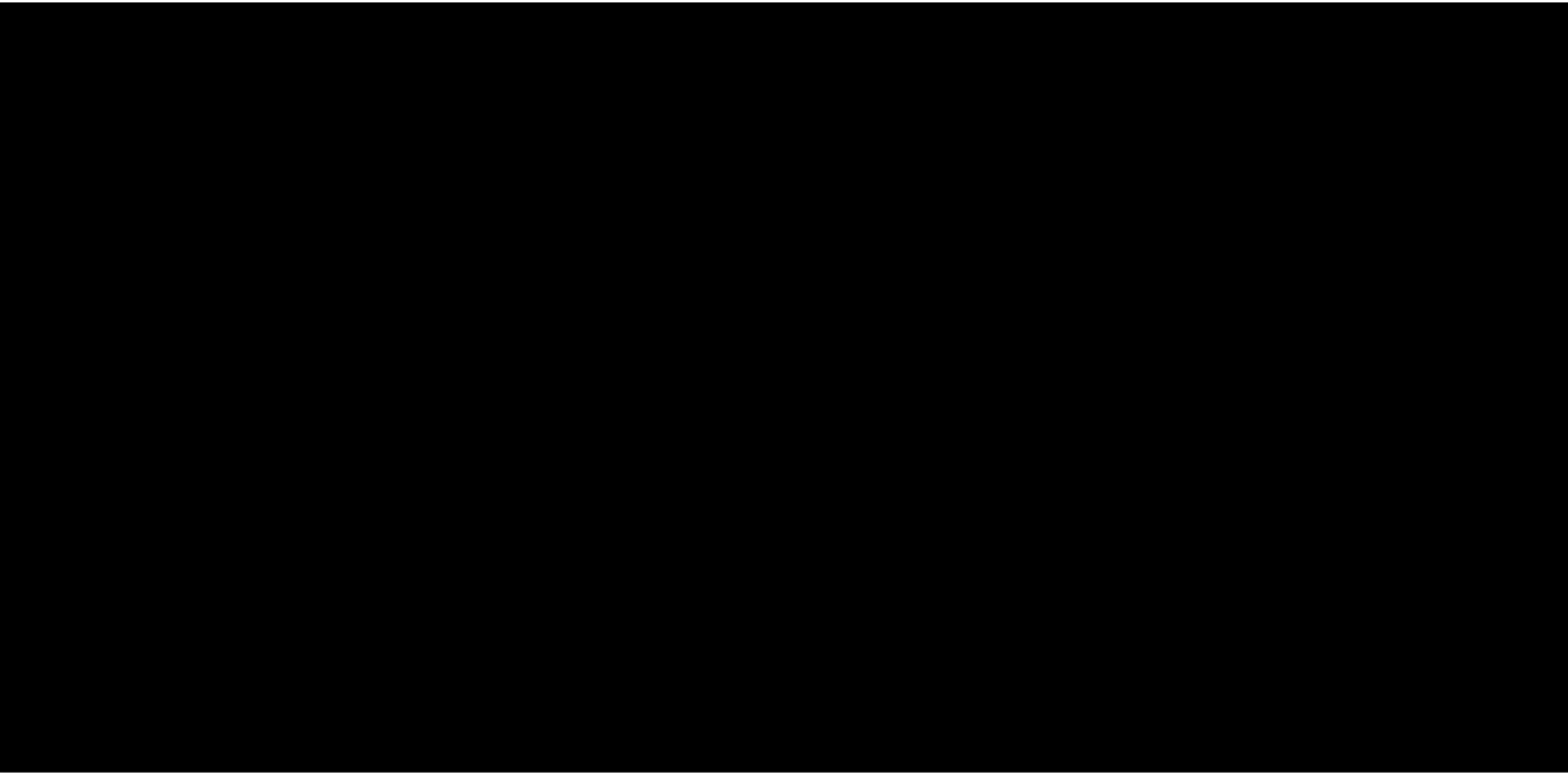
“In particular, a combination of features introduced in the HERO7 Black camera in September 2018 fundamentally transformed both GoPro and the broader action camera market. These features centered around the release of our first, in house developed digital image stabilization system which we branded as “HyperSmooth.” HyperSmooth was and is enabled by our preexisting virtual lens technology which provides various wide-angle immersive video options. HyperSmooth was and is further enhanced by a preexisting aspect-ratio transformation technology that we have marketed since 2013 as “SuperView” (an enhanced version of SuperView was later referred to as “HyperView”).”

Ex. 2024 (Lema Decl.) at ¶¶6; POR 49

“HERO7 Black was the first GoPro product to bring the combined HyperSmooth, SuperView, and digital lens package of features for customers (Horizon Leveling was released 2 years later with HERO9 Black). Internally, we considered HERO7 Black to represent a clear before-and-after moment for GoPro and for the action camera market more broadly.”

Ex. 2024 (Lema Decl.) at ¶¶10; POR 49

Objective Indicia of Non-Obviousness: Nexus



Objective Indicia of Non-Obviousness: Nexus

- ***Petiitioner***: SuperView was just one of many features in the HERO 7 so lacks nexus *Reply 29-30*
- Petitioner is **incorrect**
 - Petitioner does not rebut Mr. Lema's testimony that SuperView contributed to HERO 7's success and praise.
 - It is not necessary for SuperView to be the only feature



“It is not necessary ... that the patented invention be solely responsible for the commercial success, in order for this factor to be given weight appropriate to the evidence, along with other pertinent factors.”

Cont'l Can Co. USA, Inc. v. Monsanto Co., 948 F.2d 1264, 1273 (Fed. Cir. 1991); POR 55

Objective Indicia of Non-Obviousness: Commercial Success

- Undisputed: GoPro's HERO7 Black was highly successful



Objective Indicia of Non-Obviousness: Industry Praise

- **Undisputed:** GoPro's HERO7 Black and SuperView significant industry praise



Ex. 2036; POR 51



Ex. 2038; POR 51



Ex. 2039; POR 51

Objective Indicia of Non-Obviousness: Industry Praise

- Undisputed: GoPro's HERO7 Black and SuperView significant industry praise

Superview is a clever GoPro video mode which is designed to capture a more immersive wide angle perspective. It does this by shooting at a 4:3 aspect ratio, using the full height of the camera's sensor, and then dynamically stretching the sides to produce a widescreen 16:9 aspect ratio.

Ex. 2013; POR 52-53

The GoPro camera that we were sent was the "Hero3+ Black Edition". 'NB – here's the dull technical bit' It films up to [REDACTED] and it films w [REDACTED] which allowed even a selfie [REDACTED]

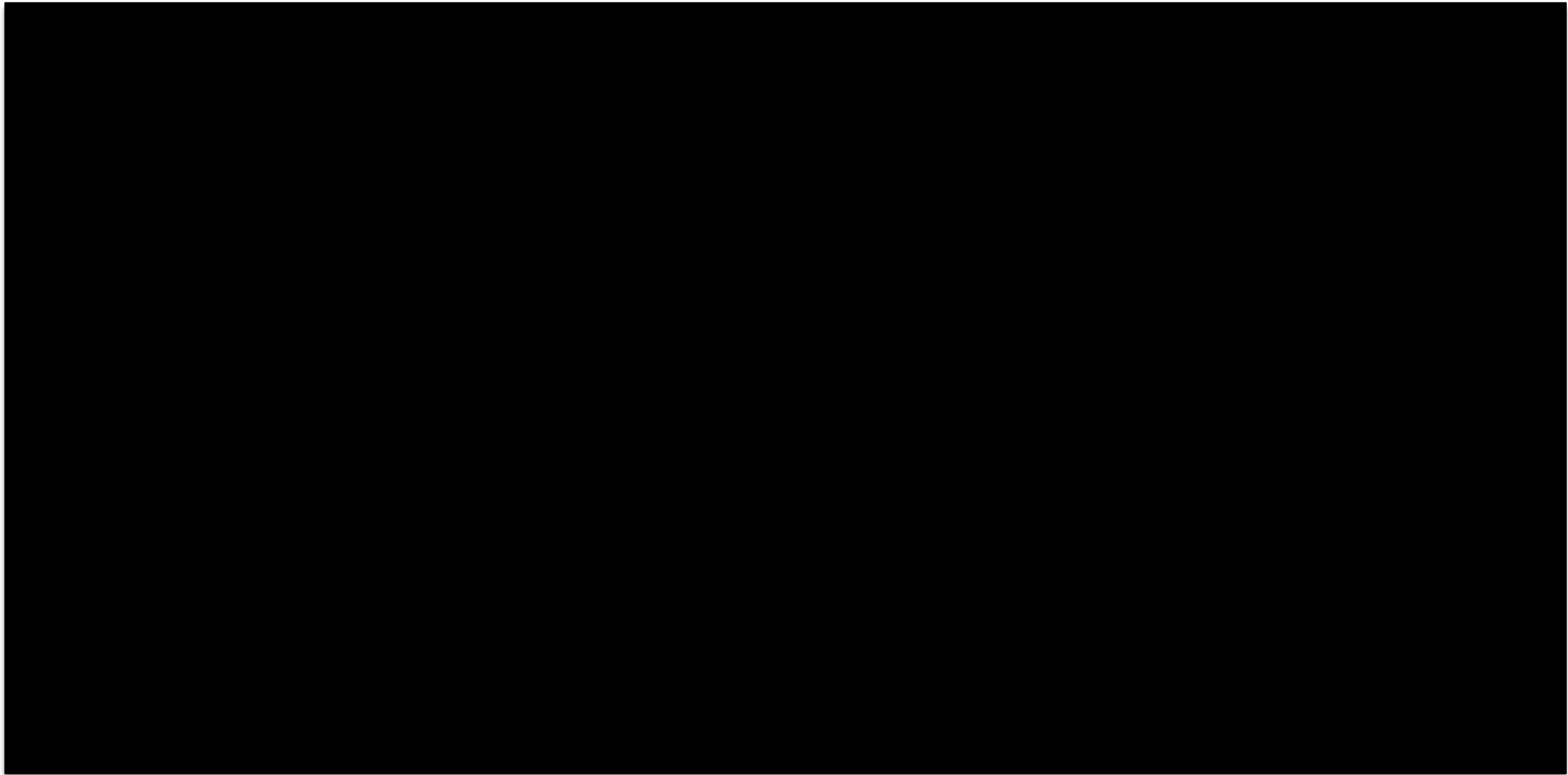
Hello, is there [REDACTED] the one in the Hero 7? Asking this because i bought the Q Action but si [REDACTED] and now i feel like i want to send the gsmo back and buy a The superview should be a software f [REDACTED] could it be possible to get it in next firmware updates? Mayt with an higher resolution than 1080p and with rocksteady on...

Ex. 2016; POR 52-53



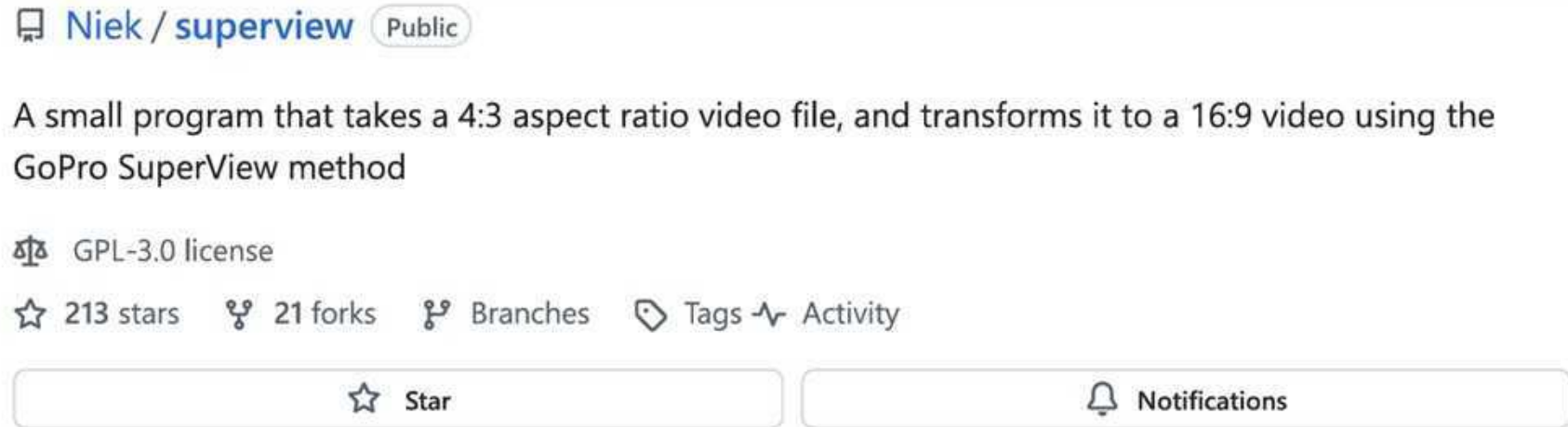
Ex. 2015; POR 53-54

Objective Indicia of Non-Obviousness: Copying



Objective Indicia of Non-Obviousness: Copying

- Others have copied SuperView



Ex. 2011; POR 52

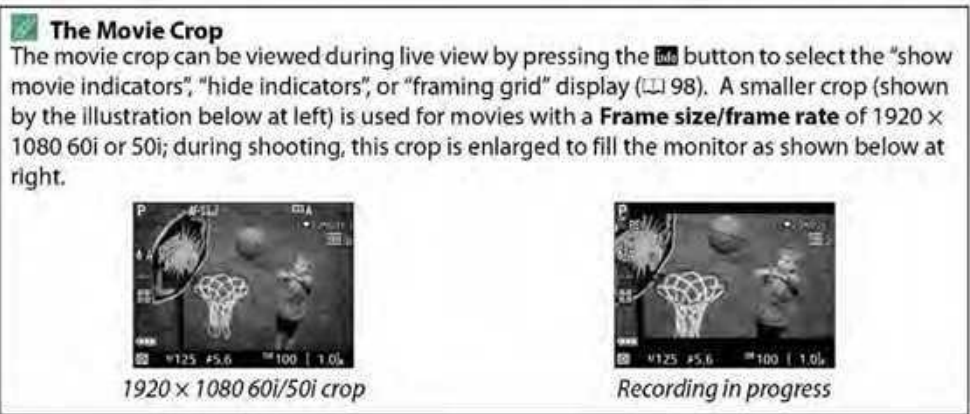
Objective Indicia of Non-Obviousness: Long-Felt Need / Unexpected Results

- Conventional wisdom was that introducing distortion was undesirable

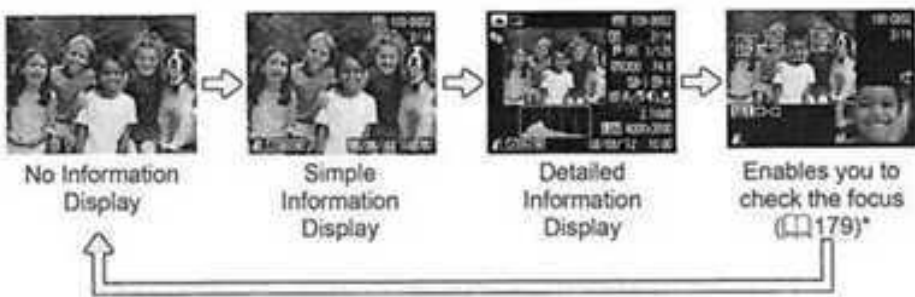
“At the time of the ’413 patent, conventional wisdom was that introducing distortion into images was undesirable. For instance, distortion, particularly geometric distortion, can stretch or compress parts of the image in an uneven way, making faces or bodies look elongated, squished, or warped. This effect is especially undesirable in personal or group photos, where people expect to see a realistic representation of themselves. Because of these reasons, photographers and camera designers traditionally seek to avoid or minimize distortion to maintain the integrity, realism, and appeal of their photos.”

Ex. 2027 (Foroosh Decl.) at ¶169

- Traditional cameras avoided introducing distortion



Ex. 2020 (Nikon D5200 Manual) at 120; POR 57



Ex. 1028 (Canon PowerShot Manual) at 176; Surreply 29

Objective Indicia of Non-Obviousness: Long-Felt Need / Unexpected Results

- SuperView had the unexpected result of improving immersiveness by introducing distortion

“In particular, GoPro’s SuperView invention utilizes a sensor that captured an image in one aspect ratio (4:3) so that the image can be “taller,” and then allowed it to be displayed and previewed on the screen in aspect ratios that are more commonly used in video settings (e.g., 16:9). To do this, however, GoPro had to intentionally introduce distortion into the image. SuperView thus applies a warping function that distorts the edges while eliminating distortion in the center. GoPro’s unconventional approach arises from the fact that GoPro’s cameras are directed at action sports, not conventional photography. GoPro recognized that video for certain action sport scenes, like skiing, surfing and bicycling, result in persons being centered in the middle of the shot, with no persons on the edges. This means that the periphery can be distorted in an acceptable manner.”

Ex. 2027 (Foroosh Decl.) at ¶171; POR 57

Ground 1: Fleming Does Not Disclose “Image Capture Apparatus” or “Camera” (All Claims)

Ground 1: Fleming Does Not Render Obvious “Camera”

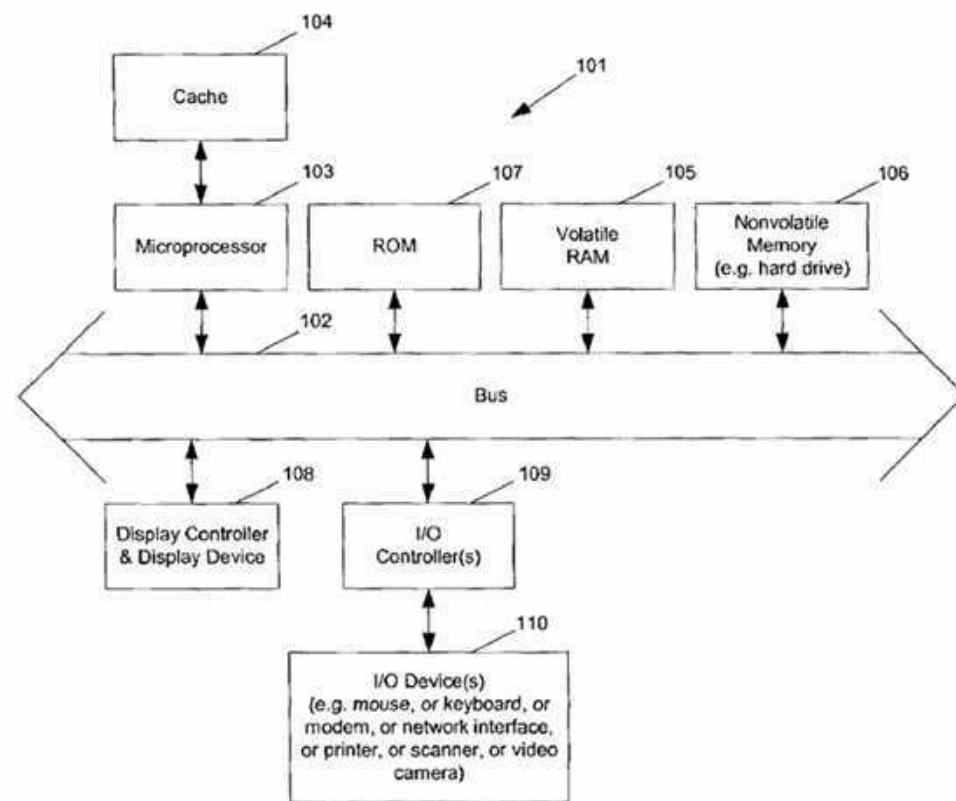
- **Claim Language:** Transformation must occur on an “image capture apparatus” (Claim 1) or “camera” (Claims 6 and 7)
- **Petitioner:**
 - Admits Fleming does not expressly disclose performing transformation on a camera
 - Argues modifying to include a camera, processors, and algorithms on the camera would be obvious
- Petitioner is **incorrect:**
 - ID: Petition lacks evidence
 - Reply relies on ***new evidence not submitted in the Petition:***
 - Canon PowerShot G15
 - Olympus E-PL5 PEN
 - Petitioner provides no motivation to use Fleming’s algorithm over conventional camera approaches
 - Petition failed to allege expectation of success

Ground 1: Fleming Does Not Render Obvious “Camera”

- Fleming:

“FIG. 1 shows one example of a typical computer system which may be used with the present invention. Note that while FIG. 1 illustrates various components of a computer system, it is not intended to represent any particular architecture or manner of interconnecting the components as such details are not germane to the present invention. It will also be appreciated that network computers and other data processing systems which have fewer components or perhaps more components may also be used with the present invention. The computer system of FIG. 1 may, for example, be an Apple Macintosh computer.”

Ex. 1004 (Fleming) at 3:8-18; Reply 11, Ex. 1002 (Katsaggelos Decl.) ¶46



Ground 1: Fleming Does Not Render Obvious “Camera”

- Petition’s motivation to combine:

“It would have been obvious for Fleming’s data processing system (including its GPU and CPU) to be included within a camera. At the time of the invention in 2013, digital cameras included processors for processing the image data captured by an image sensor. Digital cameras at this time stored files having a different aspect ratio (e.g., EX1004, 16:9-11), or included LCD displays (for viewing the captured image) that had a different aspect ratio than the image data captured by an image sensor. Since the aspect ratio of the captured image was different than the aspect ratio of the stored file or the LCD display included in the camera, it would have been obvious for the digital camera to include data processing systems to transform the captured image into an output image (having the aspect ratio of the stored file or LCD display) to be presented on the camera display. A POSITA would have been motivated to have on-camera processing and display capabilities because it would have been less burdensome, cheaper, and saves time. Having processing and display capabilities on camera are known techniques used to improve devices that are similar and/or ready for improvement.”

Pet. 28-29

Ground 1: Fleming Does Not Render Obvious “Camera”

- **ID:**

“Petitioner does not appear to rely on any evidence in Grounds 1 and 3, other than the Dr. Katseggelos’s testimony, for its assertion that in 2013, cameras captured images in one aspect ratio and stored or displayed those images in a different aspect ratio. Dr. Katsaggelos testifies that while the aspect ratios of images captured by an image sensor were 3:2 or 4:3 the aspect ratios of “the files that stored the captured images may be in a format for a different aspect ratio, such as 16:9 (e.g., for landscape orientation), or the LCD displays within the camera may have a different aspect ratio, such as 16:10 (for viewing the captured image).” Ex. 1002 ¶¶ 100, 201. Like Petitioner, Dr. Katsaggelos also does not cite to any evidence supporting this testimony.”

ID 35

“The aspect ratios of images captured by an image sensor were, e.g., 3:2 or 4:3. However, the files that stored the captured images may be in a format for a different aspect ratio, such as 16:9 (e.g., for landscape orientation), or the LCD displays within the camera may have a different aspect ratio, such as 16:10 (for viewing the captured image). Hence, the aspect ratio of the image captured by the image sensor (3:2 or 4:3, for example) may be different from the aspect ratio of the image stored in memory or displayed on the display (16:9 or 16:10, for example). Due to this difference in aspect ratios, it would have been obvious for digital cameras to include data processing systems and processors for transforming the captured image into an output image.”

Ex. 1002 (Katsaggelos Decl.) at ¶100

Ground 1: Fleming Does Not Render Obvious “Camera”

- **ID**

“Petitioner does not appear to rely on any evidence in Grounds 1 and 3, other than the Dr. Katseggelos’s testimony, for its assertion that in 2013, cameras captured images in one aspect ratio and stored or displayed those images in a different aspect ratio.”

ID 35

- **Reply:**

- Added **new** evidence (alleged Canon and Olympus documentation)
- Added **new** obviousness arguments applying Fleming to camera systems



Ex. 1026; Reply 22



Ex. 1030; Reply 22

Ground 1: Fleming Does Not Render Obvious “Camera”

- **Reply:** Canon PowerShot G15 performs aspect ratio conversion for display *Reply 22*
- **Incorrect:** Canon PowerShot G15 can only change the **shooting** aspect ratio, not display:

Changing the Aspect Ratio

Change the image aspect ratio (ratio of width to height) as follows.



Configure the setting.

- Press the <FUNC. SET> button, choose [4:3] in the menu, and choose the desired option (47).
- Once the setting is complete, the aspect ratio will be updated.
- To restore the original setting, repeat this process but choose [4:3].



Ex. 1028 (Canon Manual) at 80; Surreply 28

FUNC. Menu

Configure commonly used shooting functions through the FUNC. menu as follows.
Note that menu items and options vary depending on the shooting mode (292 – 295).



1 Access the FUNC. menu.
Press the <FUNC. SET> button.

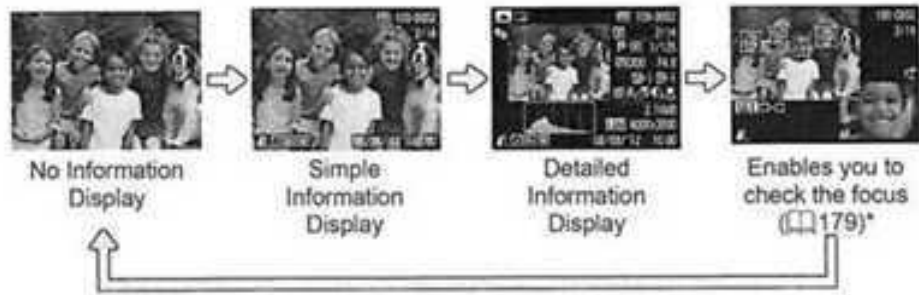
2 Choose a menu item.
Press the <▲><▼> buttons to choose a menu item.
Available options are shown at the bottom of the screen.



Ex. 1028 (Canon PowerShot Manual) at 47; Surreply 28

Ground 1: Fleming Does Not Render Obvious “Camera”

- **Reply’s New Motivation:** Cameras with different image sensor and screen aspect ratio “need” Fleming’s aspect ratio conversion *Reply 22*
- Conventional methods were preferred for digital point-and-click cameras:



Ex. 1028 (Canon PowerShot Manual) at 176; Surreply 29

“[N]on-linear transformations were expensive to design and implement, and required significant processing power to Execute. **Portable devices and cameras in 2013 had limited processing power and battery life** that a POSITA would find undesirably wasted if every photograph had to have its aspect ratio converted due to a mismatch between the image sensor and the display screen. As a result, a POSITA would have found it much more preferable for the camera and display to have the same aspect ratio in order to avoid distortion of the image and the expensive non-linear transformations claimed in the ’413 patent.”

Ex. 2027 (Foroosh Decl.) at ¶108; POR 63

- Introducing distortion for point-and-click photography was undesirable

“If one has to choose distortion or non-distortion, one probably would **choose the non-distortion** option.”

Ex. 2026 (Katsaggelos Tr.) at 87:21 -23; POR 65

“What it’s not especially useful for is any time that the highly distorted and compressed look is going to appear strange. So you **wouldn’t usually use it for standard people shots**, for example”

Ex. 2003 at 6; POR 66

Ground 1: Fleming Does Not Render Obvious “Camera”

- **Reply’s New Motivation:** motivation for performing non-uniform shifting using a separate device

“Applying non-uniform shifting by the camera’s processor(s) **reduces the need for a separate device** and associated connectors (e.g., cable), eliminates the need to transfer data between the camera and separate device, eliminates the need to wait for connecting the camera to the separate device (and associated burden) and operating the separate device. Neither PO nor its expert disputes these benefits.”

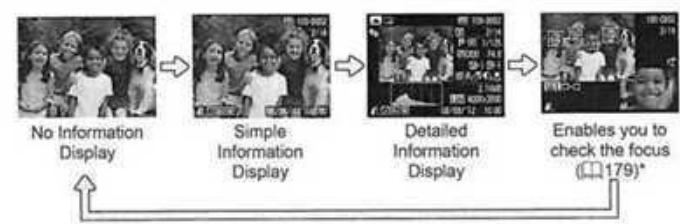
Reply 21

- Petitioner’s “separate device” motivation not directed to Petition’s combination:

“It would have been obvious for Fleming’s data processing system (including its GPU and CPU) to be included within a camera.”

Pet. 28

- Conventional cameras already presented images on a display without a separate device



Ex. 1028 (Canon PowerShot Manual) at 176; Surreply 29

Ground 1: Fleming Does Not Render Obvious “Camera”

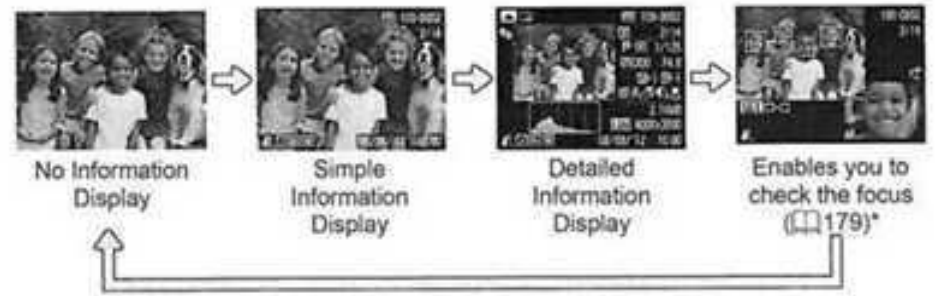
Petition’s Original Motivation: “less burdensome, cheaper, and saves time”

- **Reply’s New Motivation** abandons Petition’s original motivation:

“A POSITA had reasons (avoiding image loss, black bars, or perceivable distortion) for applying non-uniform shifting, even if it meant potentially foregoing some benefits (more required processing and longer battery life).”

Reply 23

- No evidence a POSITA would be motivated by such reasons given camera’s small 2” screen



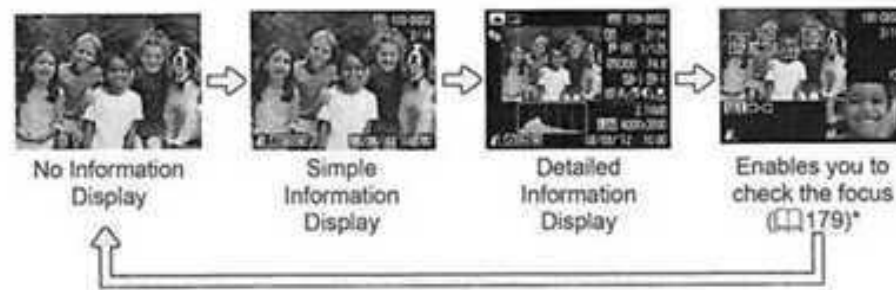
Ex. 1028 (Canon PowerShot Manual) at 176; Surreply 29

- POSITA would not be motivated to modify the image on the camera because users typically want the original image to download off the camera.

Ground 1: Fleming Does Not Render Obvious “Camera”

- Petition did not assert Reasonable Expectation of Success
 - ***Reply’s New Reasonable Expectation of Success:***
 - Fleming only uses a simple math equation
 - Fleming’s non-linear scaling could be performed using lookup tables stored in memory
- Reply 24-25*

- Petitioner provides no evidence that a point-and-click camera had sufficient processing capabilities and memory to perform such equations or look-up on every pixel
- No evidence that the cpu-heavy process would not introduce lag to the image viewer



Ex. 1028 (Canon PowerShot Manual) at 176; Surreply 29

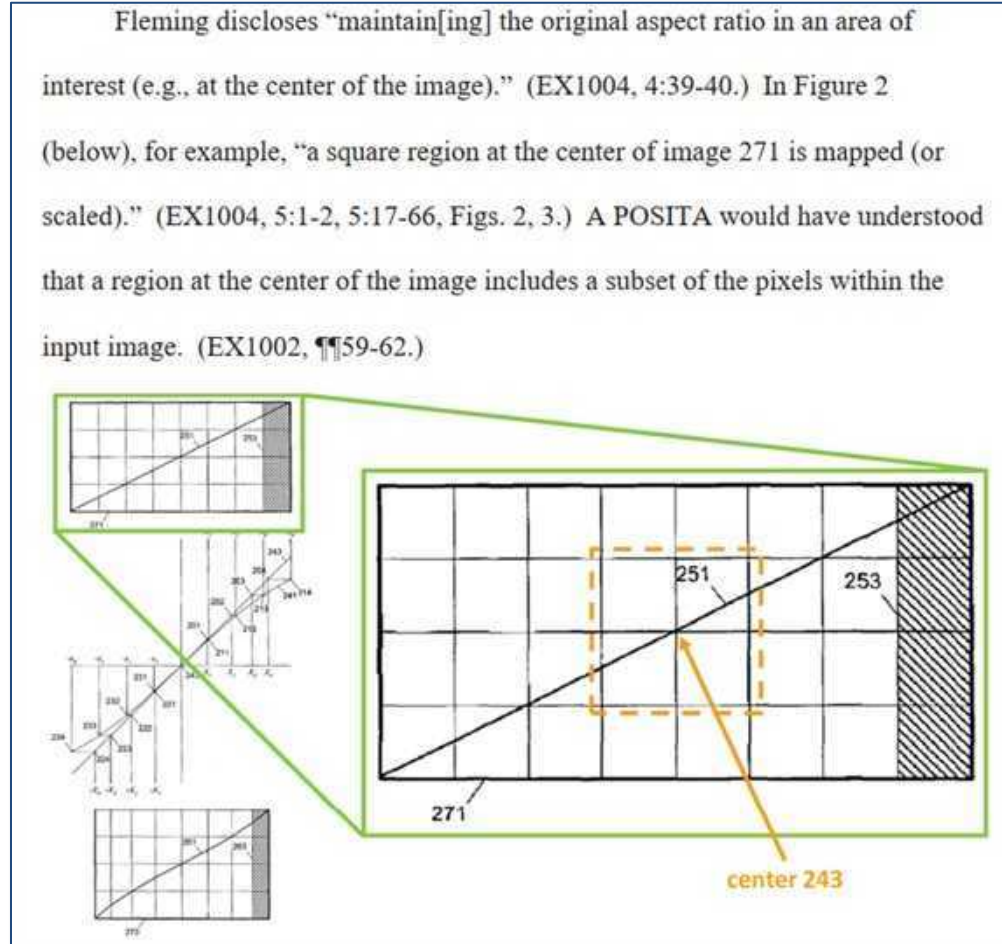
Ground 1: Fleming Does Not Disclose “Determine a Portion” (All Claims)

Ground 1: Fleming Does Not Disclose “Determine a Portion”

- Petition:
 - Presence of central region with no distortion “determines a portion”
- Reply:
 - Fleming discloses regions according to user input
 - Fleming discloses dividing frame into regions
- Petitioner is incorrect:
 - Mere presence of a non-distorted region does not disclose “determining” the portion after obtaining the image
 - Petitioner’s new Reply arguments mischaracterize Fleming, which does not disclose selecting the regions after obtaining the image, they are instead pre-selected

Ground 1: Fleming Does Not Disclose “Determine a Portion”

- Petition merely relied on the presence of a less distorted region



Pet. 88

- Fleming does not disclose **determining** the less distorted region **after** obtaining the image

Ground 1: Fleming Does Not Disclose “Determine a Portion”

- **ID**

As guidance to the parties, we note the cited art appears to support Petitioner’s contentions even accepting Patent Owner’s interpretation of the claim requiring that the determining limitation occur prior to the transformation limitation. For example, Fleming discloses that “it is desirable to maintain the original aspect ratio in an area of interest” and therefore “the image in the area of interest is presented in the target display area with reduced distortion.” Ex. 1004, 4:38–45. This indicates that in Fleming the area of interest is determined before the transformation takes place. Similarly, Obara discloses vertically dividing the image into vertical image blocks (certain of these blocks are understood to be the central area) and then horizontally reducing or enlarging the blocks. Ex. 1006 ¶¶ 52–62.

ID 33

- Fleming does not disclose **determining** the less distorted region **after** obtaining the image

Ground 1: Fleming Does Not Disclose “Determine a Portion”

- ***Reply’s New Arguments:***

“other passages in Fleming show “determine” after “obtain” and prior to “apply” as separate steps—and do not describe PO’s alleged effect of the transformation. (EX1004, 2:15-18 (regions determined according to user input), 7:5-16 (dividing frame into regions), 8:45-9:9 and Figs. 10-11 (describing receiving video stream, dividing a frame of video stream into regions, and scaling the regions in that order); POR, 70.)”

Reply 26

- Petitioner is **incorrect:**
 - None of these disclosures indicate that the regions are determined ***after*** the image is obtained
 - Each of these disclosures involves using ***pre-determined*** regions ***before*** the image is obtained
 - Figures 10-11 are piecewise linear

Ground 3: Obara-Liu Does Not Disclose “Image Capture Apparatus” or “Camera” (All Claims)

Ground 3: Obara-Liu Does Not Render Obvious “Camera”

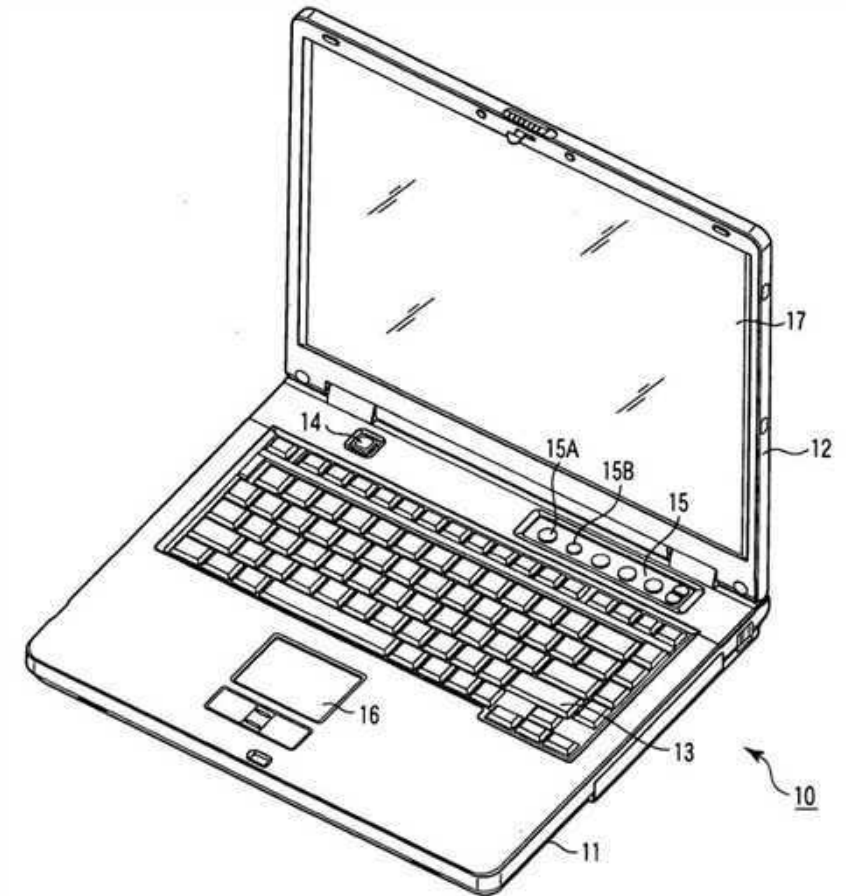
- **Claim Language:** Transformation must occur on an “image capture apparatus” (Claim 1) or “camera” (Claims 6 and 7)
- **Petitioner:**
 - Obara-Liu does not disclose performing transformation on a camera, but would have been obvious to do so
- Petitioner is **incorrect:**
 - ID: Petition lacks evidence
 - Reply relies on ***new, unauthenticated evidence not submitted in the Petition:***
 - Canon PowerShot G15
 - Olympus E-PL5 PEN
 - Petitioner provides no motivation to use Obara’s algorithm over conventional camera approaches
 - Petition did not assert reasonable expectation of success
 - Reply provides no evidence Obara’s complex algorithm would adequately run on a digital camera

Ground 1: Obara-Liu Does Not Render Obvious “Camera”

- Obara

“FIG. 1 is a perspective view that shows the state in which a display unit of the personal computer 10 is opened. The computer 10 comprises a computer main body 11 and a display unit 12. A display device that is composed of a TFT-LCD (Thin Film Transistor Liquid Crystal Display) 17 is built in the display unit 12. The display screen of the LCD 17 is positioned at an approximately central part of the display unit 12. The LCD 17 has a horizontally elongated wide display screen. The size (resolution) of the display screen is, for example, 1440×900 pixels.”

Ex. 1006 (Obara) at ¶25; Pet. 72



Ground 3: Obara-Liu Does Not Render Obvious “Camera”

- Petition’s motivation to combine:

“It would have been obvious for Obara’s image processing unit and CPU 111 to be included within a camera. At the time of the invention in 2013, Digital cameras at this time included LCD displays (for viewing the captured image), some of which had a different aspect ratio than the captured image. Since the aspect ratio of the captured image was different than the aspect ratio of the LCD display included in the camera, it would have been obvious for the digital cameras to include data processing systems to transform the captured image into an output image (having the aspect ratio of the LCD display) to be presented on the display of the camera. A POSITA would have been motivated to have on-camera processing and display capabilities because it would have been less burdensome, cheaper, and saves time. (EX1002, ¶¶200-202.) .”

Pet. 71-72

Ground 3: Obara-Liu Does Not Render Obvious “Camera”

- **ID**

“Petitioner does not appear to rely on any evidence in Grounds 1 and 3, other than the Dr. Katsaggelos’s testimony, for its assertion that in 2013, cameras captured images in one aspect ratio and stored or displayed those images in a different aspect ratio. Dr. Katsaggelos testifies that while the aspect ratios of images captured by an image sensor were 3:2 or 4:3 the aspect ratios of “the files that stored the captured images may be in a format for a different aspect ratio, such as 16:9 (e.g., for landscape orientation), or the LCD displays within the camera may have a different aspect ratio, such as 16:10 (for viewing the captured image).” Ex. 1002 ¶¶ 100, 201. Like Petitioner, Dr. Katsaggelos also does not cite to any evidence supporting this testimony.”

ID 35

“The aspect ratios of images captured by an image sensor were, e.g., 3:2 or 4:3, which may be different from the aspect ratios of files that stored the captured images. Digital cameras at this time also included LCD displays (for viewing the captured image). The aspect ratios of the images captured by an image sensor may also be different from the LCD displays within the camera. Due to differences in aspect ratios, it would have been obvious to convert from the aspect ratio of the images captured by the image sensor to the aspect ratio of the stored files and/or LCD display.”

Ex. 1002 (Katsaggelos Decl.) at ¶201

Ground 3: Obara-Liu Does Not Render Obvious “Camera”

- **ID**

“Petitioner does not appear to rely on any evidence in Grounds 1 and 3, other than the Dr. Katseggelos’s testimony, for its assertion that in 2013, cameras captured images in one aspect ratio and stored or displayed those images in a different aspect ratio.”

ID 35

- **Reply:**

- Added **new** evidence about camera systems
- Added **new** obviousness arguments applying Fleming to camera systems



Ex. 1026; Reply 22



Ex. 1030; Reply 22

Ground 3: Obara-Liu Does Not Render Obvious “Camera”

- **Reply:** Canon PowerShot G15 performs aspect ratio conversion for display *Reply 22*
- **Incorrect:** Canon PowerShot G15 can only change the **shooting** aspect ratio, not display:

Changing the Aspect Ratio

Change the image aspect ratio (ratio of width to height) as follows.



Configure the setting.

- Press the <FUNC. SET> button, choose [4:3] in the menu, and choose the desired option (47).
- Once the setting is complete, the aspect ratio will be updated.
- To restore the original setting, repeat this process but choose [4:3].

Ex. 1028 (Canon Manual) at 80; Surreply 28

FUNC. Menu

Configure commonly used shooting functions through the FUNC. menu as follows.
Note that menu items and options vary depending on the shooting mode (292 – 295).

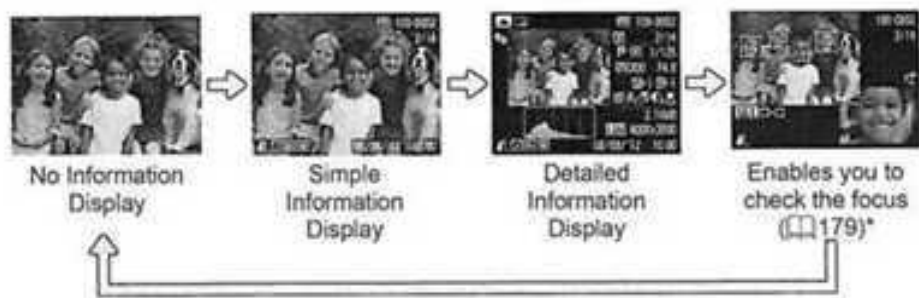


- Access the FUNC. menu.**
Press the <FUNC. SET> button.
- Choose a menu item.**
Press the <▲><▼> buttons to choose a menu item.
Available options are shown at the bottom of the screen.

Ex. 1028 (Canon PowerShot Manual) at 47; Surreply 28

Ground 3: Obara-Liu Does Not Render Obvious “Camera”

- **Reply’s New Motivation:** Cameras with different image sensor and screen aspect ratio “need” Obara’s aspect ratio conversion *Reply 26-27*
- Conventional methods were preferred for digital point-and-click cameras:



Ex. 1028 (Canon PowerShot Manual) at 176; Surreply 29

“[N]on-linear transformations were expensive to design and implement, and required significant processing power to Execute. Portable devices and cameras in 2013 had limited processing power and battery life that a POSITA would find undesirably wasted if every photograph had to have its aspect ratio converted due to a mismatch between the image sensor and the display screen. As a result, a POSITA would have found it much more preferable for the camera and display to have the same aspect ratio in order to avoid distortion of the image and the expensive non-linear transformations claimed in the ’413 patent.”

Ex. 2027 (Foroosh Decl.) at ¶108; POR 63

- Introducing distortion for point-and-click photography was undesirable

“If one has to choose distortion or non-distortion, one probably would choose the non-distortion option.”

Ex. 2026 (Katsaggelos Tr.) at 87:21 -23; POR 65

“What it’s not especially useful for is any time that the highly distorted and compressed look is going to appear strange. So you wouldn’t usually use it for standard people shots, for example”

Ex. 2003 at 6; POR 66

Ground 3: Obara-Liu Does Not Render Obvious “Camera”

- **Reply’s New Motivation:** motivation for performing non-uniform shifting using a separate device

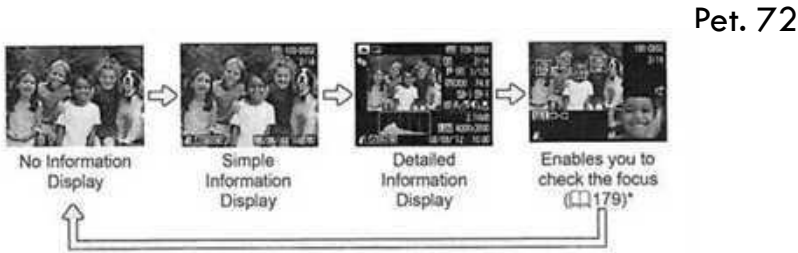
“Applying non-uniform shifting by the camera’s processor(s) **reduces the need for a separate device** and associated connectors (e.g., cable), eliminates the need to transfer data between the camera and separate device, eliminates the need to wait for connecting the camera to the separate device (and associated burden) and operating the separate device. Neither PO nor its expert disputes these benefits.”

Reply 21

- Petitioner’s “separate device” motivation not directed to Petition’s combination:

“It would have been obvious for Obara’s image processing unit and CPU 111 to be included within a camera.”

- Conventional cameras already presented images on a display without a separate device



Ex. 1028 (Canon PowerShot Manual) at 176; Surreply 29

Ground 3: Obara-Liu Does Not Render Obvious “Camera”

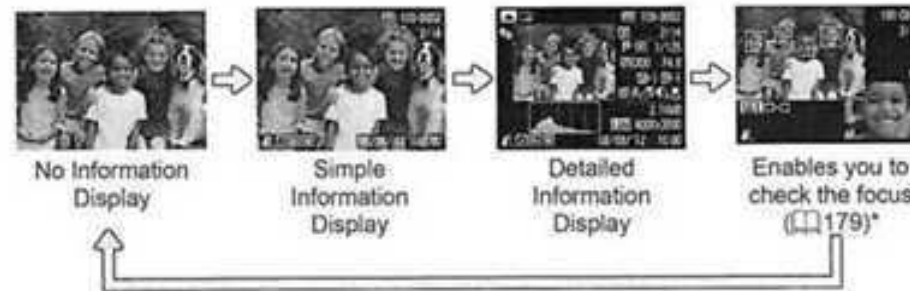
Petition’s Original Motivation: “less burdensome, cheaper, and saves time”

- **Reply’s New Motivation** abandons Petition’s original motivation:

“A POSITA had reasons (avoiding image loss, black bars, or perceivable distortion) for applying non-uniform shifting, even if it meant potentially foregoing some benefits (more required processing and longer battery life).”

Reply 23

- No evidence a POSITA would be motivated by such reasons given camera’s small 2” screen

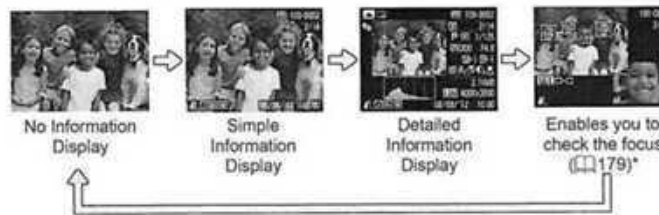


Ex. 1028 (Canon PowerShot Manual) at 176; Surreply 29

- POSITA would not be motivated to modify the image on the camera because users typically want the original image to download off the camera.

Ground 3: Obara-Liu Does Not Render Obvious “Camera”

- Petition did not assert Reasonable Expectation of Success
- ***Reply’s New Reasonable Expectation of Success:***
 - Obara only uses a simple math equation
 - Obara’s non-linear scaling could be performed using lookup tables stored in memory *Reply* 24-25
- Petitioner provides no evidence that a point-and-click camera had sufficient processing capabilities and memory to perform such equations or look-up on every pixel
- Process would have to be sufficiently quick for a user to browse the image viewer



Ex. 1028 (Canon PowerShot Manual) at 176; Surreply 29

Ground 1: Obara Does Not Disclose “Determine a Portion” (All Claims)

Ground 3: Obara Does Not Disclose “Determine a Portion”

- Petition:
 - Presence of central area with no distortion “determines a portion”
- Reply:
 - Obara discloses dividing frame into frame regions
- Petitioner is incorrect:
 - Mere presence of a non-distorted region does not disclose “determining” the portion after obtaining the image
 - Petitioner’s new Reply argument uses Fleming’s Figure 10, which is piecewise linear scaling

Ground 3: Obara Does Not Disclose “Determine a Portion”

- Petition merely relied on the presence of a less distorted area

“[N]on-linear scaling module 215 vertically divides the *original image data (A)* into, for example, eight image blocks. The non-linear scaling module 215 horizontally reduces image *blocks 3, 4, 5 and 6 which belong to the central area*” (Figure 6 below). (EX1006, [0062].) A POSITA would have understood that blocks 3-6 at the central area include a subset of the pixels within original image data A (input image). (EX1002, ¶¶163-165.)

Pet. 56

- Fleming does not disclose **determining** the less distorted area **after** obtaining the image

Ground 3: Obara Does Not Disclose “Determine a Portion”

- ID

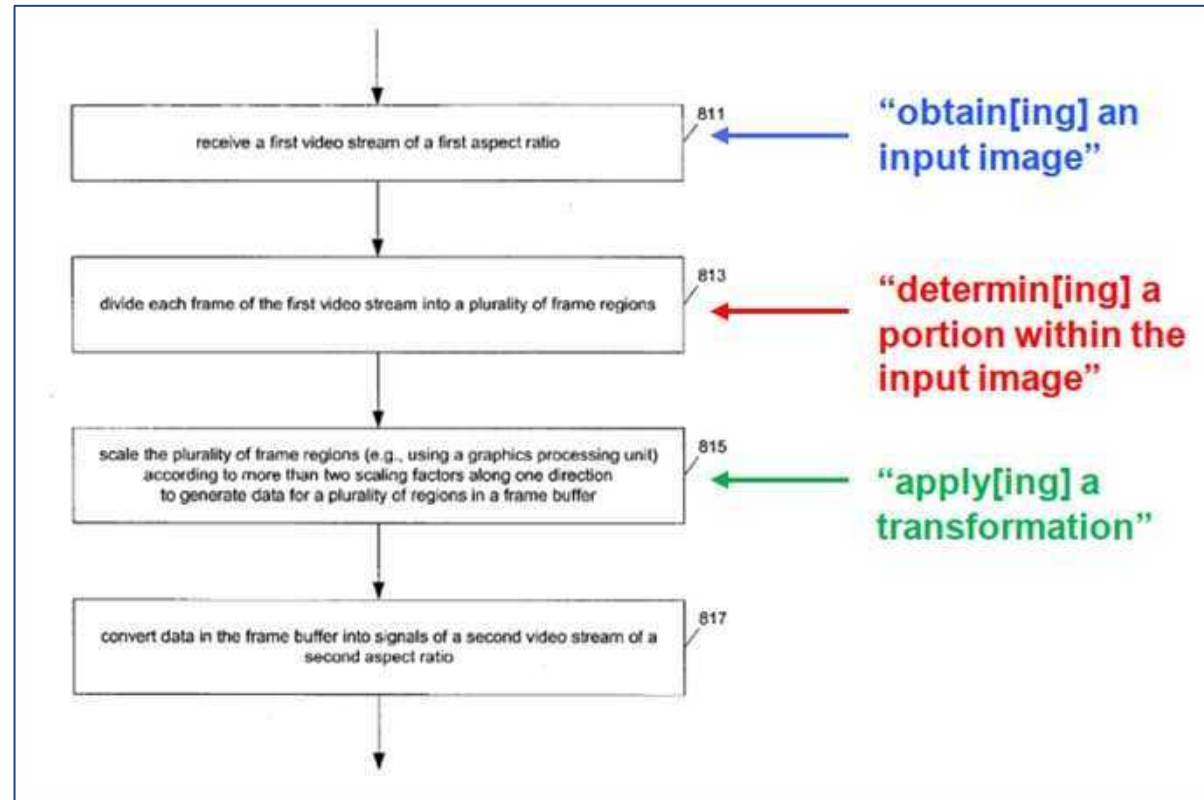
As guidance to the parties, we note the cited art appears to support Petitioner’s contentions even accepting Patent Owner’s interpretation of the claim requiring that the determining limitation occur prior to the transformation limitation. For example, Fleming discloses that “it is desirable to maintain the original aspect ratio in an area of interest” and therefore “the image in the area of interest is presented in the target display area with reduced distortion.” Ex. 1004, 4:38–45. This indicates that in Fleming the area of interest is determined before the transformation takes place. Similarly, Obara discloses vertically dividing the image into vertical image blocks (certain of these blocks are understood to be the central area) and then horizontally reducing or enlarging the blocks. Ex. 1006 ¶¶ 52–62.

ID 33

- Fleming does not disclose **determining** the less distorted area **after** obtaining the image

Ground 3: Obara Does Not Disclose “Determine a Portion”

- *Reply’s New Arguments:*



Reply 27

- Petitioner is **incorrect:**
 - This Figure is from Fleming and uses piecewise linear transformation

Patent Owner's Motion to Strike

Patent Owner's Motion to Strike

- Petitioner's Reply introduced new arguments and new evidence:
 - **Palm Treo** modified in view of Meulen
 - Meulen discloses **sub-sampling**
 - **Canon and Olympus systems** modified in view of Fleming
- The new evidence brings in teachings that are not taught in the Petition prior art:
 - Meulen does not teach a Palm Treo where image sensor and display have different aspect ratios
 - Meulen does not teach sub-sampling
 - Fleming does not teach camera systems where image sensor and display have different aspect ratios

Patent Owner's Motion to Strike: Palm Treo

- Petition **never** mentions a Palm Treo
- Petition does not assert Meulen's device captures "media source content"

d. 2[c]: "determine a portion within the input image, the portion including a subset of the pixels; and"

Meulen's media source content is an "input image." (See IV.D.3.c.)

Pet. 88

b. 1[a]: "an image sensor configured to capture an input image having a source aspect ratio, the input image including pixels located at input positions defined along a first axis and a second axis, the first axis perpendicular to the second axis;"

Element 1[a] recites limitations similar to element 2[b], except it requires "an image sensor [be] configured to capture" the input image. Digital cameras comprise an image sensor configured to capture an input image. (EX1002, ¶¶311-312.)

Pet. 104

Patent Owner's Motion to Strike: Palm Treo

- Reply introduces completely new theories relying on Palm Treo in view of Meulen

Also, Meulen's device is a Palm® Treo™. (EX1007, 2:12.) Palm Treos at that time displayed images captured by its built-in camera. (*See, e.g.*, EX1018, 63 (“[y]our Treo comes with an easy-to-use built-in camera, which allows you to instantly take, view, and share pictures”); EX1014, ¶32.)

With no viable arguments on the merits, PO resorts to mischaracterizing Meulen. It argues that there is no need for Meulen to perform aspect ratio conversion. (POR, 16.) Again, that is not what Meulen discloses. (EX1007, 4:13-19 (“the media source content may comprise an aspect ratio and/or a resolution that differs from the aspect ratio and/or the resolution of the display ... for example, the media source content may have an aspect ratio of 3:2 ..., and the display may have an aspect ratio of 1:1”).) Palm Treos' aspect ratios also do not match. (EX1022

Reply 6-7

Patent Owner's Motion to Strike: Sub-sampling

- Petition **never** mentions sub-sampling
- Petition even recognizes Meulen lacks disclosure of pixel shifting

Meulen mentions scaling, but does not disclose the details of how to perform scaling. To the extent that Meulen does not teach shifting pixel positions, it would have been obvious. Hence, a POSITA would have been motivated to turn to a prior art reference that discloses the details of scaling (known technique).

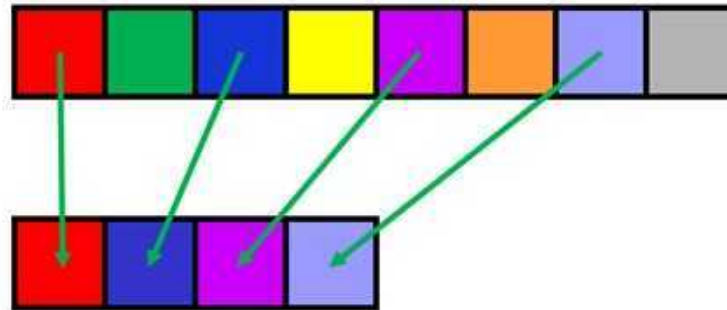
(EX1002, ¶262.)

Pet. 91

Patent Owner's Motion to Strike: Sub-sampling

- Reply introduces completely new theory that Meulen discloses sub-sampling

PO argues that Meulen's compression is performed only using polyphase filters which does not involve shifting. (POR, 11, 30.) That is incorrect. Polyphase filtering is *one* example type of sample-rate conversion that Meulen's "embodiments are not limited in this context" to. (EX1007, 5:23-29 ("perform sample-rate conversion, *such as* polyphase filtering").) Sub-sampling is a type of sample-rate conversion that shifts pixels by sampling less than all the pixels. For example, sub-sampling with a 2:1 compression ratio samples every other pixel, thereby shifting the center position of the pixels (green arrows below). (EX1014, ¶56.)



Reply 13

Patent Owner's Motion to Strike: Camera systems

- Petition **never** mentions specific camera systems
- Petition's bare assertion of modifying a camera in view of Fleming/Obara without any specific details:

“It would have been obvious for Fleming’s data processing system (including its GPU and CPU) to be included within a camera. At the time of the invention in 2013, digital cameras included processors for processing the image data captured by an image sensor. Digital cameras at this time stored files having a different aspect ratio (e.g., EX1004, 16:9-11), or included LCD displays (for viewing the captured image) that had a different aspect ratio than the image data captured by an image sensor. Since the aspect ratio of the captured image was different than the aspect ratio of the stored file or the LCD display included in the camera, it would have been obvious for the digital camera to include data processing systems to transform the captured image into an output image (having the aspect ratio of the stored file or LCD display) to be presented on the camera display. A POSITA would have been motivated to have on-camera processing and display capabilities because it would have been less burdensome, cheaper, and saves time. Having processing and display capabilities on camera are known techniques used to improve devices that are similar and/or ready for improvement.”

Pet. 28-29

- Board recognized Petition’s failure to provide evidence, but that was not an invitation to bring in new evidence and theories

Patent Owner's Motion to Strike: Camera systems

- Reply introduces new obviousness theories using different camera systems

“First, PO’s argument that camera designers in 2013 did not contemplate performing aspect ratio conversion by the camera’s processor(s) is wrong. (POR, 62.) The record evidence shows that there were digital cameras at the time that captured images with one aspect ratio and displayed or stored the captured images with another aspect ratio. This difference in aspect ratios resulted in a need for aspect ratio conversion and a reason to apply Fleming’s method by the camera’s processor(s). The Canon PowerShot G15 was one example camera (of the many) available at the time of the ’413. (EX1027; see also EX1030 (Olympus E-PL5 PEN camera available in 2012, 4:3 and 16:9 image sensor and display aspect ratios).)”

“And PO’s argument that avoiding aspect ratio conversion entirely is less burdensome, cheaper, saves more time, and is adequate does not rebut Petitioner’s motivation. (POR, 63-65.) PO ignores the applications where aspect ratio conversion is needed (e.g., with cameras available at the time such as Canon PowerShot G15 and Olympus PEN E-PL5).”

Reply 21-23

Patent Owner's Motion to Strike

Patent Owner's Motion to Strike

- Patent Owner's Surreply arguments are responsive and consistent with POR
 - '413 embodiments perform non-linear shifting on ***both axes***
 - Meulen only discloses ***uniform scaling*** on the vertical axis
 - PO properly responded to Petitioner's make a "***separate device argument***"
 - ***Presumption of nexus*** argument consistent with POR

Patent Owner's Motion to Strike

“non-linear shifting” on both axes: Surreply is responsive and consistent with POR

- **POR**

“By contrast, the ’413 patent provides an example in Figure 2A where **non-linear warping occurs on both axes**. Ex. 1001, 4:9-13 (“[A] non-linear warping is applied which results in the circles being stretched/compressed by different amounts depending on their horizontal and vertical positions resulting in example image 216.”).”

POR 40

- **Reply**

“This interpretation is consistent with the ’413 embodiments that **shift along the horizontal axis only**. (EX1001, 5:49-51 (“the horizontal warp is [a] function of the horizontal coordinate only and is independent of [the] vertical position”), 6:65-66 (“the horizontal warp distortion ... is a function of the horizontal position only”))”

Reply 19

- **Surreply**

“But **shifting still occurs along both axes** and is performed in succession rather than at the same time. EX1001, 5:29-32. Moreover, even if the embodiment only performed horizontal shifting (**it does not**), it is “a function of the horizontal position only,” and not embodied by the limitation. EX1001, 6:65-66.”

Surreply 23-24

Patent Owner's Motion to Strike

Meulen's vertical scaling: Surreply is responsive and consistent with POR

- **POR**

“Yet Meulen does not disclose that the compression in the vertical direction is performed non-uniformly. Ex. 2027, ¶99. A POSITA recognizes that this cited passage from Meulen merely describes the possibility of applying the previously disclosed piecewise linear horizontal compression followed by a **uniform vertical scaling**. Id.”

POR 40

- **Surreply**

“Petitioner assumes that “image compression and/or scaling” only refers to non-uniform compression/scaling. But Meulen describes that “image compression and/or scaling may be applied to the media source ... in one ... of the display regions,” i.e. a **uniform compression**. EX1007, 5:67-6:3; see also 8:15-18.”

Surreply 24

- **Reply**

“Meulen teaches non-uniform shifting in both directions. (Petition, 90-91.) Its phrase “image compression and/or scaling” refers to its non-uniform shifting method. **Meulen uses this same phrase in connection with describing its method** being “applied in a horizontal direction and/or a vertical direction.” (EX1007, 10:19-26; also claim 8 (narrowing claim 1’s compression to “at least one of a horizontal direction [and] a vertical direction”))”

Reply 19

Patent Owner's Motion to Strike

“Separate device”: Surreply is responsive to Reply’s “separate device” argument

- Reply

“Applying non-uniform shifting by the camera’s processor(s) reduces the need for a separate device and associated connectors (e.g., cable), eliminates the need to transfer data between the camera and separate device, eliminates the need to wait for connecting the camera to the separate device (and associated burden) and operating the separate device. Neither PO nor its expert disputes these benefits.”

Reply 21

- Surreply

“Petitioner’s motivation is really about presenting a camera’s captured image processed according to the ’413 Patent on the camera’s display without a separate device. It is a motivation to improve a nonsensical hypothetical of performing non-linear aspect ratio conversion in a separate device, and then re-uploading the image back to the camera for display. The hindsight in this theory is evident.”

Surreply 27

Patent Owner's Motion to Strike

Presumption of Nexus: Surreply is responsive and consistent with POR

- **POR**

“For over a decade, these products have included the SuperView functionality, which embodies the ’413 patent. Ex. 2006, ¶20; Ex. 2008. SuperView allows users to capture an immersive, wide-angle perspective through dynamic and non-uniform stretching of the video frame aspect ratio.”

POR 41-42

“As explained by Patent Owner’s Senior Vice President of Product Pablo Lema, there is a direct nexus between SuperView and the objective evidence.”

POR 49

- **Reply**

“GoPro does not argue presumption of nexus. It makes no attempt to show that the alleged practicing product HERO7—which includes many features unrelated to the ’413—is “coextensive with” the challenged claims.”

Reply 29

- **Surreply**

“But, Dr. Foorsh established that GoPro’s “SuperView” algorithm directly practiced and was coextensive with the ’413 Patent, EX2027, ¶¶130-154, establishing a presumptive nexus for SuperView.”

Surreply 24

Appendix: '413 Patent (Ex. 1001)

Appendix: '413 Patent (Ex. 1001)

- Ex. 1001, Fig. 1

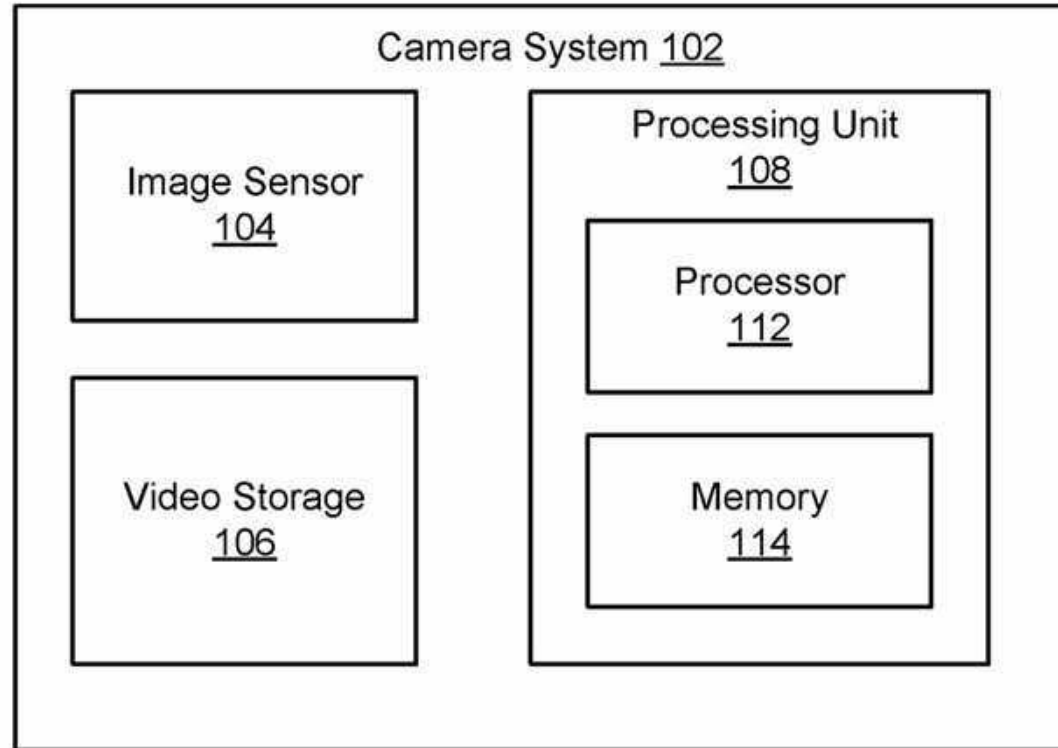
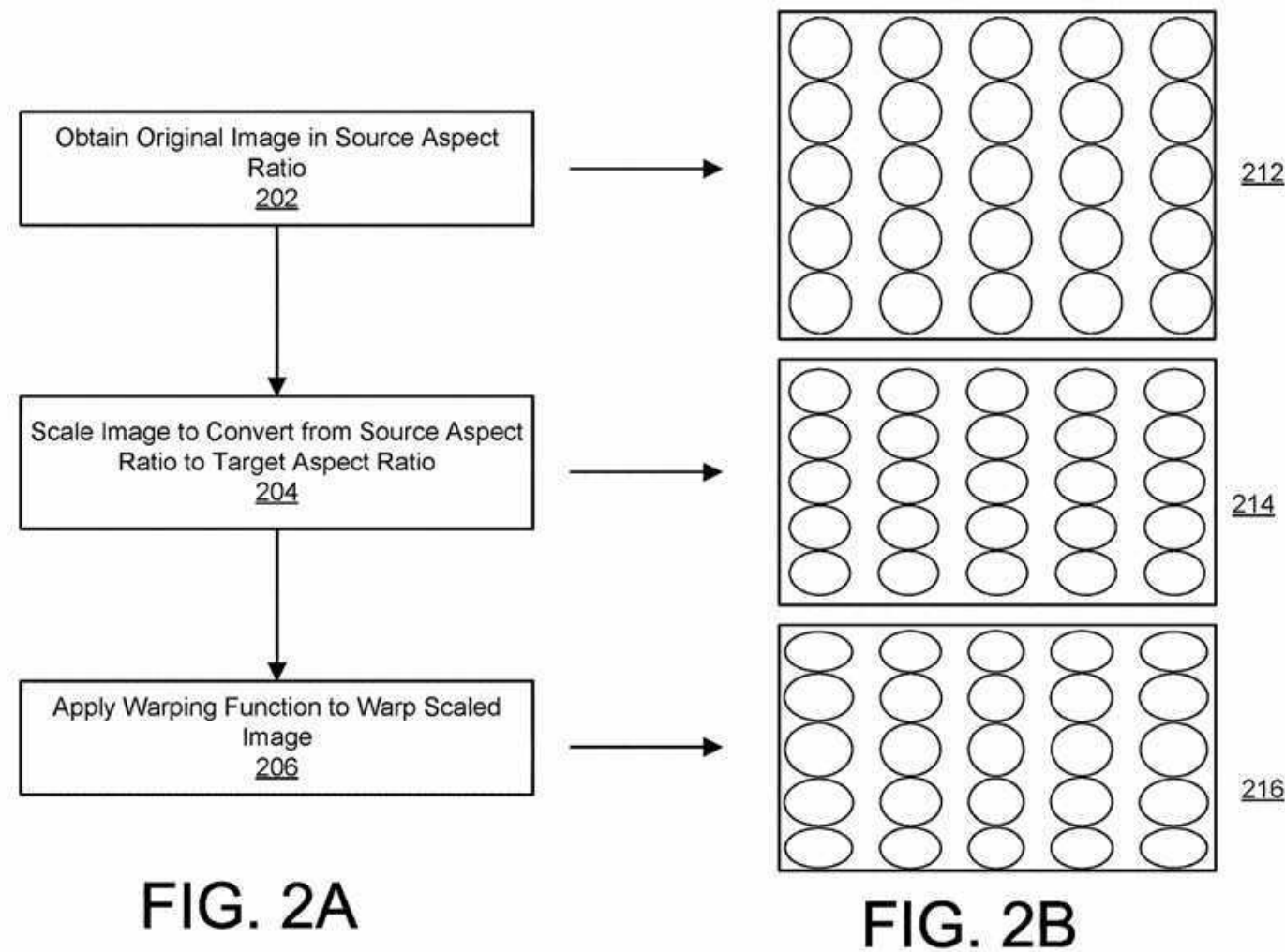


FIG. 1

Appendix: '413 Patent (Ex. 1001)

- Ex. 1001, Figs. 2A-2B



Appendix: '413 Patent (Ex. 1001)

- Ex. 1001, Fig. 3

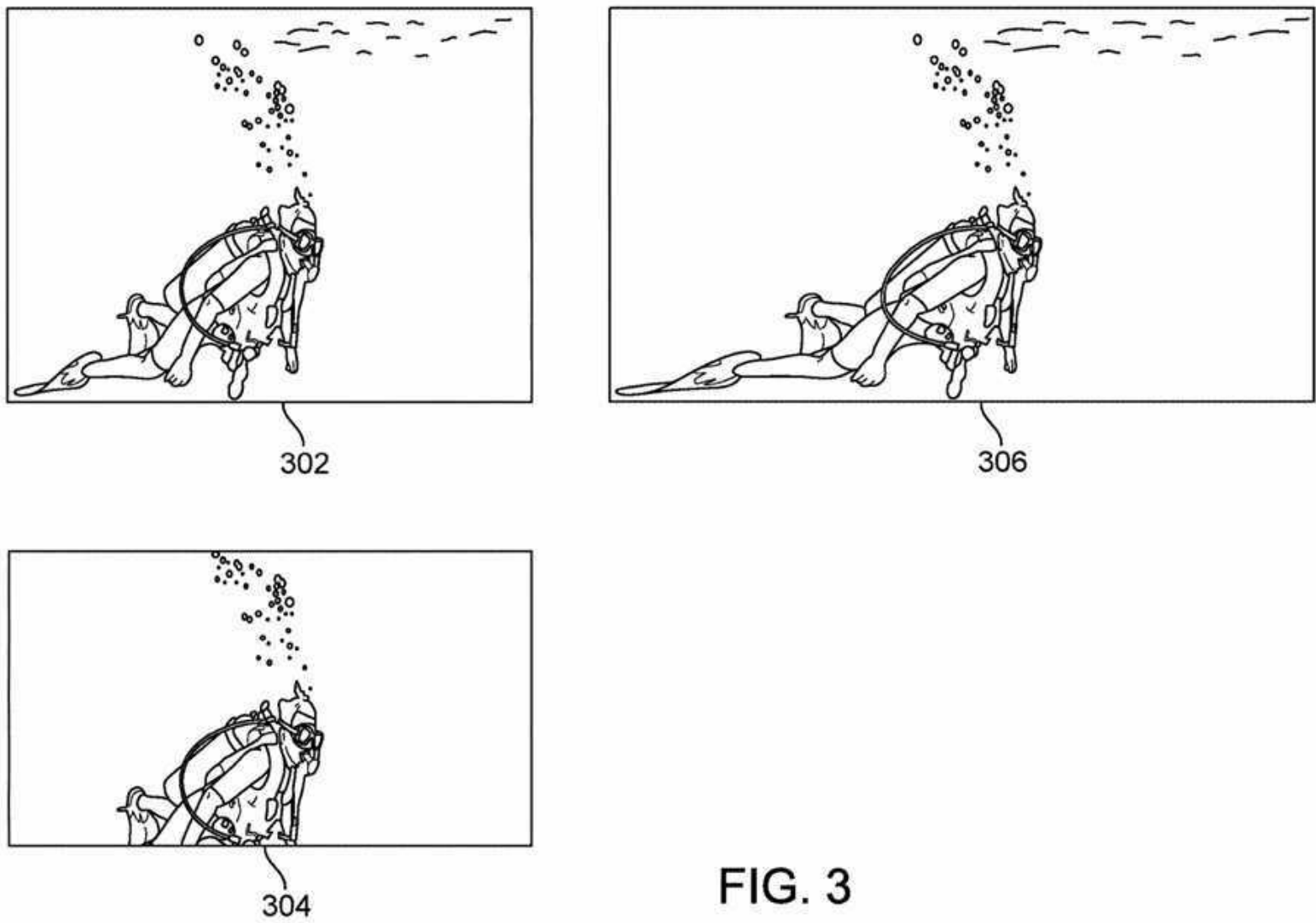
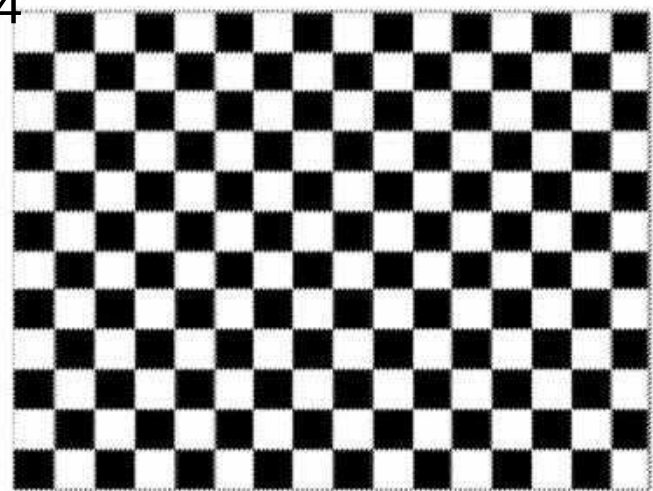


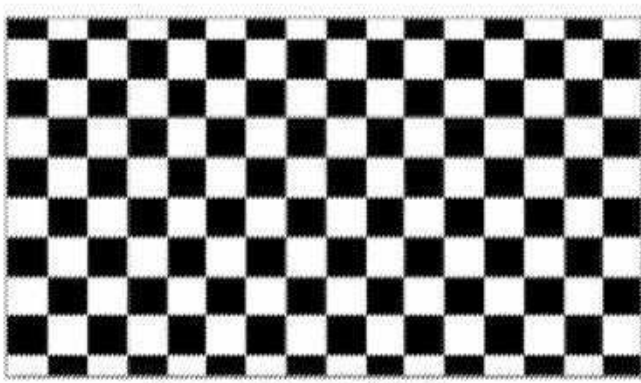
FIG. 3

Appendix: '413 Patent (Ex. 1001)

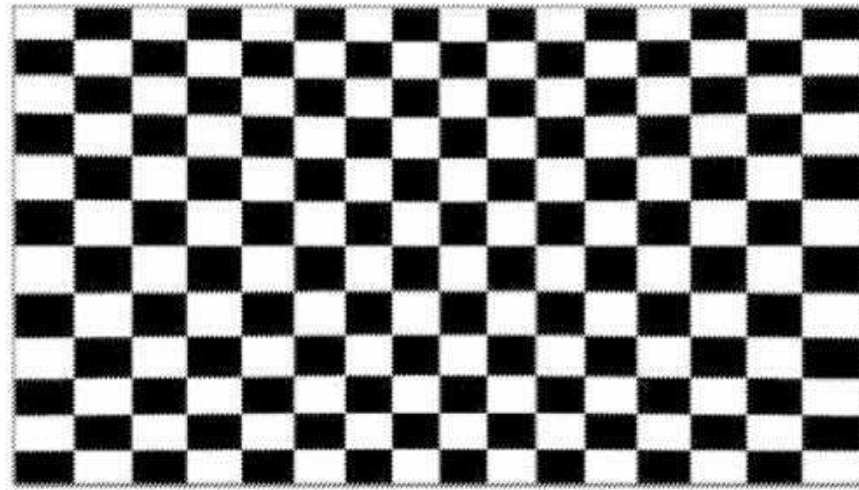
- Ex. 1001, Fig. 4



Original Image (4:3)
402



Cropped Image (16:9)
404



Non-Linear Warped
Image (16:9)
406

FIG. 4

Appendix: '413 Patent (Ex. 1001)

- Ex. 1001, Fig. 5

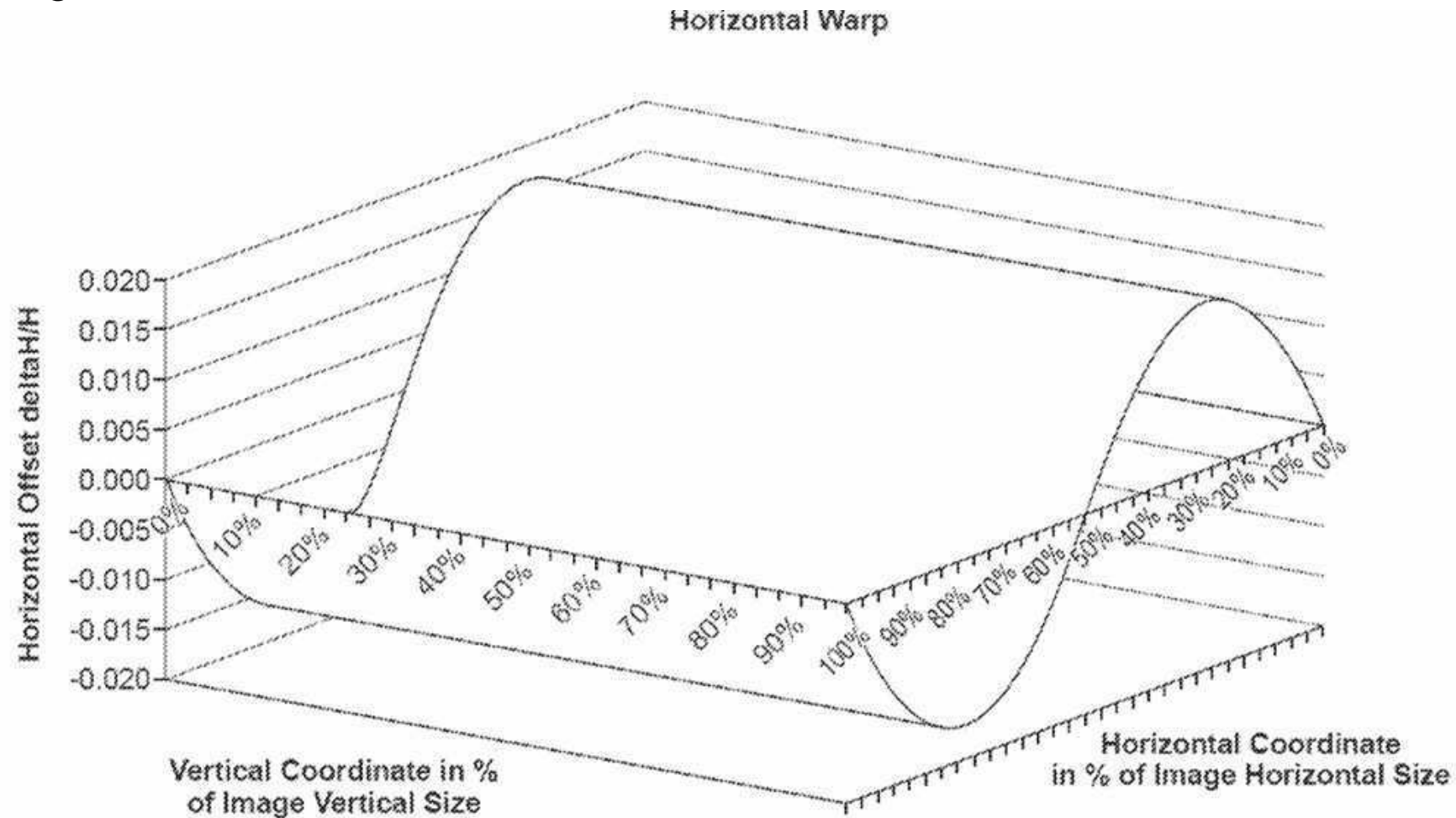


FIG. 5

Appendix: '413 Patent (Ex. 1001)

- Ex. 1001, Fig. 6

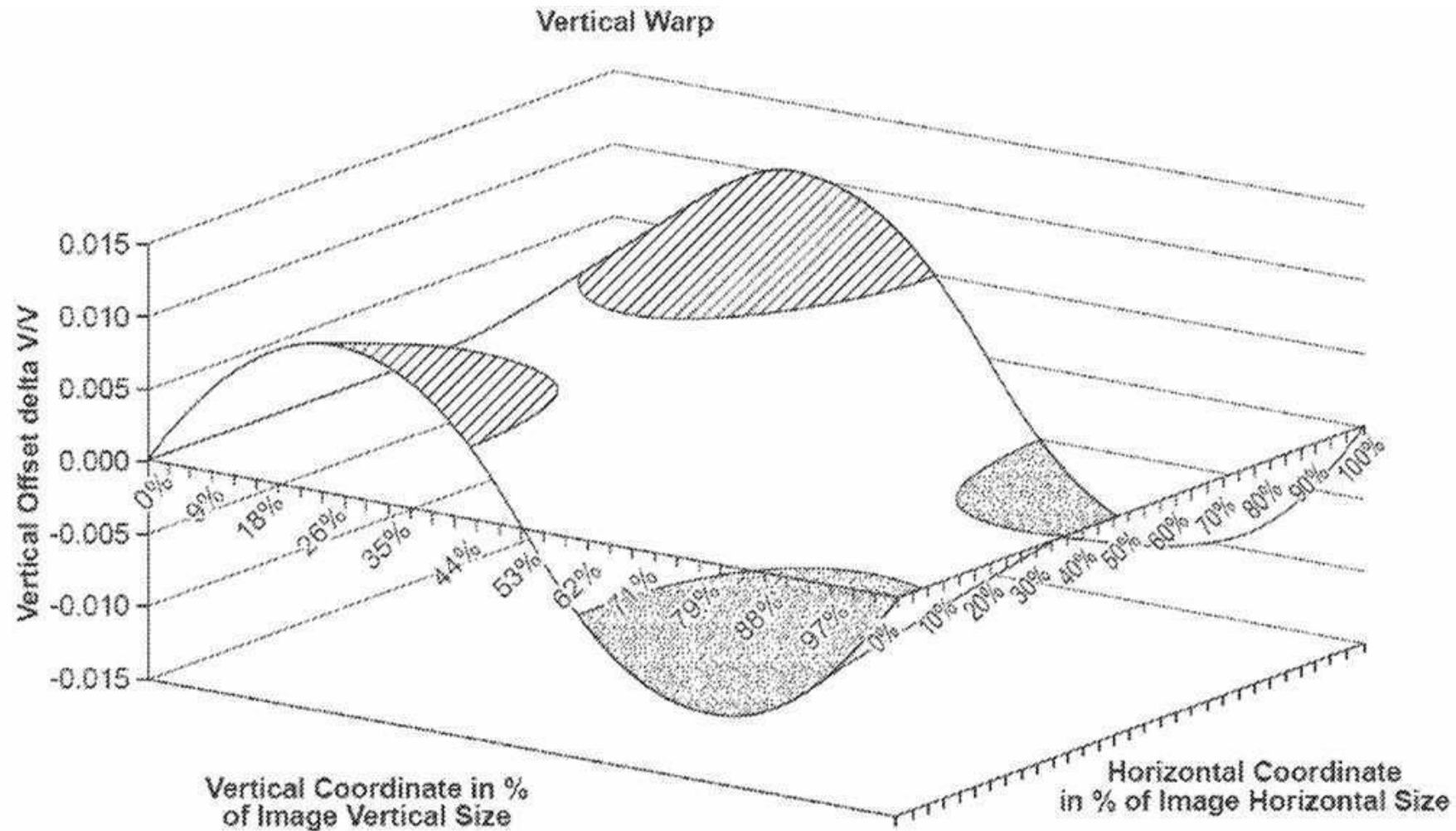


FIG. 6

Appendix: '413 Patent (Ex. 1001)

- Ex. 1001, Cols. 1-2

1
CONVERSION BETWEEN ASPECT RATIOS
IN CAMERA

BACKGROUND

Converting between aspect ratios is a common problem in the field of image and video processing. For example, a display that displays images according to a 16:9 aspect ratio may receive images natively captured according to a 4:3 aspect ratio and perform an aspect ratio conversion to fit the image to the display. Conventional conversion techniques such as cropping, linearly scaling, and padding each result in a perceivable reduction in the quality of the image or video. For example, cropping removes content from the image and reduces the field of view of the image. Linear scaling maintains the full field of view but introduces perceivable distortion into the image. For example, if an image is vertically compressed and/or horizontally stretched to convert from a 4:3 to a 16:9 aspect ratio, features in scene will appear "short and fat" relative to the original image. Padding maintains the full field of view without distortion, but results in unused display space around the sides and/or top of the image, thus rendering an image that appears too small relative to the display.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram illustrating an embodiment of a camera system.

FIG. 2A is a flowchart illustrating an embodiment of a process for converting an image from a source aspect ratio to a target aspect ratio.

FIG. 2B is set of example images being converted from a source aspect ratio to a target aspect ratio.

FIG. 3 is a first set of example images showing an original image in a source aspect ratio, a cropped image converted to a target aspect ratio using a cropping technique, and a non-linear warped image converted to the target aspect ratio using a non-linear warping technique.

FIG. 4 is a second set of example images showing an original image in a source aspect ratio, a cropped image converted to a target aspect ratio using a cropping technique, and a non-linear warped image converted to the target aspect ratio using a non-linear warping technique.

FIG. 5 is a graph illustrating an embodiment of a function for applying a horizontal warp to an image.

FIG. 6 is a graph illustrating an embodiment of a function for applying a vertical warp to an image.

DETAILED DESCRIPTION

The Figures (FIGS.) and the following description relate to preferred embodiments by way of illustration only. It should be noted that from the following discussion, alternative embodiments of the structures and methods disclosed herein will be readily recognized as viable alternatives that may be employed without departing from the principles of what is claimed.

Reference will now be made in detail to several embodiments, examples of which are illustrated in the accompanying figures. It is noted that wherever practicable similar or like reference numbers may be used in the figures and may indicate similar or like functionality. The figures depict embodiments of the disclosed system (or method) for purposes of illustration only. One skilled in the art will readily recognize from the following description that alternative

2

embodiments of the structures and methods illustrated herein may be employed without departing from the principles described herein.

Overview

A camera system comprises an image sensor, a processor, and a non-transitory computer-readable storage medium that stores instructions executable by the processor. The image sensor captures an input image having a source aspect ratio. The processor applies a transformation to the input image to scale and warp the image to generate an output image having a target aspect ratio different than the source aspect ratio. The output image has the same field of view as the input image, maintains image resolution, and limits distortion to levels that do not substantially affect the viewing experience. In one embodiment, the output image is non-linearly warped relative to the input image such that a distortion in the output image relative to the input image is greater in a corner region of the output image than a center region of the output image. The processor outputs the output image having the target aspect ratio.

In one embodiment, the aspect ratio conversion is performed directly on the camera itself rather than in an external post-processing device, such as the display device. This enables the camera to output images directly to a display in the desired target aspect ratio without requiring conversion by the display device. For example, a camera that natively captures images according to a 4:3 aspect ratio can perform conversion and output the images according to a 16:9 aspect ratio that may be displayed directly on a 16:9 display device. Furthermore, in one embodiment, the conversion may occur in substantially real-time such that images captured in the source aspect ratio can be outputted in the target aspect ratio with substantially no delay.

Beneficially, since the aspect ratio conversion preserves the entire field of view of the image, the converted image can be reverted back to its original aspect ratio without any loss of field of view or resolution. Furthermore, since the aspect ratio discrepancy is handled by the camera system, the end user needs not to make any additional adjustments apart from selecting the desired output video format on the camera. The embodiments described herein may apply to both individual images and video (which comprises a sequence of images (or "frames")).

Example Camera System

FIG. 1 illustrates an embodiment of an example camera system 102. The camera system 102 comprises an image sensor 104, video storage 106, and a processing unit 108. Other optional components of the camera system 102 such as control interfaces, display screen, etc. are omitted from the figure for clarity of description. The image sensor 104 captures digital images or video. Because video comprises a sequence of images (or "frames"), description herein referring to images can be similarly applied to video and vice versa. Each captured image comprises a two-dimensional array of pixels. The captured frames depict a "scene," which may include, for example, landscape, people, objects, etc represented by the captured pixels. Each pixel represents a depicted point in a scene captured in the digital video. Furthermore, each pixel is located at a pixel location, referring to, for example, (x,y) coordinates of the pixel within the frame. For example, a pixel may comprise {Red, Green, Blue} (RGB) values describing the relative intensities of the colors sensed by the image sensor 104 at a particular set of (x,y) coordinates in the frame. In various embodiments, the image sensor 104 may capture video suitable for providing output videos having high definition resolutions (for example, 4 k resolution, 2 k resolution,

Appendix: '413 Patent (Ex. 1001)

- Ex. 1001, Cols. 3-4

3
1080p, 1080i, 960p, or 720p), standard definition resolutions, or other resolutions. The image sensor 104 may capture video at various frame rates such as, for example, 120 frames per seconds (FPS), 60 fps, 48, fps, or 30 fps, although other frame rates may also be used. Additionally, the image sensor 104 may include a lens that allows for wide-angle or ultra wide-angle video capture having fields of view of, for example, 90 degrees, 127 degrees, or 170 degrees, although other field of view angles may also be used.

Image frames captured by the image sensor 102 may be temporarily or permanently stored in the video storage 106 which may comprise, for example, a hard drive or a solid-state memory device (e.g., a memory card or universal serial bus (USB) memory device). The processing unit 108 processes images to convert between aspect ratios as will be described in further detail below. In one embodiment, the processing unit 108 comprises a processor 112 and a memory 114. The memory 114 comprises a non-transitory computer-readable storage medium that stores computer-executable program instructions for aspect ratio conversion, among other functions. To perform the aspect ratio conversion described herein, the processor 112 loads program instructions from the memory 114 and executes the instructions. Alternatively, the processing unit 108 could implement the aspect ratio conversion in hardware, firmware, or a combination of hardware, firmware, and/or software.

In one embodiment, the processing unit 108 converts between a source aspect ratio (i.e., the aspect ratio of the original images captured by the image sensor 104) and a target aspect ratio (i.e., the aspect ratio of the image for storing or outputting by the camera system 102). In one

storing or outputting by the camera system 102). In one embodiment, the source aspect ratio makes full use of the available field of view of the image sensor 104 (remaining sensitive to mapping distortion effects). For example, in a camera system 102 having a 4:3 image sensor, the image sensor 104 captures the native content according to a 4:3 aspect ratio, although other source aspect ratios are also possible. The processing unit 108 then uses resolution compression and/or stretching in the appropriate axes to convert the captured images in the source aspect ratio to the target aspect ratio. In one embodiment, the conversion is achieved using a resolution remapping that applies a non-linear function across the image so as to minimize or reduce undesirable key feature aspect ratio distortions. Furthermore, in one embodiment, the aspect ratio conversion does not result in any loss of field of view.

In one embodiment, the aspect ratio conversion is performed by a warp module of the processing unit 108. In addition to performing the aspect ratio conversion, the warp module may perform other functions such as, for example, reducing optical lens distortions such as typical barrel and pincushion distortions.

Aspect Ratio Conversion
Examples of techniques for aspect ratio conversion are discussed in further detail below. For the sake of clarity, the following description discusses examples involving conversions between a source aspect ratio of 4:3 and a target aspect ratio of 16:9. However, the described embodiments similarly apply to any source or target aspect ratios.

FIG. 2A illustrates an example embodiment of a process for converting from a source aspect ratio to a target aspect ratio in a camera system. In the process of FIG. 2A, an input image is obtained 202 in the source aspect ratio (e.g., a 4:3 aspect ratio). The image is then scaled 204 from the source aspect ratio (e.g., the 4:3 aspect ratio) to the target aspect ratio (e.g., a 16:9 aspect ratio) using a linear scaling func-

4
tion. A non-linear warping is then applied 206 to the scaled image. FIG. 2B illustrates the process of FIG. 2A as applied to an example image 212 containing an array of circles. As can be seen, the scaling function applied in step 204 causes the circles to be stretched horizontally and compressed vertically as shown in example image 214. Because the scaling function applied in step 204 is linear, each circle is stretched/compressed equally independent of horizontal and vertical position. In the warping step 206, a non-linear warping is applied which results in the circles being stretched/compressed by different amounts depending on their horizontal and vertical positions resulting in example image 216.

As can also be observed from FIG. 2B, the scaling and warping steps 204, 206 preserves the full field of view in the resulting image 216 relative to the original image 212. To achieve this, the warp module is configured so that some areas of the image are fully corrected geometrically, while some other areas are distorted. In the illustrated example, the center of the image is chosen to be the region of interest that is most fully corrected and the distortion increases in each direction as the frame boundary is approached. Thus, in image 216, the center circle is fully corrected so that it maintains the original circular shape of the corresponding circle in the input image 212. In the vertical direction, the circles become more distorted (e.g., more vertically compressed) in the as they approach the top and bottom edges. In the horizontal direction, the circles become more distorted (e.g., more horizontally stretched) as they approach the left and right edges. As a result, the circles in the corners exhibit the most distortion, because they have the highest amount of both vertical compression and horizontal stretching. In other embodiments, a different part of the image may be selected as the area most fully corrected, or in the case, of video, the area may dynamically change between images (e.g., to follow a moving object).

The distribution of distortion levels in different portions of the image may furthermore be tuned in one embodiment to control the size of objects as they pan from horizontally within a scene of video. Fully correcting the geometrical distortion in the center of the image would result in objects looking small in the center and very large on the sides. Thus, in the case of a video, an object may noticeably grow in size as it moves from the center to the edges of the display. In one embodiment, the warping function in step 206 is tuned to reduce this effect.

In an alternative embodiment, the scaling and warping functions applied in steps 204, 206 could be performed in reverse order, such that the warping function in step 206 is applied first followed by scaling in step 204. In another embodiment, the scaling and warping are combined into a single function such that only a single transformation is applied to the image that achieves both the scaling and warping. For example, in FIG. 2B, image 212 may be converted directly to image 216 via a single function without producing the intermediate image 214.

As will be apparent to those skilled in the art, an equitable horizontal remapping can be performed in manners similar to those described herein for vertical remapping if it is desired to decrease image width without decreasing horizontal image field of view and without creating undesirable key feature distortions. For example, images having a source aspect ratio of 2.39:1 may be converted to a 16:9 aspect ratio in this similar manner.

The scaling step (e.g., step 204 of FIG. 2A) may be achieved either by scaling down one dimension (either the horizontal or vertical one, depending on the formats being

Appendix: '413 Patent (Ex. 1001)

- Ex. 1001, Cols. 5-6

5 converted) or scaling up the orthogonal direction. Scaling up has the benefit of preserving one dimension and only interpolates existing information to extend the other dimension, meaning all resolution is preserved. On the contrary, scaling down causes a loss of information.

FIG. 3 illustrates example images illustrating the aspect ratio conversion technique described herein. Image 302 illustrates an original image captured with a 4:3 aspect ratio. Image 304 illustrates a conventional cropping of the image to fit it to a 16:9 aspect ratio. As can be seen, the field of view is reduced with this technique and a portion of the image is lost. Image 306 illustrates the original image converted according to the techniques described herein. As can be seen, the aspect ratio conversion is achieved while maintaining the full field of view. Although some distortion may be introduced, the distortion is concentrated near the boundaries of the image (away from the center) and perceivable distortion is limited to an acceptable level.

FIG. 4 illustrates additional example images representing a checkerboard pattern, where image 502 represent an original 4:3 image, image 504 represents an image that has been cropped to achieve the 16:9 aspect ratio, and image 506 illustrates an image converted according to the techniques described herein. As can be seen, the perceivable distortion is small or zero near the center of image and increases somewhat at the edges and corners.

Examples of Warping Functions

In one embodiment, the warping function involves both a horizontal and vertical warping. For convenience of description, the horizontal and vertical warping is described as two steps, e.g., first horizontal then vertical. However, both horizontal and vertical warping can be executed in one step, for example, by applying a single transformation to the image. Furthermore, due to the symmetrical nature of the distortion, vertical and horizontal distortion profiles can be swapped.

An example of a horizontal warping function is illustrated in FIG. 5. In this graph, the horizontal coordinate of a pixel is expressed as a percentage of the horizontal pixel position relative to the horizontal width of the image, where the percentage increases in a direction away from a reference edge. The vertical coordinate is similarly expressed. The amount of distortion is expressed in terms of the horizontal distortion offset ΔH that a point in the image moves in a direction away from the reference edge divided by the horizontal image size H. Since the warping function is horizontally symmetric in this embodiment, the reference edge can be either the right edge or the left edge. In this embodiment, the horizontal warp is function of the horizontal coordinate only and is independent of vertical position. The horizontal warp distortion stretches pixels horizontally in a manner that corrects for the "short and fat" effect introduced by 4:3 to 16:9 stretching, or similarly undesirable effects introduced by other aspect ratio conversions. In one embodiment of the horizontal warp, pixels along vertical edges are not offset (the offset value is zero) because there is no information beyond the edges and it may be undesirable to introduce padding (e.g., black vertical lines) on the sides of the image. The magnitude of the distortion is symmetric in one embodiment, with pixels on the left half of the image shifting to the right and pixels on the right half of the image shifting to the left to compensate for the horizontal stretching effect. Since the center receives the contributions of pixels from both sides, the center region of the image is corrected the most, while the geometry on the sides is distorted to a small extent. This degradation in image quality

6 along the sides can be compensated for to some extent in the vertical warping, as will be discussed below.

In one embodiment, the amount of horizontal correction applied can be manually tuned to reduce perceivable distortion in the output image. For example, fully correcting the geometrical distortion in the center may cause the objects in the center to be much smaller than on the sides. Thus, in one embodiment, the amount of horizontal correction can be tuned in combination with the vertical correction in order to preserve the width of objects throughout the horizontal plane.

In one embodiment, a similarly shaped function is used for vertical warping. In another embodiment, the vertical warping function is modified so that the amount of compensation in the vertical direction also depends on horizontal position. FIG. 6 illustrates an example of a vertical warping function, where the amount of distortion is expressed in terms of the vertical distortion offset ΔV indicating an amount that a pixel moves in a vertical direction away from a reference edge (e.g., the top edge or the bottom edge) divided by the vertical image size V. In one embodiment, vertical warping is a function of both the horizontal and vertical coordinates.

As mentioned above, a goal of the vertical distortion is to further correct the geometry that was not already corrected by horizontal distortion. The image center could be fully corrected using the horizontal distortion only (with no vertical warping). However, spreading the correction along both horizontal and vertical axis effectively improves the geometry of not only the center, but also the top, bottom, left and right image sides. The most distorted regions in this embodiment are the four image corners, but this distortion does not substantially reduce perceived image quality. Having residual distortion in the corners while correcting the rest

35 in the image results in a much more natural-looking effect since key scene features are usually not framed in the corners. Examples of this effect can be visible, however, in FIG. 4, where the squares in the corners are more rectangular (as opposed to square), and FIG. 3 where the diver's right flipper is somewhat stretched. However, as a good trade off for these corner distortions, it can be observed that the central rows of squares in FIG. 4 maintain a relatively consistent width throughout the image. Furthermore, the diver's body in FIG. 3 has proportions that are relatively preserved across the image width.

In one embodiment, tunable parameters can be controlled to adjust the vertical distortion along both horizontal and vertical directions, resulting in the 3D-wave in FIG. 7. In one embodiment, pixels are not offset along the horizontal edges to avoid introducing horizontal padding (e.g., black bands along the edges).

In the description above, the horizontal and vertical distortions have been decoupled for the sake of clarity. However, both distortions could be implemented in a single step. Similarly, in the description above, the aspect ratio conversion (4:3 to 16:9 in this example) has been decoupled from warp distortion, but these could also be combined into a single step.

Example equations are provided below for implementing an example warping function applied in step 206 of FIG. 2. Those of ordinary skill in the art will understand that variations of these equations may also be used, and other implementations may apply that result in functions having substantially similar characteristics.

As explained above, the horizontal warp distortion in one embodiment is a function of the horizontal position only. It can therefore be represented in terms of a horizontal coordinate

Appendix: '413 Patent (Ex. 1001)

Ex. 1001, Cols. 7-8

7
dinate h in an equation such as the example equation below,
or an equation substantially similar to the equation below:
$$\Delta h/H = 0.1107h^6 + 0.534h^5 - 1.7813h^4 + 1.3167h^3 - 0.0722h^2 - 0.1073h$$

The vertical warp distortion, in one embodiment, is a
function of both the horizontal and vertical positions. There-
fore, the vertical warp distortion can be represented as a
family of equations $\Delta V/V_n$, where "n" is the horizontal
coordinate expressed as a percent of the total horizontal
image size. Intermediate horizontal coordinates can be inter-
polated from the example equations given below, or equa-
tions substantially similar to these:
$$\Delta v/V_0 = -0.0749v^6 - 0.3167v^5 + 1.0939v^4 - 0.8113v^3 + 0.0408v^2 + 0.0678v$$

$$\Delta v/V_{10} = -0.0658v^6 - 0.2785v^5 + 0.9617v^4 - 0.7132v^3 + 0.0359v^2 + 0.0596v$$

$$\Delta v/V_{20} = -0.0568v^6 - 0.2402v^5 + 0.8294v^4 - 0.6152v^3 + 0.0309v^2 + 0.0514v$$

$$\Delta v/V_{30} = -0.0444v^6 - 0.188v^5 + 0.6491v^4 - 0.4814v^3 + 0.0242v^2 + 0.0403v$$

$$\Delta v/V_{40} = -0.0337v^6 - 0.1427v^5 + 0.4928v^4 - 0.3655v^3 + 0.0184v^2 + 0.0306v$$

$$\Delta v/V_{50} = -0.0296v^6 - 0.1253v^5 + 0.4327v^4 - 0.3209v^3 + 0.0161v^2 + 0.0268v$$

$$\Delta v/V_{60} = -0.0337v^6 - 0.1427v^5 + 0.4928v^4 - 0.3655v^3 + 0.0184v^2 + 0.0306v$$

$$\Delta v/V_{70} = -0.0444v^6 - 0.188v^5 + 0.6491v^4 - 0.4814v^3 + 0.0242v^2 + 0.0403v$$

$$\Delta v/V_{80} = -0.0568v^6 - 0.2402v^5 + 0.8294v^4 - 0.6152v^3 + 0.0309v^2 + 0.0514v$$

$$\Delta v/V_{90} = -0.0658v^6 - 0.2785v^5 + 0.9617v^4 - 0.7132v^3 + 0.0359v^2 + 0.0596v$$

$$\Delta v/V_{100} = -0.0749v^6 - 0.3167v^5 + 1.0939v^4 - 0.8113v^3 + 0.0408v^2 + 0.0678v$$

The warp distortion described above may be used with
cameras producing either rectilinear images (without distor-
tion) or images affected by optical lens distortion character-
ized by the following equation:
$$(FOV) = (21.904) * (IH) - (0.6852) * (IH)^2$$

where FOV is the field of view and IH is image height. As
will be apparent to those of ordinary skill in the art, the
equations above may vary when used with lenses having
different distortion characteristics.
As described above, the above equations can be combined
with a scaling function to generate a transformation that
performs both scaling and warping.
In alternative embodiments, different equations may be
used to achieve similar distortion characteristics. As men-
tioned above, the symmetric nature of the distortion and the
fact that it can be expressed as a percent of image coordi-
nates allows swapping horizontal and vertical distortions.
Thus, in an alternative embodiment, the equations above for
the horizontal axis can instead be applied in the vertical
direction and vice versa.
Furthermore, in other alternative embodiments, the equa-
tions may change dynamically in different frames of a video.
For example, warping may be applied in a manner that
reduces distortion applied to a particular object in motion

8
across different frames of a video since the viewer is likely
to focus attention on the object in motion.
Although the example embodiments described herein
relate to conversion between a 4:3 aspect ratio and a 16:9
aspect ratio, other aspect ratios may be used for the target
and source aspect ratios. For example, in alternative embodi-
ments the source aspect ratio can comprise a 4:3 aspect ratio,
a 16:9 aspect ratio, a 1.85:1 aspect ratio, a 2.39:1 aspect
ratio, a 3:2 aspect ratio, a 5:3 aspect ratio, a 5:4 aspect ratio,
a 1:1 aspect ratio or any other aspect ratio. Similarly, in
alternative embodiments, the target aspect ratio can com-
prise a 4:3 aspect ratio, a 16:9 aspect ratio, a 1.85:1 aspect
ratio, a 2.39:1 aspect ratio, a 3:2 aspect ratio, a 5:3 aspect
ratio, a 5:4 aspect ratio, a 1:1 aspect ratio or any other aspect
ratio.
ADDITIONAL CONFIGURATION
CONSIDERATIONS
Throughout this specification, some embodiments have
used the expression "coupled" along with its derivatives.
The term "coupled" as used herein is not necessarily limited
to two or more elements being in direct physical or electrical
contact. Rather, the term "coupled" may also encompass two
or more elements are not in direct contact with each other,
but yet still co-operate or interact with each other, or are
structured to provide a thermal conduction path between the
elements.
Likewise, as used herein, the terms "comprises," "com-
prising," "includes," "including," "has," "having" or any
other variation thereof, are intended to cover a non-exclusive
inclusion. For example, a process, method, article, or appa-
ratus that comprises a list of elements is not necessarily
limited to only those elements but may include other ele-
ments not expressly listed or inherent to such process,
method, article, or apparatus.

In addition, use of the "a" or "an" are employed to
describe elements and components of the embodiments
herein. This is done merely for convenience and to give a
general sense of the invention. This description should be
read to include one or at least one and the singular also
includes the plural unless it is obvious that it is meant
otherwise.
Finally, as used herein any reference to "one embodi-
ment" or "an embodiment" means that a particular element,
feature, structure, or characteristic described in connection
with the embodiment is included in at least one embodiment.
The appearances of the phrase "in one embodiment" in
various places in the specification are not necessarily all
referring to the same embodiment.
Upon reading this disclosure, those of skill in the art will
appreciate still additional alternative structural and func-
tional designs for a camera expansion module as disclosed
from the principles herein. Thus, while particular embodi-
ments and applications have been illustrated and described,
it is to be understood that the disclosed embodiments are not
limited to the precise construction and components disclosed
herein. Various modifications, changes and variations, which
will be apparent to those skilled in the art, may be made in
the arrangement, operation and details of the method and
apparatus disclosed herein without departing from the spirit
and scope defined in the appended claims.
The invention claimed is:
1. An image capture apparatus, comprising:
an image sensor configured to capture an input image
having a source aspect ratio, the input image including

Appendix: '413 Patent (Ex. 1001)

• Ex. 1001, Cols. 9-11

9

pixels located at input positions defined along a first axis and a second axis, the first axis perpendicular to the second axis;
a display coupled to one or more physical processors; and
the one or more physical processors configured by machine-readable instructions to:
obtain the input image;
determine a portion within the input image, the portion including a subset of the pixels; and
apply a transformation to the input image to generate an output image, the output image having a target aspect ratio different than the source aspect ratio, the output image including the pixels located at output positions defined along the first axis and the second axis, wherein:
the transformation non-uniformly shifts the pixels from the input positions to the output positions based on (1) the input positions along the first axis, and (2) the input positions along the second axis; and
the transformation non-uniformly shifts the pixels such that differences between the input positions and the output positions of the subset of the pixels within the portion are less than differences between the input positions and the output positions of others of the pixels; and
present the output image on the display.

2. A system that converts between aspect ratios, the system comprising:
one or more physical processors configured by machine-readable instructions to:
obtain an input image, the input image having a source aspect ratio, the input image including pixels located at input positions defined along a first axis and a second axis, the first axis perpendicular to the second axis;

.....
determine a portion within the input image, the portion including a subset of the pixels; and
apply a transformation to the input image to generate an output image, the output image having a target aspect ratio different than the source aspect ratio, the output image including the pixels located at output positions defined along the first axis and the second axis, wherein:
the transformation non-uniformly shifts the pixels from the input positions to the output positions based on (1) the input positions along the first axis, and (2) the input positions along the second axis; and
the transformation non-uniformly shifts the pixels such that differences between the input positions and the output positions of the subset of the pixels within the portion are less than differences between the input positions and the output positions of others of the pixels.

3. The system of claim 2, wherein the transformation includes:
a scaling function to uniformly stretch or compress the input image along the first axis; and
a warping function to non-uniformly warp the input image along the second axis.

4. The system of claim 3, wherein the scaling function and the warping function preserves the input positions of the subset of the pixels as the output positions of the subset of the pixels.

5. The system of claim 3, wherein the scaling function changes the source aspect ratio to the target aspect ratio.

10

6. The system of claim 2, wherein the one or more physical processors are included within a camera.

7. The system of claim 6, wherein the output image is presented on a display of the camera.

8. The system of claim 2, wherein the portion corresponds to a center portion of the input image.

9. The system of claim 2, wherein the portion corresponds to a non-center portion of the input image.

10. The system of claim 2, wherein the transformation non-uniformly shifts the pixels from the input positions to the output positions further based on distances between the input positions and the portion.

11. The system of claim 2, wherein the one or more physical processors are further configured by machine-readable instructions to determine the portion based on a location of an object within the input image such that the portion includes the object.

12. The system of claim 2, wherein the input image includes a video frame of a video.

13. A method for converting between aspect ratios, the method performed by a computing system including one or more physical processors, the method comprising:
obtaining, by the computing system, an input image, the input image having a source aspect ratio, the input image including pixels located at input positions defined along a first axis and a second axis, the first axis perpendicular to the second axis;
determining a portion within the input image, the portion including a subset of the pixels; and
applying, by the computing system, a transformation to the input image to generate an output image, the output image having a target aspect ratio different than the source aspect ratio, the output image including the pixels located at output positions defined along the first axis and the second axis, wherein:

the transformation non-uniformly shifts the pixels from the input positions to the output positions based on (1) the input positions along the first axis, and (2) the input positions along the second axis; and
the transformation non-uniformly shifts the pixels such that differences between the input positions and the output positions of the subset of the pixels within the portion are less than differences between the input positions and the output positions of others of the pixels.

14. The method of claim 13, wherein the transformation includes:
a scaling function to uniformly stretch or compress the input image along the first axis; and
a warping function to non-uniformly warp the input image along the second axis.

15. The method of claim 14, wherein the scaling function and the warping function preserves the input positions of the subset of the pixels as the output positions of the subset of the pixels.

16. The method of claim 13, wherein the portion corresponds to a center portion of the input image.

17. The method of claim 13, wherein the portion corresponds to a non-center portion of the input image.

18. The method of claim 13, wherein the transformation non-uniformly shifts the pixels from the input positions to the output positions further based on distances between the input positions and the portion.

19. The method of claim 13, further comprising determining the portion based on a location of an object within the input image such that the portion includes the object.

11

20. The method of claim 13, wherein the input image includes a video frame of a video.

* * * * *

Appendix: Meulen (Ex. 1007)

Appendix: Meulen (Ex. 1007)

- Ex. 1007, Figs. 1-3

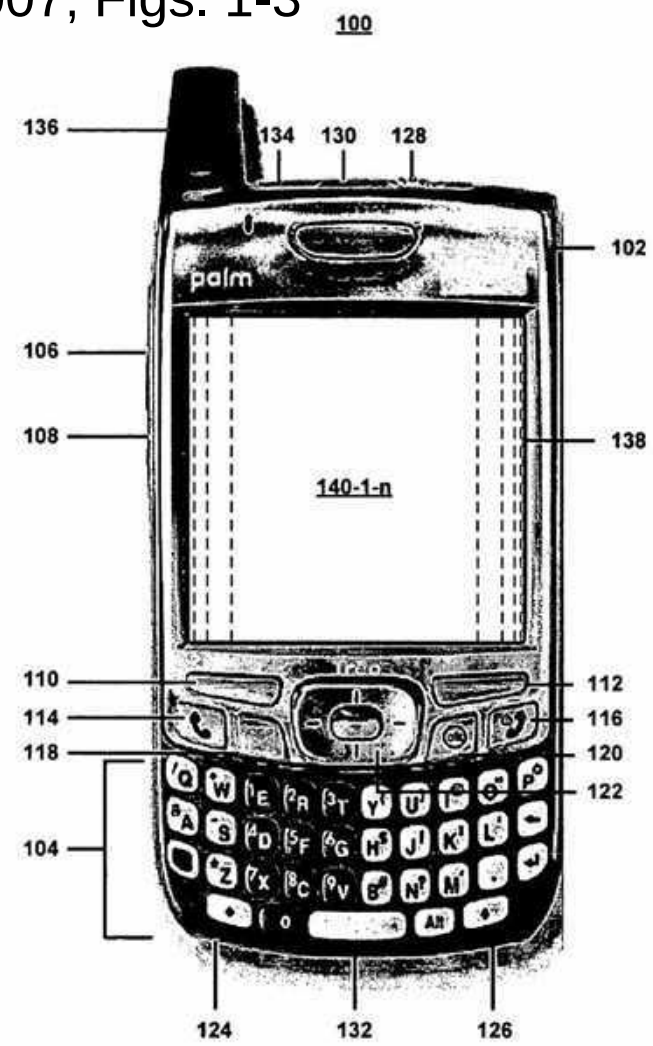


FIG. 1

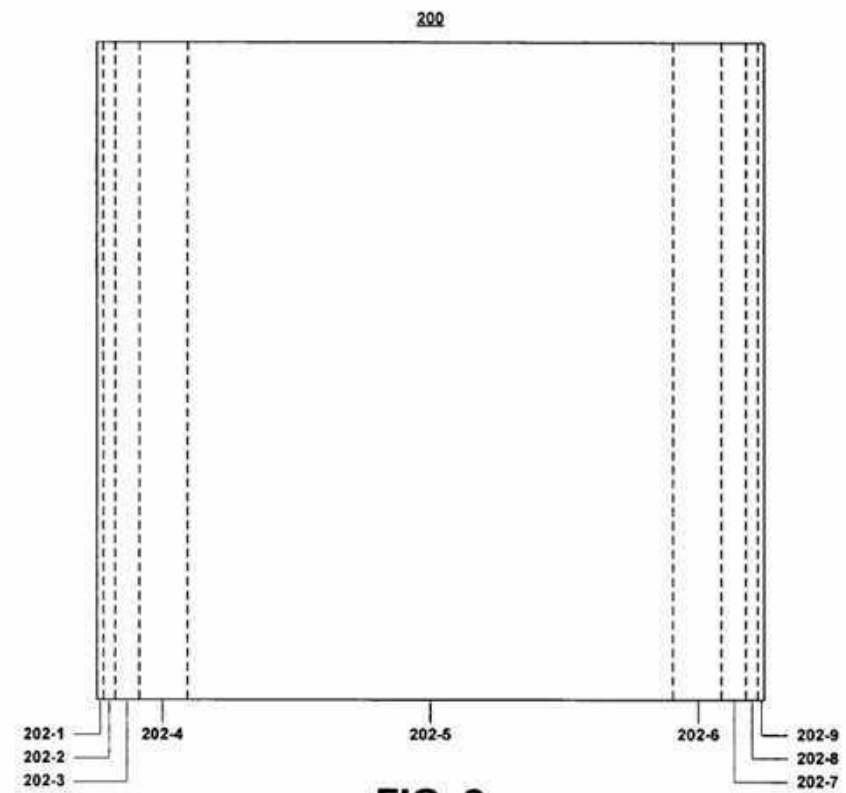


FIG. 2

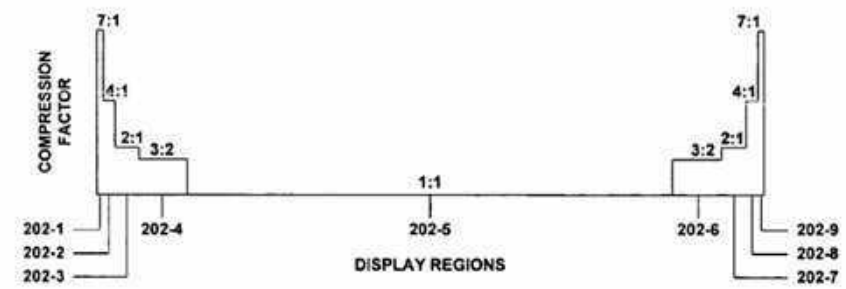
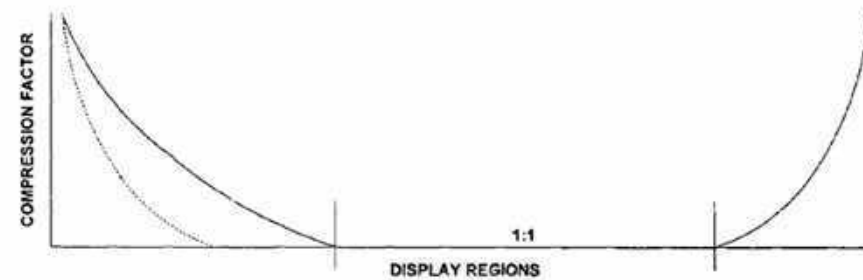
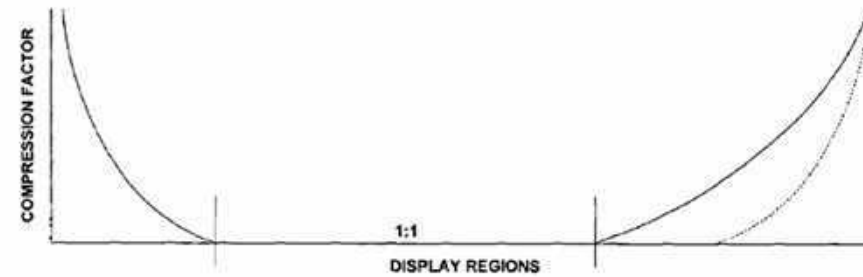
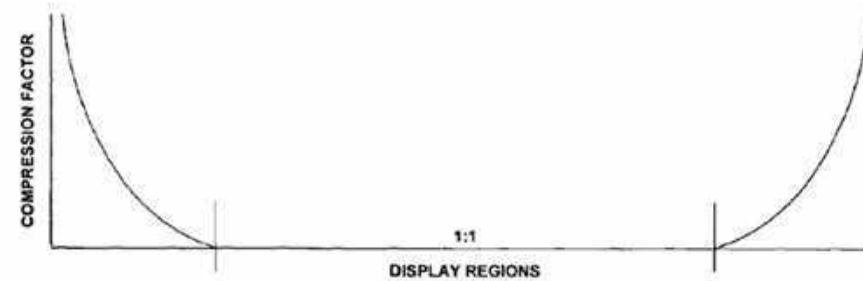


FIG. 3

Appendix: Meulen (Ex. 1007)

- Ex. 1007, Figs. 4A-4C



Appendix: Meulen (Ex. 1007)

- Ex. 1007, Fig. 5

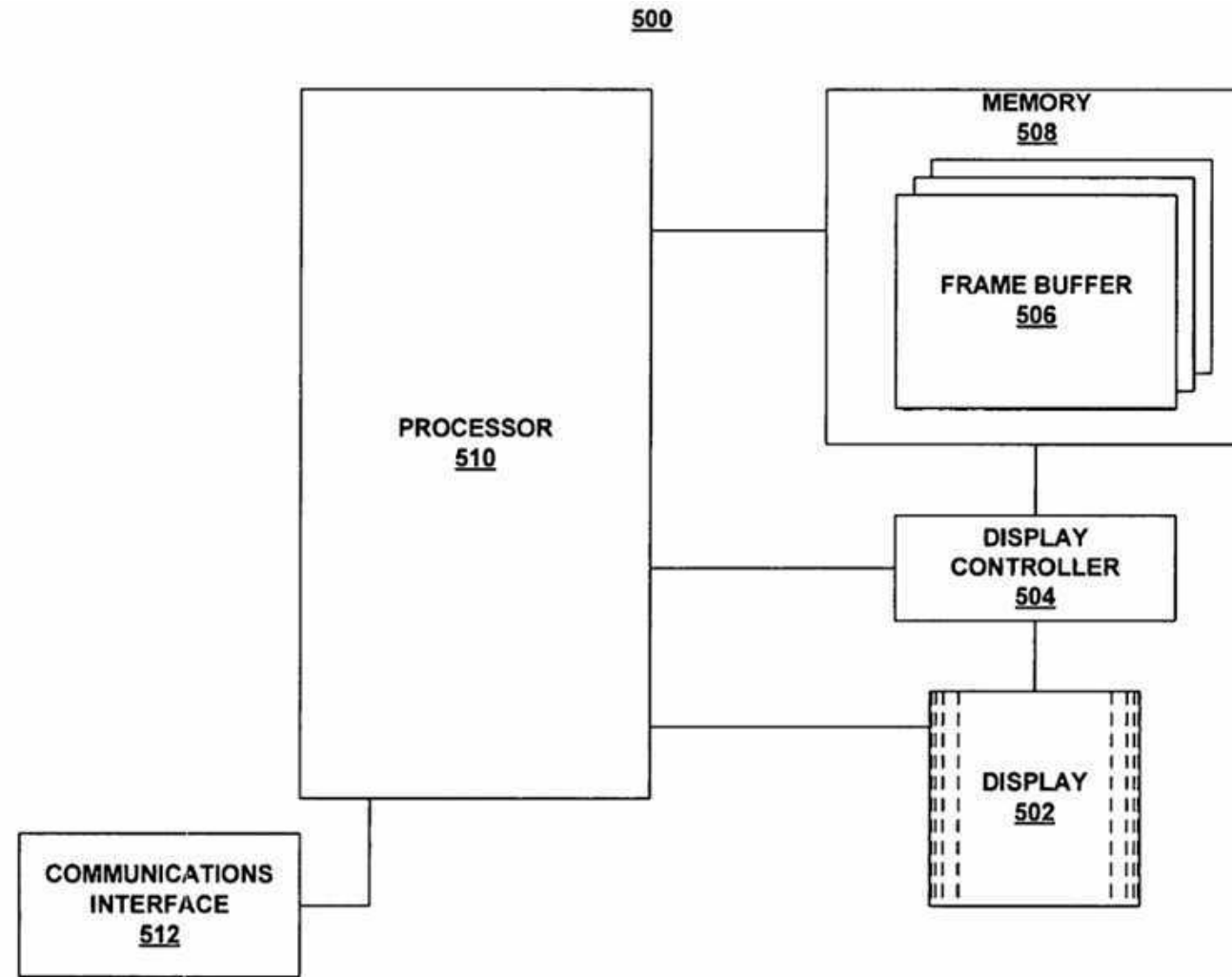


FIG. 5

Appendix: Meulen (Ex. 1007)

- Ex. 1007, Cols. 1-2

1
PANORAMIC DISPLAY FOR A WIRELESS
DEVICE

BACKGROUND

A wireless device such as a handheld computer and/or mobile telephone may incorporate features such as a color display screen, a digital camera, Web access, and multimedia capability. The display screen for a wireless device may be implemented, for example, by a liquid crystal display (LCD) containing a matrix of pixels and employing a back light to project light through the matrix.

The resolution or aspect ratio of a display screen generally refers to the number of pixels that can be displayed, typically expressed as a product of the number of columns and the number or rows. In many cases, a wireless device may include a square display screen having an aspect ratio of 1:1 with a resolution of 320×320 or 480×480, for example. Most media sources, however, have a non-square aspect ratio such as 4:3, 3:2, 16:9, or even 2:1. For example, the aspect ratio for standard television as well as for many web sites and multimedia products is 4:3. High-definition television (HDTV) and many digital video cameras use an aspect ratio of 16:9. Digital video disc (DVD) players may be designed to use an aspect ratio of either 4:3 or 16:9.

Conventional best fit algorithms such as those defined by the Synchronized Multimedia Integration Language (SMIL) specification may be suitable for fitting content to non-square displays, but produce a black bar on square displays. Consequently, there may be a need for improved display systems and techniques for a wireless device.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 illustrates one embodiment of a wireless device. FIG. 2 illustrates one embodiment of a panoramic display. FIG. 3 illustrates one embodiment of compression factors. FIGS. 4A-C illustrate one embodiment of compression factors.

FIG. 5 illustrates one embodiment of a computing system.

DETAILED DESCRIPTION

Various embodiments may be directed to systems and techniques that may potentially improve the performance of a wireless device to display content from a media source. In one embodiment, for example, a wireless device may include a display comprising a plurality of display regions. The plurality of display regions may extend horizontally between the edges of the display, and each of the display regions may be defined by a number of pixels in a horizontal direction. The display regions may comprise a central display region and multiple lateral display regions arranged to provide a panoramic display.

In various implementations, image compression and/or scaling may be applied to media content having an aspect ratio and/or resolution that is different from the aspect ratio and/or resolution of the display. Image compression may be applied, for example, to media content having an aspect ratio of 3:2 with a resolution of 480×320 that is to be displayed by a display having an aspect ratio of 1:1 with a resolution of 320×320. The image compression may be applied progressively to the media content that is to be displayed in one or more of the lateral display regions based on the proximity of the lateral display region to the edge of the display. For example, more image compression may be applied to the media content which is to be displayed at or near the edges of

2

the display than to the media content which is to be displayed at or near the central region. In such implementations, a wireless device having a square or substantially square display may be arranged to emulate a non-square display and to better match the aspect ratio of typical video content without cropping. Accordingly, a user may realize enhanced products and services.

FIG. 1 illustrates one embodiment of a wireless device 100. As shown, the wireless device 100 may be implemented as a combination handheld computer and mobile telephone or smartphone. Examples of smartphones include, for example, Palm® products such as Palm® Treo™ smartphones. Although some embodiments may be described with the wireless device 100 implemented as a smartphone by way of example, it may be appreciated that the embodiments are not limited in this context. For example, the wireless device 100 may comprise, or be implemented as, a handheld computer, mobile telephone, personal digital assistant (PDA), combination mobile telephone/PDA, data transmission device, one-way pager, two-way pager, and so forth.

The wireless device 100 may comprise a housing 102. The housing 102 may include one or more materials such as plastic, metal, ceramic, glass, and so forth, suitable for enclosing and protecting the internal components of the wireless device 100. The wireless device 100 may comprise various internal components such as, for example, a processor, a memory, one or more transceivers, one or more printed circuit board (PCBs), and so forth.

The wireless device 100 may comprise various input/output (I/O) devices such as a keyboard, keys, buttons, switches, a microphone, an audio headset, a camera, a touch-sensitive display screen, a stylus, and so forth. As shown in FIG. 1, for example, the wireless device 100 may comprise an alphanumeric keyboard 104 having a QWERTY key layout and an

integrated number dial pad. The wireless device 100 may comprise various buttons such as, for example, a volume button 106, a customizable button 108, a left action button 110, a right action button 112, a phone/send button 114, a power/end button 116, a start button 118, an OK button 120, and a navigation button 122. The wireless device 100 may comprise an audio port 124 to connect an audio headset, a microphone 126, a ringer on/off switch 128 having a vibrate mode, and an expansion slot 130 to support a multimedia and/or memory card, for example.

The wireless device 100 may comprise a serial connection port 132, an infrared port 134, and/or integrated Bluetooth® wireless capability to enable wired (e.g., USB cable) and/or wireless connection to a local computer system, such as a local personal computer (PC). In various implementations, the wireless device 100 may be arranged to transfer and/or synchronize information with the local computer system. The wireless device 100 also may comprise a rechargeable battery, such as a removable and rechargeable lithium ion battery, and an AC adapter.

The wireless device 100 may comprise an antenna system including one or more antennas, such as an external antenna 136 implemented by a stub antenna, a whip antenna, an extendable antenna, and so forth. In various embodiments, the antenna system may comprise one or more internal antennas such as a planar inverted-F antenna, a planar inverted-L antenna, an inverted-F antenna with a helical structure, an inverted-L antenna with a helical structure, a monopole antenna, a meandered monopole antenna, a dipole antenna, a balanced antenna, a printed helical antenna, a chip antenna, a ceramic antenna, and so forth.

In various embodiments, the antenna system may be tuned for operating at one or more frequency bands such as the

Appendix: Meulen (Ex. 1007)

• Ex. 1007, Cols. 3-4

3
824-894 Megahertz (MHz) frequency band for GSM operations, the 1850-1990 MHz frequency band for Personal Communications Services (PCS) operations, the 1575 MHz frequency band for Global Positioning System (GPS) operations, the 824-860 MHz frequency band for NAMPS operations, the 1710-2170 MHz frequency band for Wide-band CDMA/Universal Mobile Telephone System (WCDMA/UMTS), ISM band in 2.4 GHz range for WiFi and Bluetooth, and other frequency bands. In various implementations, the antenna system may be used to implement spatial diversity techniques such as Evolution Data Optimized (EVDO) diversity at both 800 MHz (cellular) and 1900 MHz (PCS) bands.
The wireless device 100 may comprise a display 138. The display 138 may be implemented by a LCD or other type of suitable visual interface. The display 138 may comprise, for example, a touch-sensitive color (e.g., 16-bit color) display screen. In various implementations, the display 138 may comprise a thin-film transistor (TFT) LCD including embedded transistors. In such implementations, the display 138 may comprise a transistor for each pixel to implement an active matrix. While the embodiments are not limited in this context, an active matrix requires lower current to trigger pixel illumination and is more responsive to change than a passive matrix.
The display 138 may be arranged to display media source content received by the wireless device 100. In various embodiments, the display 138 may be arranged to display the media source content in one or more modes, such as in a panoramic mode, a wide mode, and/or an optimized mode. In such embodiments, the modes may be selected under user-control.

The media source content may comprise, for example, image information, video information, audio/visual (A/V) information, and/or graphics information from a media source. In various embodiments, the media source content may comprise data derived from or associated with one or more images, image files, image groups, pictures, digital photographs, videos, video clips, video files, video sequences, video feeds, video streams, movies, broadcast programming, web pages, user interfaces, graphics, windows, alerts, games, regions, objects, frames, slices, macroblocks, blocks, pixels, sub-pixels, signals, and so forth.
In various implementations, the media source content received and to be displayed may comprise pixels derived from or associated with one or more static or video images. The pixels may comprise, for example, red-green-blue (RGB) pixels and/or luminance-chrominance pixels (e.g., YUV, YCC) and may include real and/or integer values. The embodiments are not limited in this context.
The media source generally may comprise any source capable of delivering media source content to the wireless device 100. An example of a media source may include a source for static or video image information, such as from a computer to a display. The media source may include a server, such as a web server arranged to deliver web content. Another example of a media source may include a source for A/V information such as television signals. The media source may be arranged to deliver standard analog television content, digital television content, HDTV content, and so forth. The media source may provide broadcast or streaming analog or digital video content. The media source may include a device arranged to deliver pre-recorded media content stored in various formats, such as a DVD device, a Video Home System (VHS) device, a digital VHS device, a digital camera, video camera, a portable media player, a gaming device, and so forth.

4
In various embodiments, the wireless device 100 may be arranged to receive media source content through one or more wireless wide area network (WWAN) data communication services. Examples of cellular data communication systems offering WWAN data communication services may include a GSM with General Packet Radio Service (GPRS) systems (GSM/GPRS), CDMA/1xRTT systems, Enhanced Data Rates for Global Evolution (EDGE) systems, Evolution Data Only or EVDO systems, Evolution for Data and Voice (EV-DV) systems, High Speed Downlink Packet Access (HSDPA) systems, and so forth. The embodiments are not limited in this context.
In various embodiments, the media source content may comprise an aspect ratio and/or a resolution that differs from the aspect ratio and/or the resolution of the display 138. In one embodiment, for example, the media source content may have an aspect ratio of 3:2 with a resolution of 480x320, and the display may have an aspect ratio of 1:1 with a resolution of 320x320. In another embodiment, for example, the media source content may have an aspect ratio of 2:1 with a resolution of 640x320, and the display may have an aspect ratio of 3:2 with a resolution of 480x320. The embodiments are not limited in this context.
As shown, the display 138 may comprise a plurality of display regions, such as regions 140-1-n, where n represents a positive integer value. In various embodiments, the plurality of display regions 140-1-n may be arranged to implement a panoramic display for a wireless device 100 having a square or substantially square display 138, such as a display having an aspect ratio of 1:1 with a resolution of 320x320, 480x480, and so forth. The embodiments, however, are not limited in this context. For example, in some embodiments, the display 138 may have an aspect ratio of 3:2 with a resolution of 480x320 or other aspect ratio and/or resolution.

In various embodiments, the media source content may have a different aspect ratio and/or a resolution than the display 138. In such embodiments, image compression and/or scaling may be applied to the media source content that is to be displayed in one or more of the display regions 140-1-n.
The image compression and/or scaling may comprise visual scaling with a factor smaller than one in order to reduce size. In various implementations, the image compression and/or scaling may be applied progressively to the media source content based on the proximity of a display region to the edge of the display 138. For example, more image compression and/or scaling may be applied to the media source content that is to be displayed closer to edges of the display 138, than to the media source content that is to be displayed closer to the center of the display 138.
In various embodiments, the image compression and/or scaling may be applied to the media source content using various physical or logical elements implemented as hardware, software, or any combination thereof, as desired for a given set of design parameters or performance constraints. In some embodiments, for example, the wireless device 100 may comprise display control hardware implemented by one or more processors, controllers, encoder devices, decoder devices, coder/decoder (CODEC) devices, scaling devices, filters, converters, circuits, chips, logic devices, logic gates, switches, registers, semiconductor devices, transistors, or combination thereof.
In some embodiments, for example, the wireless device 100 may comprise display control software implemented by one or more applications, drivers, programs, modules, sub-routines, instruction sets, instructions, computing codes, or combination thereof. The display control software may be implemented according to a predefined computer language

Appendix: Meulen (Ex. 1007)

Ex. 1007, Cols. 5-6

5
for instructing a processor to perform a certain function. Examples of a computer language may include C, C++, Java, BASIC, Perl, Matlab, Pascal, Visual BASIC, assembly language, machine code, micro-code, and so forth.

10
In various implementations, the wireless device 100 may be arranged to process a line of source pixels to generate a compressed and/or scaled line of display pixels. The wireless device 100 may comprise, for example, various physical or logical elements arranged to process media source pixels and to generate display pixels by performing processing operations such as adding, subtracting, multiplying, dividing, sampling, compressing, converting, filtering, scaling, interpolating, and so forth.

15
In various embodiments, the wireless device 100 may be arranged to perform image compression and/or scaling using one or more sample-rate converters implemented by hardware and/or software. In some embodiments, for example, the wireless device 100 may comprise sample-rate converters implemented in hardware by programmable polyphase filters arranged to sample a line of source pixels and to generate a compressed and/or scaled line of display pixels based on a compression factor. When implemented in hardware, the polyphase filters typically may comprise 3 to 7 taps per filter. In some embodiments, for example, the wireless device 100 may comprise sample-rate conversion software to perform sample-rate conversion, such as polyphase filtering. In some implementations, the polyphase filtering may be optimized for multimedia extensions (MMX). The embodiments are not limited in this context.

20
In various implementations, the plurality of display regions 140-*n* may extend horizontally between the edges of the display 138, and each one of the display regions 140-1-*n* may comprise or be defined by a number of pixels in a horizontal direction. Although not limited in this context, the number of display regions 140-1-*n* may comprise an odd number greater than or equal to five (e.g., *n*=5, 7, 9, ...).

6
In one embodiment, for example, the plurality of display regions 140-1-*n* may comprise five display regions (e.g., *n*=5). In this embodiment, the plurality of display regions 140-1-5 may comprise a central display region 140-3, lateral display regions 140-1 and 140-2 to the left of the central display region 140-3, and lateral display regions 140-4 and 140-5 to the right of the central display region 140-3. Each of the plurality of display regions 140-1-5 may comprise or be defined by a number of pixels in a horizontal direction. The total number of pixels defining the display regions 140-1-5 may correspond to the horizontal resolution of the display 138.

Table 1 illustrates an implementation for one embodiment of a display 138 comprising display regions 140-1-5.

TABLE 1						
	Region					Total
	1	2	3	4	5	
Pixels/Region	10	50	200	50	10	320
Compression Ratio	4:1	2:1	none	2:1	4:1	
Source Pixels	40	100	200	100	40	480

In this implementation, the display 138 may have an aspect ratio of 1:1 with a resolution of 320x320, and the media source content to be displayed may have an aspect ratio of 3:2 with a resolution of 480x320. The display regions 140-1-5 may be defined by the 320 horizontal pixels of the display 138. Because the media source content has a different aspect

ratio and/or resolution than the display 138, image compression and/or scaling may be applied to the media source that is to be displayed in one or more of the display regions 140-1-5. In this embodiment, for example, image compression and/or scaling may be applied to the media source content that is to be displayed in the lateral display regions 140-1 and 140-2 and the lateral display regions 140-4 and 140-5. The image compression and/or scaling may be applied progressively. For example, more image compression and/or scaling may be applied to the media source content for the lateral display regions 140-1 and 140-5, which are closer to edges of the display 138, than to the media source content for the lateral regions 140-2 and 140-4.

In some embodiments, the lateral display regions 140-1 and 140-2 to the left of the central region 140-3 may be symmetric with the lateral display regions 140-5 and 140-4 to the right of the central display region 140-3. For example, the number of pixels and/or image compression ratio for each the lateral display regions 140-1 and 140-2 to the left of the central region 140-3 may be symmetric with the number of pixels and/or image compression ratio for each of the lateral display regions 140-5 and 140-4 to the right of the central display region 140-3. In such embodiments, image compression and/or scaling may be applied symmetrically for the lateral display regions 140-1 and 140-5 and for the lateral display regions 140-2 and 140-4.

As shown in Table 1, for example, the central display region 140-3 may comprise 200 pixels and require the media source content to undergo no image compression (e.g., 200 source pixels to 200 display pixels). The lateral display regions 140-1 and 140-5 each may comprise 10 pixels and subject media source content to a 4:1 image compression ratio (e.g., 40 source pixels to 10 display pixels). The lateral display regions 140-2 and 140-4 each may comprise 50 pixels and subject media source content to a 2:1 image compression ratio (e.g., 100 source pixels to 50 display pixels).

The embodiment illustrated by Table 1 may comprise, for example, sample-rate converter circuitry implemented by two symmetrically arranged programmable hardware-based sample-rate converters. One of the hardware-based sample-rate converters may be arranged to generate display pixels for lateral display regions 140-1 and 140-5 based on a 4:1 compression ratio. The other hardware-based sample-rate converter may be arranged to generate display pixels for lateral display regions 140-2 and 140-4 based on a 2:1 compression ratio. The embodiments, however, are not limited in this context. For example, the embodiment illustrated by Table 1 may comprise software to perform sample-rate conversion.

FIG. 2 illustrates one embodiment of a panoramic display 200. In various embodiments, the panoramic display 200 may comprise, or be implemented by, the wireless device 100 of FIG. 1. The embodiments, however, are not limited in this context.

As shown, the panoramic display 200 may comprise a plurality of display regions 202-1-9 including a central display region 202-1-5, lateral display regions 202-1-4 to the left of the central display region 202-5, and lateral display regions 202-6-9 to the right of the central display region 202-5. Each of the plurality of display regions 202-1-9 may comprise or be defined by a number of pixels in a horizontal direction. In various implementations, the total number of pixels defining the display regions 202-1-9 may correspond to the horizontal resolution of the panoramic display 200. For example, the display regions 202-1-9 may be defined by the 320 horizontal pixels of a panoramic display 200 having an aspect ratio of 1:1 with a resolution of 320x320.

Appendix: Meulen (Ex. 1007)

- Ex. 1007, Cols. 7-8

7

FIG. 3 illustrates one embodiment of compression factors for a plurality of display regions. In various embodiments, the compression factors may be implemented by the display regions 202-1-9 of the panoramic display 200 of FIG. 2. The embodiments are not limited in this context.

As shown, the image compression and/or scaling may be applied progressively, and the image compression ratios for each of the lateral display regions 202-1-4 to the left of the central region 202-5 may be symmetric with the image compression ratios for each of the lateral display regions 202-9-6 to the right of the central display region 202-5. In this embodiment, the image compression and/or scaling may be applied by four symmetrically arranged programmable hardware-based sample-rate converters. The first hardware-based sample-rate converter may be arranged to generate display pixels for lateral display regions 202-1 and 202-9 based on a 7:1 compression ratio. The second hardware-based sample-rate converter may be arranged to generate display pixels for lateral display regions 202-2 and 202-8 based on a 4:1 compression ratio. The third hardware-based sample-rate converter may be arranged to generate display pixels for lateral display regions 202-3 and 202-7 based on a 2:1 compression ratio. The fourth hardware-based sample-rate converter may be arranged to generate display pixels for display regions 204-1 and 206-9 based on a 3:2 compression ratio. The embodiments, however, are not limited in this context. For example, the embodiments may comprise software to perform sample-rate conversion.

In various embodiments, the panoramic display 200 may be implemented as either low compression or high compression based on the number of pixels and/or image compression ratio used for the plurality of display regions 202-1-9.

Table 2 illustrates a low compression implementation for a panoramic display 200 comprising display regions 202-1-9.

TABLE 2

	Region									
	1	2	3	4	5	6	7	8	9	Total
Pixels/Region	4	8	16	32	200	32	16	8	4	320
Compression Ratio	7:1	4:1	2:1	3:2	none	3:2	2:1	4:1	7:1	
Source Pixels	28	32	32	48	200	48	32	32	28	480

In this low compression implementation, the panoramic display 200 may have an aspect ratio of 1:1 with a resolution of 320x320, and the media source content may have an aspect ratio of 3:2 with a resolution of 480x320. The lateral display regions 202-1 and 202-9 each may comprise 4 pixels and subject the media source content to a 7:1 image compression ratio (e.g., 28 source pixels to 4 display pixels). The lateral display regions 202-2 and 202-8 each may comprise 8 pixels and subject the media source content to a 4:1 image compression ratio (e.g., 32 source pixels to 8 display pixels). The lateral display regions 202-3 and 202-7 each may comprise 16 pixels and subject the media source content to a 2:1 image compression ratio (e.g., 32 source pixels to 16 display pixels). The lateral display regions 202-4 and 202-6 each may comprise 32 pixels and subject the media source content to a 3:2 image compression ratio (e.g., 48 source pixels to 32 display pixels). The central display region 202-5 may comprise 200 pixels and require the media source content to undergo no image compression (e.g., 200 media source pixels to 200 display pixels).

Table 3 illustrates a high compression implementation for a panoramic display 200 comprising display regions 202-1-9.

8

TABLE 3

	Region									
	1	2	3	4	5	6	7	8	9	Total
Pixels/Region	2	4	8	16	260	16	8	4	2	320
Compression Ratio	14:1	8:1	4:1	3:1	none	3:1	4:1	8:1	14:1	
Source Pixels	28	32	32	48	260	48	32	32	28	480

In this high compression implementation, the panoramic display 200 may have an aspect ratio of 1:1 with a resolution of 320x320, and the media source content may have an aspect ratio of 3:2 with a resolution of 480x320. The lateral display regions 202-1 and 202-9 each may comprise 2 pixels and subject the media source content to a 14:1 image compression ratio (e.g., 28 source pixels to 2 display pixels). The lateral display regions 202-2 and 202-8 each may comprise 4 pixels and subject the media source content to an 8:1 image compression ratio (e.g., 32 source pixels to 4 display pixels). The lateral display regions 202-3 and 202-7 each may comprise 8 pixels and subject the media source content to a 4:1 image compression ratio (e.g., 32 source pixels to 8 display pixels). The lateral display regions 202-4 and 202-6 each may comprise 16 pixels and subject the media source content to a 3:1 image compression ratio (e.g., 48 source pixels to 16 display pixels). The central display region 202-5 may comprise 260 pixels and require the media source content to undergo no image compression (e.g., 260 media source pixels to 260 display pixels).

In various embodiments, the central display region 202-5 may comprise an expanded linear area. For example, in the high compression implementation illustrated by Table 3, each of the left lateral display regions 202-1-4 and the right lateral display regions 202-6-9 may be arranged to display 30 pixels (e.g., 2+4+8+16) by compressing and/or scaling 140 horizontal source pixels (e.g., 28+32+32+48). As shown in Table 3, the high compression may allow the central display region 202-5 of a display with a resolution of 320x320 to be expanded laterally in one or both directions to display 260 media source pixels which are not subjected to image compression and/or scaling. In some implementations, the central display region 202-5 may be reduced to create an expanded non-linear area. In general, however, it may be desirable to configure the central display region 202-5 to be as large as possible.

In the embodiments illustrated by Table 3, the lateral display regions 202-1-4 to the left of the central region 202-5 may be symmetric with the lateral display regions 202-9-6 to the right of the central display region 202-5. In such embodiments, image compression and/or scaling may be applied by four symmetrically arranged hardware-based sample-rate converters. The first hardware-based sample-rate converter may be arranged to generate display pixels for lateral display regions 202-1 and 202-9 based on a 14:1 compression ratio. The second hardware-based sample-rate converter may be arranged to generate display pixels for lateral display regions 202-2 and 202-8 based on an 8:1 compression ratio. The third hardware-based sample-rate converter may be arranged to generate display pixels for lateral display regions 202-3 and 202-7 based on a 4:1 compression ratio. The fourth hardware-based sample-rate converter may be arranged to generate display pixels for display regions 204-1 and 206-9 based on a 3:1 compression ratio. The embodiments, however, are not limited in this context. For example, the embodiments may comprise software to perform sample-rate conversion.

Appendix: Meulen (Ex. 1007)

- Ex. 1007, Cols. 9-10

9
In various embodiments, image compression and/or scaling may be applied to the left lateral display regions 202-1-5 and the right lateral display regions 202-9-6 asymmetrically. In some embodiments, for example, the center of linearity and/or center of compression for the panoramic display 200 may be shifted to the left and/or to the right under user-control (e.g., navigation button 122) to provide a shifted or magnified effect. In one embodiment, for example, the panoramic display 200 may be shifted to the left such that the left lateral display regions 202-1-4 include fewer display pixels and subject the media source content to higher compression and/or scaling than the right lateral display regions 202-6-9. Similarly, the panoramic display 200 may be shifted to the right such that the right lateral display regions 202-6-9 include fewer display pixels and subject the media source content to higher compression and/or scaling and than the left lateral display regions 202-1-4.

Table 4 illustrates a left shifted implementation for a panoramic display 200 comprising display regions 202-1-9.

TABLE 4										
	Region									Total
	1	2	3	4	5	6	7	8	9	
Pixels/Region	2	4	8	16	230	32	16	8	4	320
Compression Ratio	14:1	8:1	4:1	3:1	none	3:2	2:1	4:1	7:1	
Source Pixels	28	32	32	48	230	48	32	32	28	480

In this left shifted implementation, the panoramic display 200 may have an aspect ratio of 1:1 with a resolution of 320x320, and the media source content may have an aspect ratio of 3:2 with a resolution of 480x320. The lateral display region 202-1 may comprise 2 pixels and subject the media source content to a 14:1 image compression ratio (e.g., 28 source pixels to 2 display pixels). The lateral display region 202-2 may comprise 4 pixels and subject the media source content to an 8:1 image compression ratio (e.g., 32 source pixels to 4 display pixels). The lateral display region 202-3 may comprise 8 pixels and subject the media source content to a 4:1 image compression ratio (e.g., 32 source pixels to 8 display pixels). The lateral display region 202-4 may comprise 16 pixels and subject the media source content to a 3:1 image compression ratio (e.g., 48 source pixels to 16 display pixels).

The lateral display region 202-6 may comprise 32 pixels and subject the media source content to a 3:2 image compression ratio (e.g., 48 source pixels to 32 display pixels). The lateral display region 202-7 may comprise 16 pixels and subject the media source content to a 2:1 image compression ratio (e.g., 32 source pixels to 16 display pixels). The lateral display region 202-8 may comprise 8 pixels and subject the media source content to a 4:1 image compression ratio (e.g., 32 source pixels to 8 display pixels). The lateral display region 202-9 may comprise 4 pixels and subject the media source content to a 7:1 image compression ratio (e.g., 28 source pixels to 4 display pixels).

As shown in Table 4, the central display region 202-5 may comprise 230 pixels which undergo no image compression (e.g., 230 media source pixels to 230 display pixels). In this left shifted implementation, the left lateral display regions 202-1-4 may be arranged to display 30 pixels (e.g., 2+4+8+16) by compressing and/or scaling 140 horizontal source pixels (e.g., 28+32+32+48). The right lateral display regions 202-6-9 may be arranged to display 60 pixels (e.g., 32+16+8+4) by compressing and/or scaling 140 horizontal source

10
pixels (e.g., 48+32+32+28). In this case, the higher compression and/or scaling may allow the central display region 202-5 of a display with a resolution of 320x320 to be expanded laterally in the direction of the shift to display additional media source pixels which are not subjected to image compression and/or scaling.

The embodiment illustrated by Table 4 may comprise, for example, sample-rate converter circuitry implemented by seven hardware-based sample-rate converters. One of the hardware-based sample-rate converters may be arranged to generate display pixels for both lateral display regions 202-3 and 202-8 based on a 4:1 compression ratio. The other six hardware-based sample-rate converters may be arranged to generate display pixels for corresponding lateral display regions 202-1, 202-2, 202-4, 202-7, 202-7 and 202-9. The embodiments, however, are not limited in this context. For example, the embodiment illustrated by Table 4 may comprise software to perform sample-rate conversion.

In some embodiments, image compression and/or scaling may be applied in a horizontal direction and/or in a vertical direction. When applied in both the horizontal and vertical direction, the image compression and/or scaling may allow expansion of the lateral display regions 202-1-4 and 202-6-9 to create a magnified effect or to simulate a video graphics array (VGA) having a resolution of 640x480 with a display having a resolution of 320x320.

FIGS. 4A-C illustrate one embodiment of compression factors for a plurality of display regions. In various embodiments, the compression factors may be implemented by the wireless device 100 of FIG. 1 or by the panoramic display 200 of FIG. 2. The embodiments are not limited in this context.

As shown in FIG. 4A, the image compression and/or scaling may be applied progressively, and the image compression ratios for left lateral display regions may be symmetric with the image compression ratios for right lateral display regions. As shown in FIG. 4B, the image compression and/or scaling may be shifted to the left such that left lateral display regions and subject media source content to higher compression and/or scaling than right lateral display regions. As shown in FIG. 4C, the image compression and/or scaling may be shifted to right such that right lateral display regions subject media source content to higher compression and/or scaling than left lateral display regions.

In this embodiment, a panoramic display may be implemented by employing significantly more display regions (e.g., >30) and hardware-based sample-rate converters to smooth the visual transitions from one display region to another. In some implementations, the panoramic display may employ software to perform the sample-rate conversion at the expense of more memory and processing power. When performed by software, the sample-rate conversion may be implemented in registers rather than random access memory (RAM) to minimize the number of accesses or taps and to reduce processing expense. The embodiments are not limited in this context.

FIG. 5 illustrates one embodiment of a computing system 500. In various embodiments, the computing system 500 may comprise or be implemented by the wireless device 100 of FIG. 1 or any other type of wireless device. The embodiments are not limited in this context.

The computing system 500 generally may comprise various physical or logical elements implemented as hardware, software, or any combination thereof, as desired for a given set of design parameters or performance constraints. In various embodiments, the physical or logical elements may be connected by one or more communications media. Communications media generally may comprise any medium capable

Appendix: Meulen (Ex. 1007)

• Ex. 1007, Cols. 11-12

11

of carrying information signals. For example, communication media may comprise wired communication media, wireless communication media, or a combination of both, as desired for a given implementation.

As shown, the computing system 500 may comprise a display 502. The display 502 may be implemented using any type of visual interface such as an LCD. As shown, the display 502 may comprise a plurality of display regions arranged to implement a panoramic display, as described above. In some embodiments, the display 510 may have an aspect ratio of 1:1 with a resolution of 320x320, 480x480, and so forth. In other embodiments, the display 502 may have an aspect ratio of 3:2 with a resolution of 480x320 or other aspect ratio and/or resolution. The embodiments are not limited in this context.

The computing system 500 may comprise a display controller 504. The display controller 504 may comprise, or be implemented as hardware, software, or any combination thereof. In some embodiments, for example, the display controller 504 may comprise one or more processors, controllers, encoder devices, decoder devices, CODEC devices, scaling devices, filters, converters, circuits, chips, logic devices, logic gates, switches, registers, semiconductor devices, transistors, or combination thereof. In some embodiments, for example, the display controller 504 may comprise software implemented by one or more applications, drivers, programs, modules, subroutines, instruction sets, instructions, computing codes, or combination thereof.

The display controller 504 may be arranged to performing various processing operations such as adding, subtracting, multiplying, dividing, sampling, compressing, converting, filtering, scaling, interpolating, and so forth. In various embodiments, the display controller 504 may comprise one or more sample-rate converters implemented by hardware and/or software. In some embodiments, for example, the display

controller 504 may comprise sample-rate converters implemented in hardware by programmable polyphase filters arranged to sample a line of source pixels and to generate a compressed and/or scaled line of display pixels based on a compression factor. In some embodiments, for example, the display controller 504 may comprise sample-rate conversion software to perform sample-rate conversion, such as polyphase filtering (e.g., MMX optimized polyphase filtering). The embodiments are not limited in this context.

As shown, the computing system 500 may comprise a frame buffer 506. In various embodiments, the frame buffer 506 may be arranged to store media source content. The media source content may comprise pixels derived from or associated with one or more static or video images. In various implementations, the frame buffer 508 may comprise one or more overlay planes and comprise pixels such as RGB pixels, YUV pixels, and/or YCC pixels. The embodiments are not limited in this context.

In various embodiments, the frame buffer 506 may be arranged to store media source content having an aspect ratio and/or a resolution that differs from the aspect ratio and/or the resolution of the display 502. In one embodiment, for example, the frame buffer 506 may be arranged to store media source content having an aspect ratio of 3:2 with a resolution of 480x320 to be displayed by a display 502 having an aspect ratio of 1:1 with a resolution of 320x320. In another embodiment, for example, the frame buffer 506 may be arranged to store media source content having an aspect ratio of 2:1 with a resolution of 640x320 to be displayed by a display 502 having an aspect ratio of 3:2 with a resolution of 480x320. The embodiments are not limited in this context.

In various embodiments, the display controller 504 may be arranged to apply image compression and/or scaling to the

12

media source content stored in the frame buffer 506. For example, the display controller 504 may be arranged to retrieve and process a line of source pixels from the display buffer 506 to generate a compressed and/or scaled line of display pixels. In some embodiments, the display controller 504 may be arranged to apply image compression and/or scaling in the horizontal direction and/or in the vertical direction.

In various implementations, the display controller 504 may be arranged to apply image compression and/or scaling progressively to the media source content based on the proximity of a display region to the edge of the display 502. For example, the display controller 504 may apply more image compression and/or scaling to the media source content that is to be displayed closer to edge of the display 502, than to the media source content that is to be displayed closer to the center of the display 502.

In some implementations, the display controller 504 may be arranged to apply image compression and/or scaling symmetrically to left and right lateral display regions. In some implementations, the display controller 504 may be arranged to apply more image compression and/or scaling to either the left or the right lateral display regions. In other implementations, the display controller 504 may be arranged to expand one or more display regions to create a magnified effect.

As shown, the frame buffer 506 may be implemented by a memory 508. In various embodiments, the memory 508 may comprise any machine-readable or computer-readable media capable of storing data, including both volatile and non-volatile memory. For example, memory may include read-only memory (ROM), random-access memory (RAM), dynamic RAM (DRAM), Double-Data-Rate DRAM (DDR-RAM), synchronous DRAM (SDRAM), static RAM (SRAM), programmable ROM (PROM), erasable programmable ROM

(EPROM), electrically erasable programmable ROM (EEPROM), flash memory (e.g., NOR or NAND flash memory), content addressable memory (CAM), polymer memory (e.g., ferroelectric polymer memory), phase-change memory (e.g., ovonic memory), ferroelectric memory, silicon-oxide-nitride-oxide-silicon (SONOS) memory, disk memory (e.g., floppy disk, hard drive, optical disk, magnetic disk), or card (e.g., magnetic card, optical card), or any other type of media suitable for storing information.

The computing system 500 may comprise a processor 510 such as a central processing unit (CPU). In various embodiments, the processor 510 may be implemented as a general purpose processor, a chip multiprocessor (CMP), a dedicated processor, an embedded processor, a digital signal processor (DSP), a network processor, a media processor, an input/output (I/O) processor, a media access control (MAC) processor, a radio baseband processor, a co-processor, a microprocessor such as a complex instruction set computer (CISC) microprocessor, a reduced instruction set computing (RISC) microprocessor, and/or a very long instruction word (VLIW) microprocessor, or other processing device. The processor 510 also may be implemented by a controller, a microcontroller, an application specific integrated circuit (ASIC), a field programmable gate array (FPGA), a programmable logic device (PLD), and so forth. In some embodiments, the display controller 504 may comprise software running on the processor 510.

In various embodiments, the processor 510 may be arranged to run an operating system (OS) and various mobile applications. Examples of an OS include, for example, a Microsoft® Windows OS, a Palm OS®, and any other proprietary or open source OS. Examples of mobile applications include, for example, a telephone application, a camera (e.g.,

Appendix: Meulen (Ex. 1007)

• Ex. 1007, Cols. 13-14

13

digital camera, video camera) application, a browser application, a multimedia player application, a gaming application, a messaging application (e.g., e-mail, short message, multimedia), a viewer application, and so forth.

In various embodiments, the processor 510 may be arranged to receive information, such as media source content, through a communications interface 512. The communications interface 512 may comprise any suitable hardware, software, or combination of hardware and software that is capable of coupling the computing system 512 to one or more networks and/or network devices. The communications interface 512 may be arranged to operate with any suitable technique for controlling information signals using a desired set of communications protocols, services or operating procedures. The communications interface 512 may include the appropriate physical connectors to connect with a corresponding communications medium.

In various embodiments, the communications interface 512 may comprise one or more interfaces such as, for example, a wireless communications interface, a wired communications interface, a network interface, a transmit interface, a receive interface, a media interface, a system interface, a component interface, a switching interface, a chip interface, a controller, and so forth. When implemented by a wireless device or within wireless system, for example, the computing system 500 may include a wireless interface comprising one or more antennas, transmitters, receivers, transceivers, amplifiers, filters, control logic, and so forth.

In various implementations, the described embodiments may communicate over wireless shared media in accordance with a number of wireless protocols. Examples of wireless protocols may include various wireless local area network (WLAN) protocols, including the Institute of Electrical and

Electronics Engineers (IEEE) 802.xx series of protocols, such as IEEE 802.11a/b/g/n, IEEE 802.16, IEEE 802.20, and so forth. Other examples of wireless protocols may include various wireless wide area network (WWAN) protocols, such as GSM cellular radiotelephone system protocols with GPRS, CDMA cellular radiotelephone communication systems with 1xRTT, EDGE systems, EV-DO systems, EV-DV systems, HSDPA systems, and so forth. Further examples of wireless protocols may include wireless personal area network (PAN) protocols, such as an Infrared protocol, a protocol from the Bluetooth Special Interest Group (SIG) series of protocols, including Bluetooth Specification versions v1.0, v1.1, v1.2, v2.0, v2.0 with Enhanced Data Rate (EDR), as well as one or more Bluetooth Profiles, and so forth. Yet another example of wireless protocols may include near-field communication techniques and protocols, such as electromagnetic induction (EMI) techniques. An example of EMI techniques may include passive or active radio-frequency identification (RFID) protocols and devices. Other suitable protocols may include Ultra Wide Band (UWB), Digital Office (DO), Digital Home, Trusted Platform Module (TPM), ZigBee, and so forth.

In various implementations, the described embodiments may comprise part of a cellular communication system. Examples of cellular communication systems may include CDMA cellular radiotelephone communication systems, GSM cellular radiotelephone systems, North American Digital Cellular (NADC) cellular radiotelephone systems, Time Division Multiple Access (TDMA) cellular radiotelephone systems, Extended-TDMA (E-TDMA) cellular radiotelephone systems, Narrowband Advanced Mobile Phone Service (NAMPS) cellular radiotelephone systems, third generation (3G) systems such as WCDMA, CDMA-2000, UMTS

14

cellular radiotelephone systems compliant with the Third-Generation Partnership Project (3GPP), and so forth.

Some embodiments may be implemented, for example, using a machine-readable medium or article which may store an instruction or a set of instructions that, if executed by a machine, may cause the machine to perform a method and/or operations in accordance with the embodiments. Such a machine may include, for example, any suitable processing platform, computing platform, computing device, processing device, computing system, processing system, computer, processor, or the like, and may be implemented using any suitable combination of hardware and/or software. The machine-readable medium or article may include, for example, any suitable type of memory unit, memory device, memory article, memory medium, storage device, storage article, storage medium and/or storage unit, for example, memory, removable or non-removable media, erasable or non-erasable media, writeable or re-writeable media, digital or analog media, hard disk, floppy disk, Compact Disk Read Only Memory (CD-ROM), Compact Disk Recordable (CD-R), Compact Disk Rewritable (CD-RW), optical disk, magnetic media, magneto-optical media, removable memory cards or disks, various types of Digital Versatile Disk (DVD), a tape, a cassette, or the like. The instructions may include any suitable type of code, such as source code, compiled code, interpreted code, executable code, static code, dynamic code, and the like. The instructions may be implemented using any suitable high-level, low-level, object-oriented, visual, compiled and/or interpreted programming language, such as C, C++, Java, BASIC, Perl, Matlab, Pascal, Visual BASIC, assembly language, machine code, and so forth.

Unless specifically stated otherwise, it may be appreciated that terms such as "processing," "computing," "calculating," "determining," or the like, refer to the action and/or processes of a computer or computing system, or similar electronic computing device, that manipulates and/or transforms data represented as physical quantities (e.g., electronic) within the computing system's registers and/or memories into other data similarly represented as physical quantities within the computing system's memories, registers or other such information storage, transmission or display devices.

Numerous specific details have been set forth herein to provide a thorough understanding of the embodiments. It will be understood by those skilled in the art, however, that the embodiments may be practiced without these specific details. In other instances, well-known operations, components and circuits have not been described in detail so as not to obscure the embodiments. It can be appreciated that the specific structural and functional details disclosed herein may be representative and do not necessarily limit the scope of the embodiments.

It is also worthy to note that any reference to "one embodiment" or "an embodiment" means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment. The appearances of the phrase "in one embodiment" in various places in the specification are not necessarily all referring to the same embodiment.

While certain features of the embodiments have been illustrated as described herein, many modifications, substitutions, changes and equivalents will now occur to those skilled in the art. It is therefore to be understood that the appended claims are intended to cover all such modifications and changes as fall within the true spirit of the embodiments.

The invention claimed is:
1. A wireless device, comprising:
a panoramic display having a first aspect ratio, said panoramic display comprising a plurality of display regions

Appendix: Meulen (Ex. 1007)

- Ex. 1007, Cols. 15-16

15

including a central display region and a plurality of lateral display regions.

said wireless device to receive media source content having a second aspect ratio that differs from said first aspect ratio, said wireless device to compress said media source content to be displayed in one or more of said lateral display regions based on proximity of said one or more lateral display regions to an edge of said display.

2. The wireless device of claim 1, further comprising a frame buffer to store media content having said second aspect ratio.

3. The wireless device of claim 2, wherein said frame buffer comprises one or more overlay planes.

4. The wireless device of claim 1, further comprising a display controller to compress media content having said second aspect ratio.

5. The wireless device of claim 4, said display controller to apply image compression progressively for left lateral display regions and right lateral display regions.

6. The wireless device of claim 4, said display controller to apply image compression symmetrically for left lateral display regions and right lateral display regions.

7. The wireless device of claim 4, said display controller to apply image compression asymmetrically for left lateral display regions and right lateral display regions.

8. The wireless device of claim 4, said display controller to apply compression in at least one of a horizontal direction a vertical direction.

9. The wireless device of claim 4, said display controller to expand one or more of said display regions.

10. The wireless device of claim 4, said display controller to apply a compression ratio to a line of media source pixels to generate a compressed line of display pixels.

11. The wireless device of claim 4, said display controller comprising a plurality of hardware-based sample-rate converters.

12. The wireless device of claim 4, said display controller comprising a sample-rate converter software application.

16

13. A system for a wireless device, comprising:
a display controller to compress media source content stored by a wireless device, said wireless comprising a panoramic display having a first aspect ratio, said panoramic display comprising a plurality of display regions including a central display region and a plurality of lateral display regions, said media source content having a second aspect ratio that differs from said first aspect ratio,

said display controller to compress media source content to be displayed in one or more of said lateral display regions based on proximity of said one or more lateral display regions to an edge of said display.

14. The system of claim 13, said display controller comprising a plurality of hardware-based sample-rate converters.

15. The system of claim 13, said display controller comprising a sample-rate converter software application.

16. The system of claim 13, further comprising a frame buffer to store media content having said second aspect ratio.

17. A method comprising:
retrieving media source content stored by a wireless device, said wireless comprising a panoramic display having a first aspect ratio, said panoramic display comprising a plurality of display regions including a central display region and a plurality of lateral display regions, said media source content having a second aspect ratio that differs from said first aspect ratio; and
compressing media source content to be displayed in one or more of said lateral display regions based on proximity of said one or more lateral display regions to an edge of said display.

18. The method of claim 17, further comprising receiving said media source content at said wireless device.

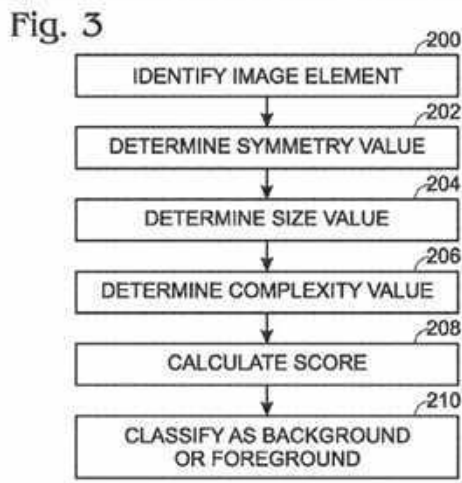
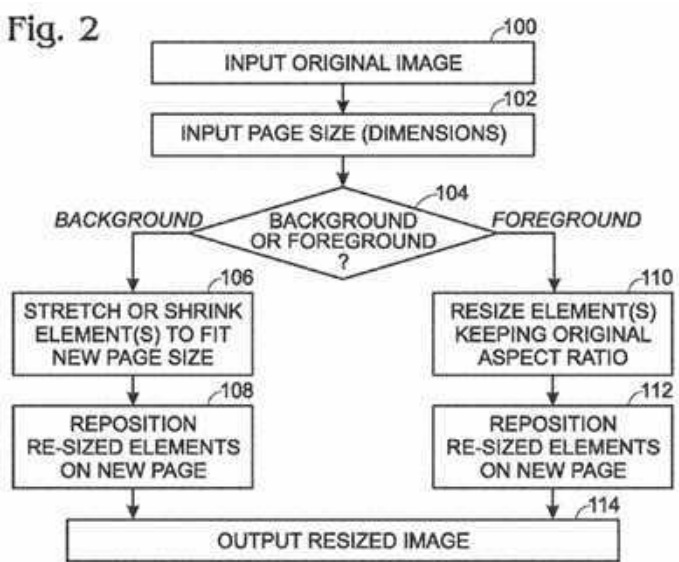
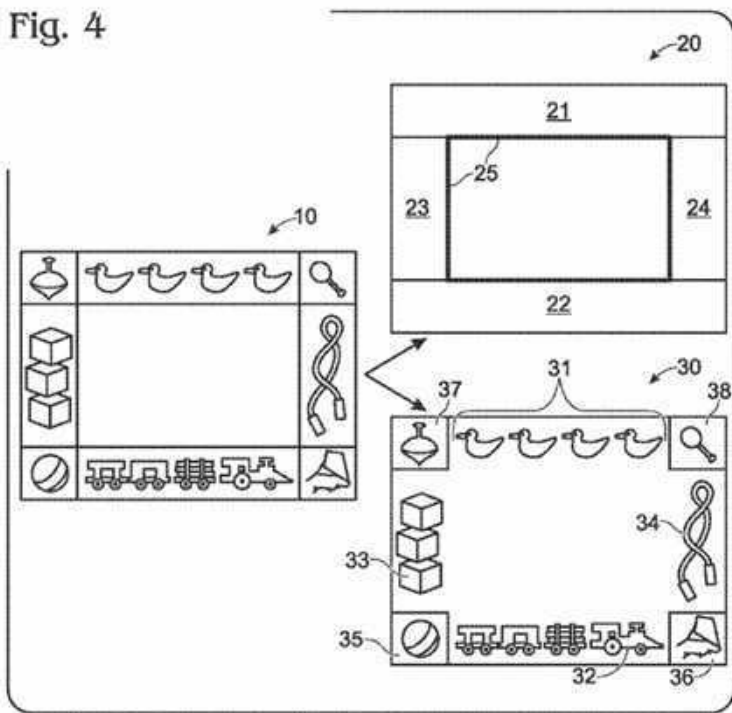
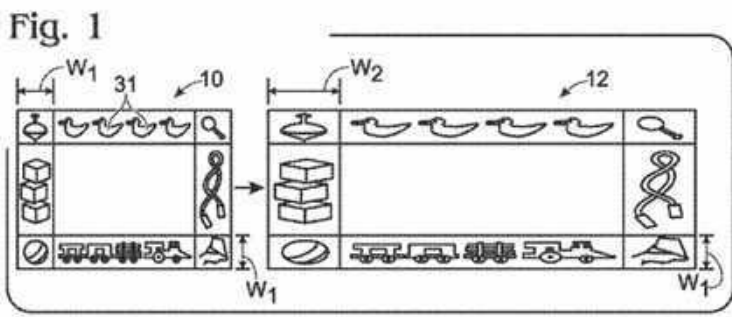
19. The method of claim 17, further comprising displaying compressed media source content in one or more of said lateral display regions.

* * * * *

Appendix: Chao (Ex. 1008)

Appendix: Chao (Ex. 1008)

- Ex. 1008, Figs. 1-4



Appendix: Chao (Ex. 1008)

- Ex. 1008, Figs. 5-7

Fig. 5

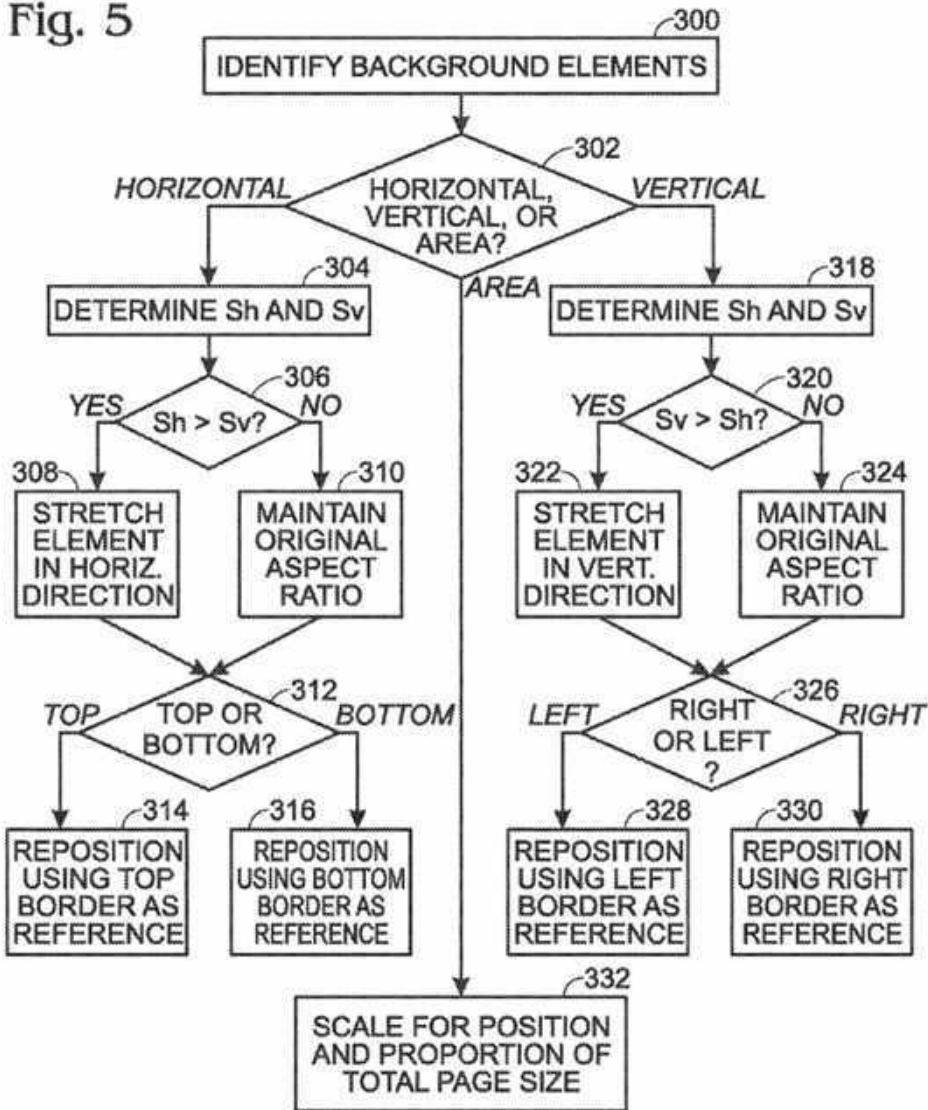


Fig. 6

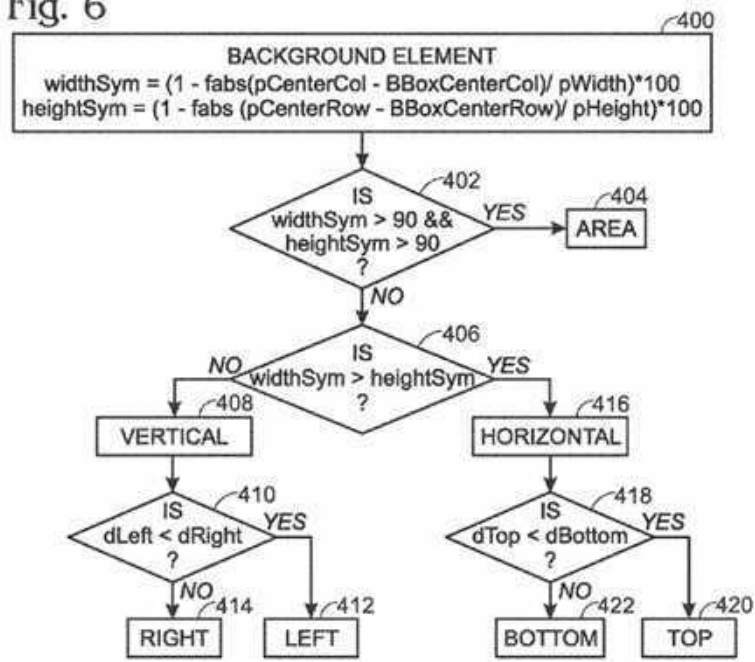
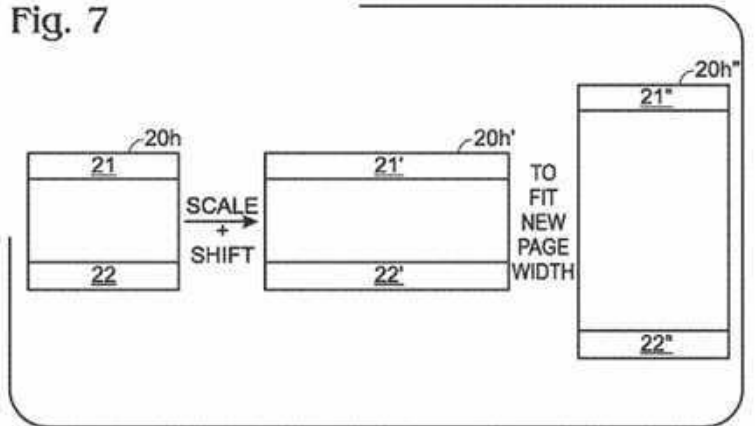
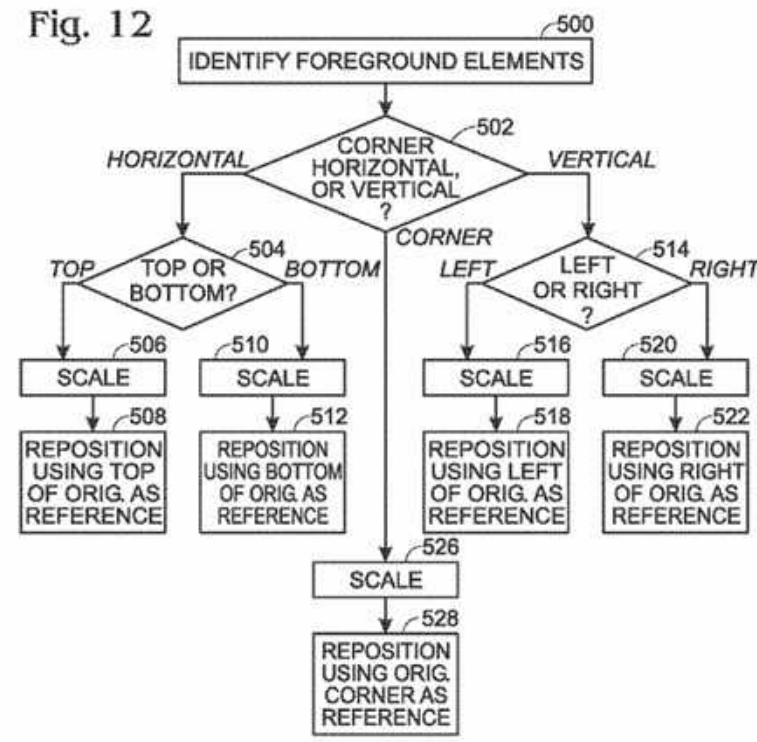
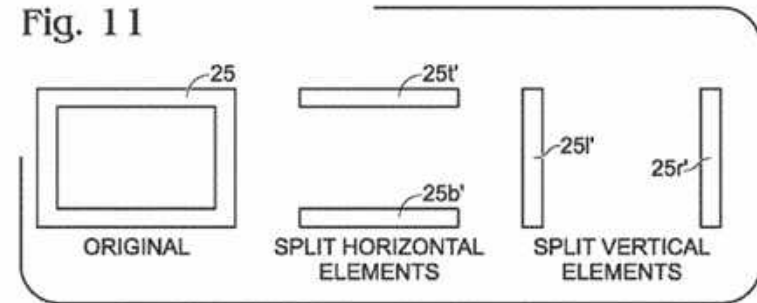
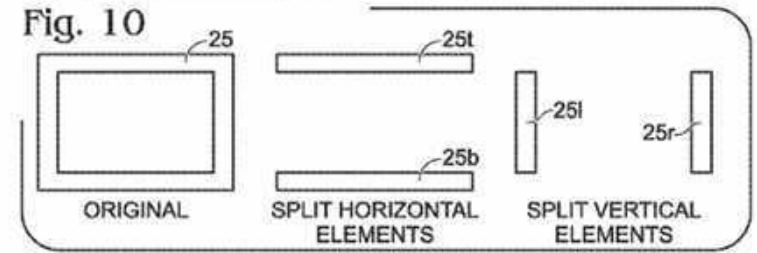
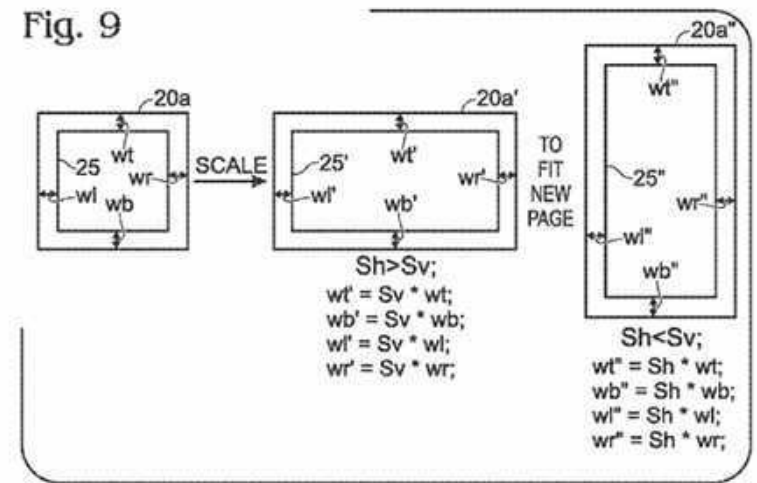
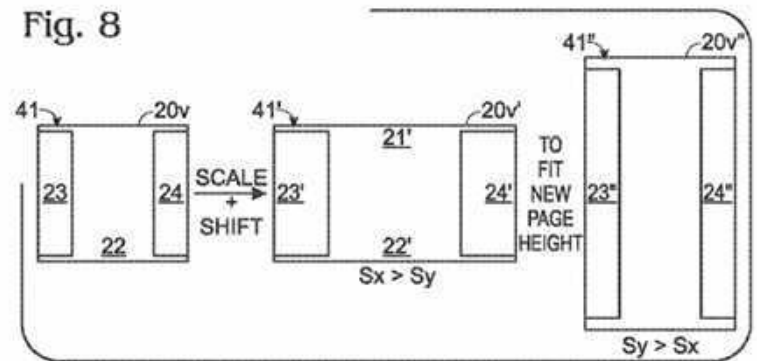


Fig. 7



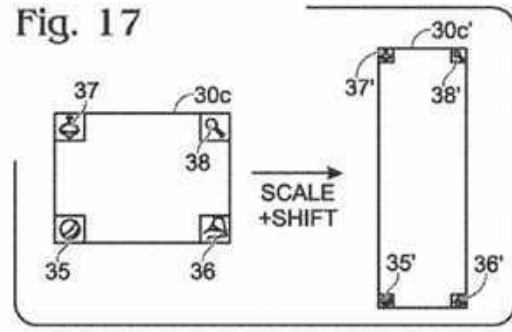
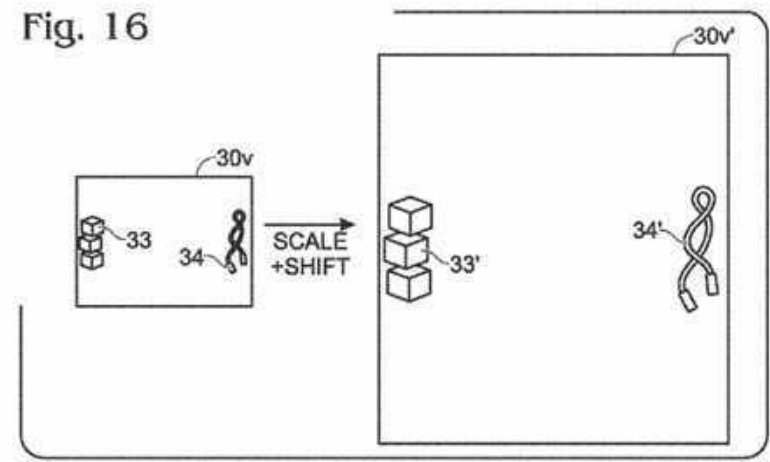
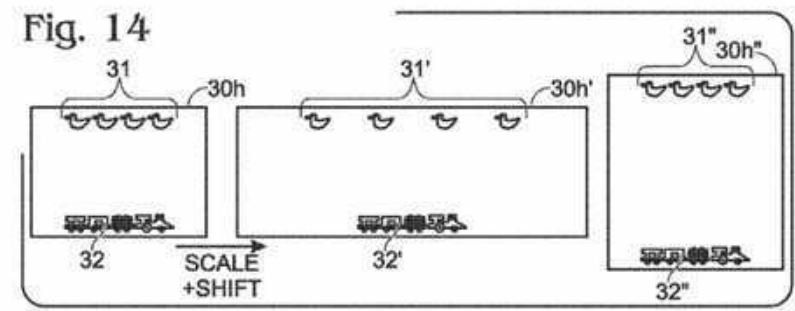
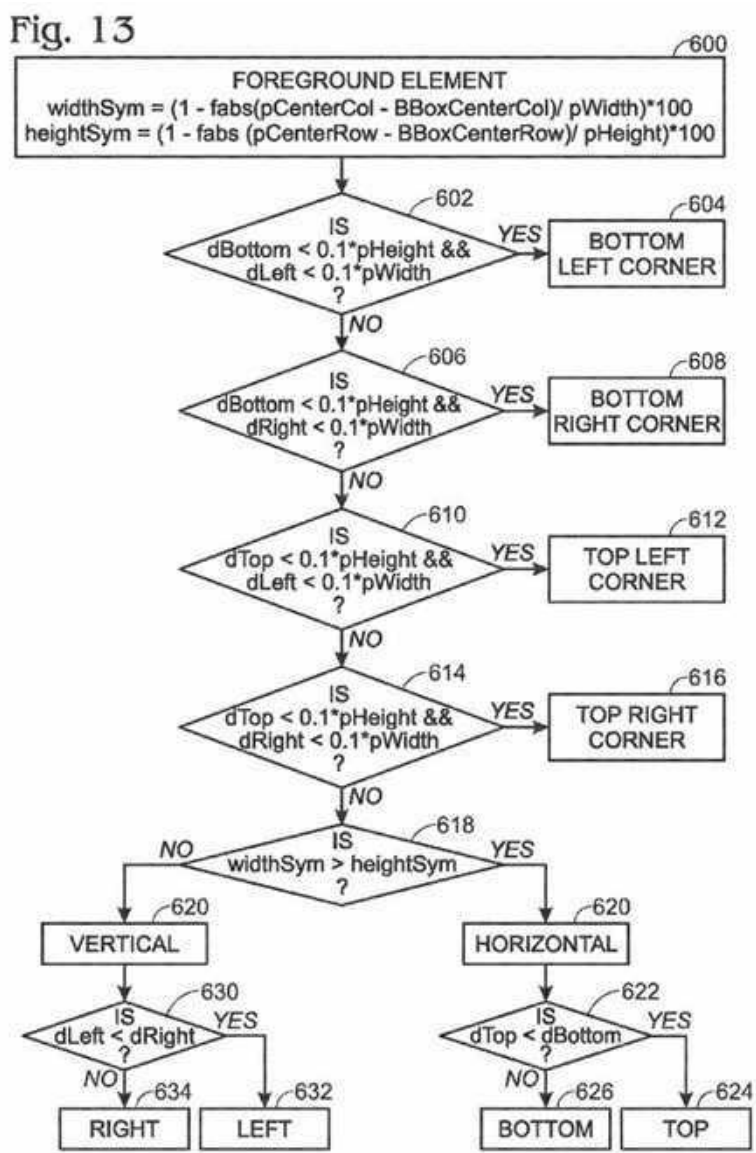
Appendix: Chao (Ex. 1008)

- Ex. 1008, Figs. 8-12



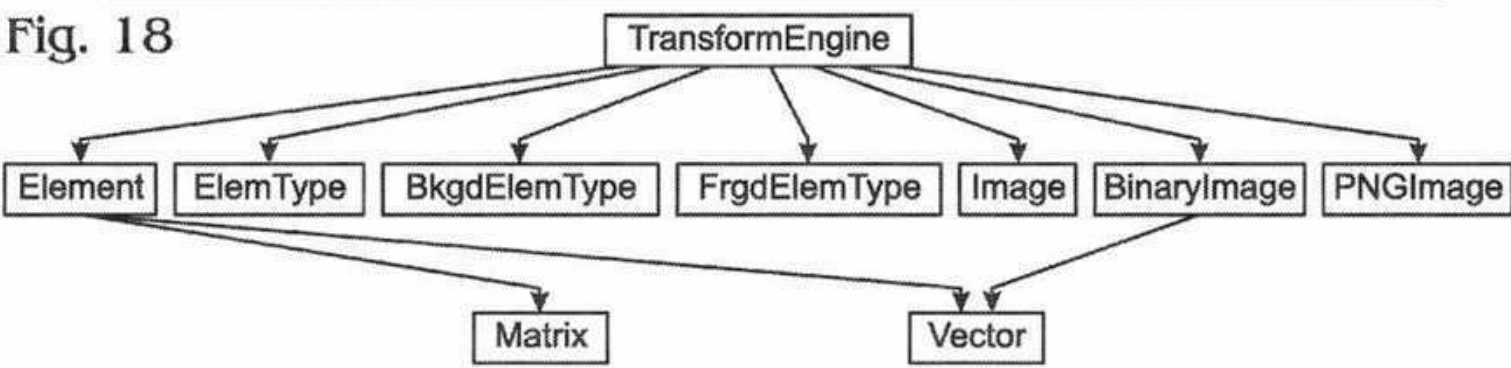
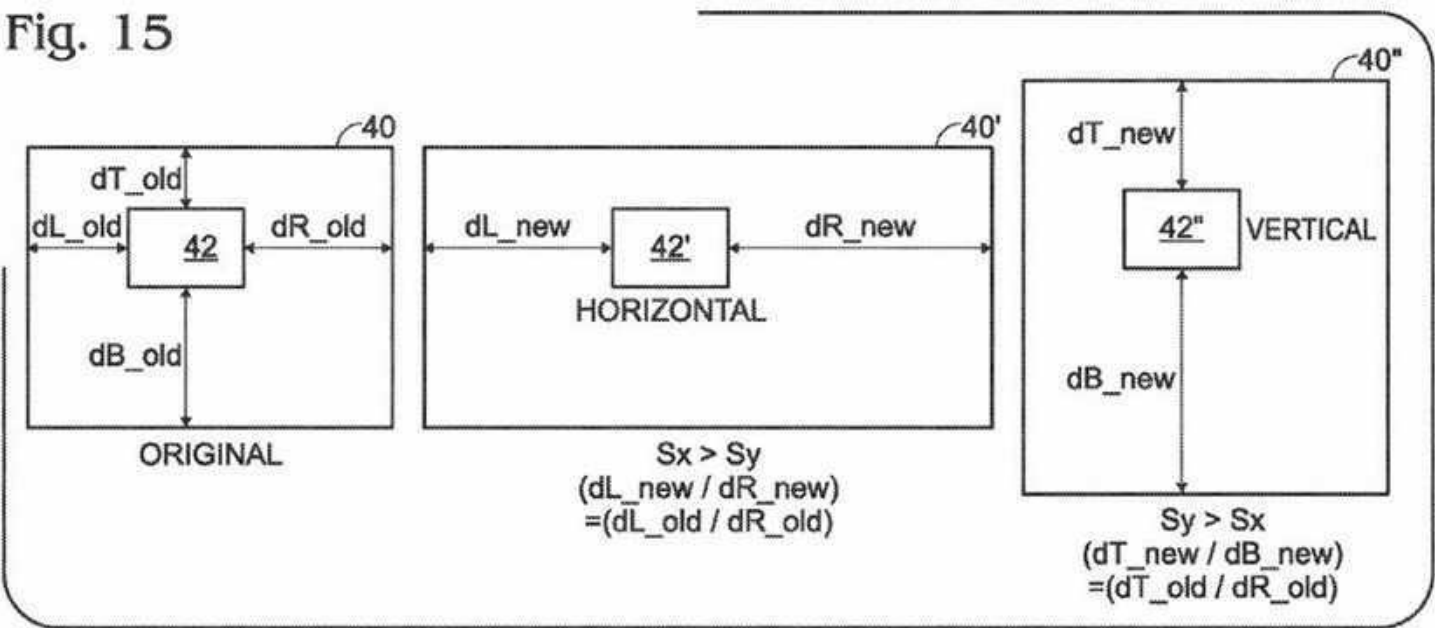
Appendix: Chao (Ex. 1008)

- Ex. 1008, Figs. 13-14, 16-17



Appendix: Chao (Ex. 1008)

- Ex. 1008, Figs. 15, 18



Appendix: Chao (Ex. 1008)

• Ex. 1008, ¶¶1-24

DIGITAL IMAGE AUTO-RESIZING

BACKGROUND

[0001] As digital hardware is more readily available, people are personalizing documents in a wide variety of ways. Frequently, documents include images that are designed to be scaled to specific page sizes. For example, people frequently like to incorporate professionally designed borders in their photos, newsletters, postcards, greeting cards and the like. As these images become more and more complex, they frequently incorporate elements that are designed for a specific page size. However, given the multitude of graphic design programs that allow people to resize their documents to any desired size, it would be extremely difficult, if not impossible for a graphic designer to create different versions of the same design for each possible page size.

[0002] Typically, current automatic image resizing programs simply scale an original image to fit a new page size without regard to the original design intent of the original image. For example, FIG. 1 depicts an original image 10 that has been resized from an original 4x5 template size to a 4x10 template size (at 12). As shown, the ducks that are used as upper border elements 31 have been stretched to fit the width of the image size, distorting the original proportions and overall look of the ducks. Moreover, depending upon the method used to stretch the image, the relative proportions of the side edge elements to the top and bottom edge elements may be altered, so that instead of having an image with the same size border all around the edge, the side edge may be thicker or thinner than the top or bottom edges (compare, e.g. w_1 with w_2). Furthermore, some of the image resizing programs simply crop the portions of the images that do not fit onto the new page size, after enlarging the original image, if necessary. Such cropping could clearly damage the design intent of an image including components such as that shown in FIG. 1.

[0003] Accordingly, methods and systems for automatically resizing an image while maintaining the original design intent of the image are needed.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] FIG. 1 illustrates a prior art method for automatically resizing a digital image.

[0005] FIG. 2 is a flowchart depicting a method for automatically resizing a digital image according to an embodiment of the present disclosure.

[0006] FIG. 3 is a flowchart depicting a method for determining whether an image element is a background or foreground element according to an embodiment of the present disclosure.

[0007] FIG. 4 depicts the various layers which may make up a digital image according to an embodiment of the present disclosure.

[0008] FIG. 5 is a flowchart depicting a method for re-sizing and repositioning a background element according to one embodiment of the present disclosure.

[0009] FIG. 6 is a flowchart depicting a method for sub-categorizing background elements according to an embodiment of the present disclosure.

[0010] FIG. 7 depicts the scaling of horizontal background elements according to an embodiment of the present disclosure.

[0011] FIG. 8 depicts the scaling of vertical background elements according to an embodiment of the present disclosure.

[0012] FIG. 9 depicts the scaling of area background elements according to an embodiment of the present disclosure.

[0013] FIG. 10 depicts a method for splitting an area element according to an embodiment of the present disclosure.

[0014] FIG. 11 depicts another method for splitting an area element according to another embodiment of the present disclosure.

[0015] FIG. 12 is a flowchart depicting a method for re-sizing and repositioning a foreground element according to one embodiment of the present disclosure.

[0016] FIG. 13 is a flowchart depicting a method for sub-categorizing foreground elements according to an embodiment of the present disclosure.

[0017] FIG. 14 depicts the scaling of foreground elements according to an embodiment of the present invention.

[0018] FIG. 15 depicts the calculations for scaling of foreground elements according to an embodiment of the present disclosure.

[0019] FIG. 16 depicts the scaling of vertical foreground elements according to an embodiment of the present disclosure.

[0020] FIG. 17 depicts the scaling of corner foreground elements according to an embodiment of the present disclosure.

[0021] FIG. 18 depicts a representation of SnapFish class hierarchy.

DETAILED DESCRIPTION

[0022] The present disclosure provides methods and systems for automatically resizing an image while maintaining the original design intent of the original image.

[0023] FIG. 2 provides a flowchart depicting a basic method for automatically resizing a digital image according to an embodiment of the present invention. At block 100 an original image is received. The original image may be, for example, a digital image designed to be a border for a photograph, postcard, newsletter, or the like, or may be any other digital image. The image may be in any suitable format including, for example, vector format, such as a .pdf, .svg, .vml, etc., or bitmap format such as .png, .tiff, .psd, etc. The image may be provided from any suitable data source including, for example, a folder on a computer or other hardware device, a digital device memory card, an on-line website, or other similar data source.

[0024] At block 102, a request to resize the original image to a new image size is received from the user. The new image size may be selected by the user from a list of options or may be input in free form by the user. The new image size may be bigger or smaller than the original image in one or more directions. Accordingly, the new image may or may not retain the aspect ratio of the original image. The request may be received as a direct command to resize the original image from the user, or may be received indirectly, for example, by the user requesting that the original image be printed at a different size, by the user resizing a window on the screen, by the user cropping a photograph or other image that will be inserted into or otherwise associated with the

Appendix: Chao (Ex. 1008)

• Ex. 1008, ¶¶24-48

original image, by the detection of a different size of paper in a printer, or by any other indication that the user wishes to resize the original image.

[0025] At block 104, foreground elements are separated from background elements. Any suitable method for distinguishing foreground and background elements may be used. For example, some images are designed in various layers wherein each layer is labeled as either foreground or background. Alternatively, an image may be made up of several elements which may be retrieved from one or more sources (clip art files, web-sites, etc.) and each element may include metadata indicating whether the element is a background or foreground element. As a further alternative, elements can be determined to be either foreground or background elements based on their position and other characteristics within the context of the original image. One implementation of a method for distinguishing between background and foreground elements is described below with reference to FIG. 3.

[0026] Typically, the image will include at least one of a background element or a foreground element. More typically, the image will include at least one background element or at least one foreground element. For ease of discussion, an image may be referred to as comprising multiple layers such as one or more background layers and one or more foreground layers. Moreover, one or more background elements may be referred to as making up a background layer and one or more foreground elements may be referred to as making up a foreground layer. However, it should be appreciated that such characterization is not intended to limit the disclosure to only those images which are provided in a multilayered format or to only those images having multiple layers wherein the designer of the image has divided and/or identified various image elements as "background layer elements" or "foreground layer elements."

[0027] FIG. 4 shows original image 10 broken down into constituent layers, background layer 20 and foreground layer 30. As shown, original image 10 includes multiple elements which can be divided into background layer elements 21-25 and foreground layer elements 31-38.

[0028] Returning to FIG. 2, if an element is determined to be a background element, the method proceeds to block 106, where the background element is re-sized, by stretching or shrinking, as necessary, to fit the new page size. It should be noted that, according to one embodiment, the re-sizing of the background element is performed without regard to the original aspect ratio of the element. One implementation of a method for resizing background elements is described below with reference to FIGS. 5-11.

[0029] Still referring to FIG. 2, at block 108, the resized background element is repositioned on the new page in order to maintain the design intent of the original image. For the purposes of the present disclosure the phrase "in order to maintain the design intent of the original image" is intended to mean that the resized image maintains the look, feel, and relative proportions of the original image while still fitting to the new page size. In other words, the re-sized image should look as if the depicted image was designed for the current page size. One implementation for repositioning background elements is described below with references to FIGS. 5-11.

[0030] Returning to block 104, if an element is determined to be a foreground element, the method proceeds to block 110, where the element is re-sized to fit the new page size, while maintaining the original aspect ratio of the element.

[0031] At block 112, the resized foreground element is repositioned on the new page in order to maintain the design intent of the original image. One implementation for resizing

and repositioning foreground elements is described below with references to FIGS. 12-17.

[0032] Still referring to FIG. 2, after all the background and foreground elements in the image have been resized and repositioned to create the new re-sized image, the re-sized image is output to the user at block 114. The image may be output by displaying the image on or as part of: a display screen such as a computer monitor, camera screen, printer screen, printed page, projected image, or the like.

[0033] Turning now to FIG. 3, a method for distinguishing between background and foreground elements is shown. Briefly, each element undergoes a variety of measurements which evaluate and identify the position and features of the element with respect to the original image. Accordingly the "page" referred to with respect to the steps of FIG. 3, is the page for which the original image was designed.

[0034] At block 200, an image element is identified. At block 202, a symmetry value for the image element is determined. According to one embodiment, the symmetry value may be determined as follows:

$$HSymmetry = \text{fabs}(pHCenter - hCenter) / pWidth;$$

$$VSymmetry = \text{fabs}(pVCenter - vCenter) / pHeight;$$

$$symmetry = (1 - \min(HSymmetry, VSymmetry)) * 100;$$

[0035] Where:

[0036] pHCenter is the page horizontal center;

[0037] pVCenter is the page vertical center;

[0038] hCenter is the horizontal center of the image element; and

[0039] vCenter is the vertical center of the image element.

[0040] Accordingly, the symmetry value for an image may range from >0 to 100. If the image element is placed in the horizontal or vertical center of the page, the symmetry value for the image will be 100.

[0041] At block 204, a size value for the image element is determined. According to one embodiment, the size value may be determined as follows:

$$widthRatio = \text{fabs}(pWidth - width) / pWidth;$$

$$heightRatio = \text{fabs}(pHeight - height) / pHeight;$$

$$size = (1 - \min(widthRatio, heightRatio)) * 100;$$

[0042] Where:

[0043] pWidth is the page width;

[0044] pHeight is the page height;

[0045] width is the image element width;

[0046] height is the image element height

[0047] Accordingly, the size value may range from >0 to 100. If one size of the bounding box of the image element is the same as one side of the page, the size value will be 100.

[0048] At block 206, a complexity value for the image element is determined. According to one embodiment, the complexity value may be determined as follows:

```
If(type=="Drawing")
{
    xCmplx=fabs(pCenter.x-HBoxCenter)/pWidth*100;
    yCmplx=fabs(pCenter.y-VBoxCenter)/pHeight*100;
    complexity1=min(xCmplx, yCmplx);
    if(complexity1>5)
```

Appendix: Chao (Ex. 1008)

- Ex. 1008, ¶¶48-61

-continued

```
{
    int numOfAnchorPoints=getAnchorPoints(element);
    complexity2=numOfAnchorPoints;
    complexity=complexity2;
}
else
    complexity=complexity1;
}
else
    complexity=0;
if complexity1>5
{
    int
    numOfAnchorPoints=getAnchorPoints((PDEPath)element);
    complexity2=numOfAnchorPoints;
    complexity=complexity2;
}
else
    complexity=complexity1;
}
```

[0049] Where, pcenter.x and pcenter.y are each the center of the 2D bounding polygon of the image element in the horizontal and vertical direction, respectively. HBoxCenter and VBoxCenter are each the center of the bounding box of the image element in horizontal and vertical direction, respectively. "Complexity1" is a complexity measure of the element, based on the symmetry of the element. "numOfAnchorPoints" is the number of anchor points in the drawing element.

[0050] Drawing elements are composed by lines, curves and rectangles with fill and stroke. A line has two anchor points—the start and the end points. One type of curve that is frequently used in computer graphics is a Bezier curve. A Bezier curve has 4 anchor points, the start point, the end point and two control points. A rectangle also has 4 anchor points, one for each corner of the rectangle. Accordingly, "complexity2" uses the number of anchor points along the drawing path to decide the complexity of the shape.

[0051] Accordingly, the complexity value may range from 0 to 100. For non-drawing elements, such as text, the complexity measure is set to be 100. For drawing elements, the complexity measure is based on the symmetry of the bounding polygon of the element and the number of anchor points in the drawing path. If the element is symmetrical, that is if the horizontal or vertical center of the bounding polygon is also the horizontal or vertical center of the bounding box, then the complexity is 0. Otherwise, the number of anchor points is included in the calculation.

[0052] Once the symmetry, size, and complexity values are determined for a given element, a score is calculated at 208 and the element is classified as either a background or foreground element at 210. In general, background elements tend to be symmetrically placed on the page (i.e. have a high symmetry value), their size is closer to the dimension of the page (i.e. they have a high size value) and they tend to be simpler (i.e. they have a low complexity value.)

[0053] Turning now to FIG. 5, a method for resizing and repositioning background elements is shown. At block 300, background elements are identified, such as by using one or more of the methods described above. If necessary, individual background elements may be extracted, for example, from a background layer. For example, background elements may be extracted by binarizing the alpha channel or lumi-

nance of a background layer to create a binary image. Connected components in the binary image can then be identified.

[0054] At block 302, each background element is classified as either a horizontal, vertical, or area element. Each element is then scaled based on its classification.

[0055] Turning briefly to FIG. 6, one implementation for classifying background elements is described. Generally, the method classifies and sub-classifies each element based on its relative position and proximity to the various borders on the original page.

[0056] According to the depicted embodiment, an element is classified as an area element if the bounding box of the element lies within a predefined area around the center of the image. At 400, widthsym and heightsym are calculated. As calculated, widthsym reflects the symmetry of an element in the horizontal direction. Correspondingly, heightsym reflects the symmetry of an element in the vertical direction.

[0057] At 402, the method determines if widthSym>90 and if heightsym>90. If both of these are true, the element is classified as an area element at 404. An example of an area element is shown at 25 in FIG. 4.

[0058] An element is classified as a vertical element if the element is more symmetrically placed in the vertical direction than the horizontal direction. Accordingly, at 406 if widthsym is not greater than heightsym the element is classified as a vertical element (408). The method then proceeds to block 410 to determine if the element is a left or right vertical element by determining the distance from the left edge of the element to the left border of the original page (dLeft) and the distance from the right edge of the element to the right border of the original page (dRight). If dLeft is less than dRight, the element is classified as a left vertical element (412). An example of a left vertical background element is shown at 23 in FIG. 4. If dLeft is not less than dRight, the element is classified as a right vertical element (414). An example of a right vertical background element is shown at 24 in FIG. 4.

[0059] Returning to block 406, an element is classified as a horizontal element if the element is more symmetrically placed in the horizontal direction than the vertical direction. Accordingly, if widthsym is greater than HeightSym, the element is identified as a horizontal element (416).

[0060] The method then proceeds to block 418 to determine if the element is a top or bottom element. Briefly, an element is sub-classified as a top element if it lies closer to the top border than the bottom border. Conversely, the element is sub-classified as a bottom element if it lies closer to the bottom border than the top border. Accordingly, the distance from the top edge of the element to the top border of the original page (dtop) and the distance from the bottom edge of the element to the bottom border of the original page (dbottom) is determined. If dtop is less than dbottom, the element is classified as a top horizontal element. An example of a top horizontal background element is shown at 21 in FIG. 4. If dtop is not less than dbottom, the element is classified as a bottom horizontal element. An example of a bottom horizontal background element is shown at 22 in FIG. 4.

[0061] Returning to FIG. 5, if an element is a horizontal element, the method proceeds to block 304, where the horizontal and vertical scaling factors are determined. The horizontal scaling factor (Sh) is the ratio of the width of the new page (i.e. the width of the page on which the re-sized

Appendix: Chao (Ex. 1008)

• Ex. 1008, ¶¶61-76

image will appear) to the width of the original page. The vertical scaling factor (S_v) is the ratio of the height of the new page to the height of the original page.

[0062] At 306, if Sh is greater than S_v , the element is allowed to stretch (or shrink) in the horizontal direction (308). In this case, the element is scaled by Sh' in horizontal direction and scaled by S_v in vertical direction. $Sh' = (Sh - S_v) * \text{origPW} / \text{origEW} + S_v$. Where origPW is the original page width and origEW is the original element's bounding box width. If the bounding box width of the element is the same as the page width of the original page, then $Sh' = Sh$. Sh' is calculated so that any gaps between element's edge and the re-sized page edge are scaled proportionally in S_v to any gaps in the original image.

[0063] For the purposes of the present disclosure, it should be understood that the terms "stretch" and "shrink" are intended to mean the resizing of an element without regard to the original aspect ratio of the image. Accordingly, a "stretched" element may be elongated in only one direction, i.e. may be resized such that it is wider than the original element in the horizontal direction, but the same size as the original in the vertical direction. If Sh is not greater than S_v , the element is not allowed to stretch (or shrink) in the horizontal direction and, if the element is scaled, the original aspect ratio is maintained, i.e., the element is scaled by S_v in both the horizontal and vertical directions(310).

[0064] At 312, the element is identified as either a top or bottom element. If the element is a top horizontal element, the element is repositioned and scaled, as necessary, using the top border as a reference (314). If the element is a bottom horizontal element, the element is repositioned and scaled, as necessary, using the bottom border as a reference (316).

[0065] Turning to FIG. 7, the scaling of horizontal background elements according to an embodiment of the present disclosure is depicted. As shown, original background layer 20h includes horizontal elements 21 and 22. Element 21 is a top horizontal element and element 22 is a bottom horizontal element.

[0066] New page 20h' is wider (i.e. larger in the x-direction) than original page 20h, but has the same height. Accordingly, elements 21' and 22' have been stretched in the x-direction (i.e. in width). Because original elements 21 and 22 extended the entire length of page 20h, re-sized elements 21' and 22' have been stretched to extend along the entire length of page 20h'.

[0067] New page 20h'' has increased height (i.e. is longer in the y-direction) than original page 20h, but has the same width. Accordingly, elements 21'' and 22'' have not been stretched. However, elements 21'' and 22'' have been repositioned to the top and bottom of the new page, respectively.

[0068] Returning to block 302 of FIG. 5, if the element is a vertical element, the method proceeds to block 318 to determine the horizontal and vertical scaling factors. At 320, if S_v is greater than Sh , the element is allowed to stretch in the vertical direction (322). The element is scaled by Sh in horizontal direction and S_v' in vertical direction. $S_v' = (S_v - Sh) * \text{origPH} / \text{origEH} + Sh$. Where origPH is the original Page height and origEH is the original element's bounding box height. If the bounding box height of the element is the same as the page height in the original page, then $S_v' = S_v$. Again S_v' is calculated so that any gaps between the element(s) and the page edges are scaled proportionally in Sh to any gaps in the original image. If S_v is not greater than Sh , the element is not allowed to stretch in the vertical direction and,

if the element is scaled, the original aspect ratio is maintained, i.e., the element is scaled by Sh in both horizontal and vertical direction(324).

[0069] At 326, the element is identified as either a left or right element. If the element is a left vertical element, the element is repositioned and scaled, as necessary, using the left border as a reference (328). If the element is a right vertical element, the element is repositioned and scaled, as necessary, using the right border as a reference (330).

[0070] Turning to FIG. 8, the scaling of vertical background elements according to an embodiment of the present disclosure is depicted. As shown, original background layer 20v includes vertical elements 23 and 24. Element 23 is a left vertical element and element 24 is a right vertical element.

[0071] New page 20v' is wider (i.e. longer in the x-direction) than original page 20h, and has increased slightly in height (i.e. is slightly longer in the y-direction), $Sh > S_v$. Accordingly, elements 23' and 24' have been scaled proportionally by S_v in both the vertical and horizontal directions (324) such that they cover the same relative amount of space on page 20v' as they did on page 20v. Put another way, because element 23 does not extend the entire length of the vertical edge, a gap 41 appears between the edge of page 20v' and the top end of element 23. Accordingly, the method of FIG. 5 is configured to stretch element 23' so that gap 41' is the same proportionate size as gap 41. Furthermore, elements 23' and 24' have been repositioned to the same relative space on page 20v' as the placement of elements 23 and 24 on page 20v. In other words, elements 23' and 24' have been repositioned to the left and right edges of page 20v', respectively.

[0072] Following the same logic, it can be seen that page 20v'' is taller than page 20v, but the same width. Accordingly elements 23'' and 24'' have been stretched in the vertical direction to take up the same relative space as elements 23 and 24 on page 20v.

[0073] Returning to block 302 in FIG. 5, if the element is determined to be an area element, the element is scaled to fit the new page so that it is placed at the same relative position while keeping the gaps between the element borders and corresponding page borders to be in the same proportion as the gaps in the original page. When $Sh > S_v$, the element's scaling factor in the horizontal and vertical directions are Sh' and S_v , respectively. When $Sh < S_v$, the element scaling in horizontal and vertical direction are Sh and S_v' respectively.

[0074] Turning to FIG. 9, the scaling of an area image is shown. Page 20a includes a single area element 25, which is centered within page 20a. New page 20a' is wider than page 20a, but the same height. Accordingly, area element 25' has been scaled to be wider than element 25, but is also centered within page 20a'.

[0075] Similarly, page 20a'' is taller than page 20a, but the same width. Accordingly area element 25'' has been scaled to be taller than element 25, but, again, is centered within page 20a''.

[0076] Alternatively, the individual elements of an area element could be split up and categorized as horizontal and vertical elements and then scaled and repositioned according to the methods described above. For example, rectangular area element 25 could be treated as two vertical elements (one left and one right) and two horizontal elements (one top and one bottom) and scaled and repositioned as such. According to some embodiments, this may lead to a more uniform scaling in the horizontal and vertical direction.

Appendix: Chao (Ex. 1008)

• Ex. 1008, ¶¶77-90

[0077] One method for splitting an area element is shown in FIG. 10. In the depicted method, two horizontal cuts split the area element into two horizontal elements 25*t* and 25*b* and two vertical elements 25*l* and 25*r*. A second method is depicted in FIG. 11. In this method, two vertical cuts split the area element into two horizontal elements 25'*t* and 25'*b* and two vertical elements 25'*l* and 25'*r*. The horizontal cut method of FIG. 10 may be selected, for example, when the horizontal scaling factor is greater than the vertical scaling factor. This is an effective method in this case because the split horizontal elements cover the entire breadth of the page in both the original and the resized image and the vertical split elements get scaled symmetrically, preserving the aspect ratio. Similarly, a vertical cut when the scaled image has a vertical stretch may be preferred according to some embodiments.

[0078] The splitting of elements may be accomplished by projecting the element pixel count in the background layer along the horizontal and vertical directions. The projected signals are differentiated and peaks in the differentiated signal are computed. The locations of the two largest peaks in the differentiated projection signal along the vertical direction determine the row coordinates of the inner corners of the hollow element. Similarly, the locations of the two largest peaks in the differentiated projection signal along the horizontal direction determine the column coordinates of the inner corners of the hollow element. Additional checks may be employed to ensure that the largest peaks correspond to the inner corners of the hollow element and not something else. For example, a threshold may be employed that requires that the magnitude of the peaks be at least 20 percent of the element dimension (element width for vertical direction peaks, element height for horizontal direction peaks). Alternatively or additionally, the system may require that the value of the derivative at the left peak be negative and the right peak be positive. Similar checks may be employed for top and bottom peaks.

[0079] Turning now to FIG. 12, a method for scaling and repositioning foreground elements according to one embodiment of the present disclosure is provided. At block 500, foreground elements are identified. As with the background elements, individual foreground elements may be extracted, for example, from a foreground layer. For example, foreground elements may be extracted by binarizing the alpha channel or luminance of a foreground layer to create a binary image. Connected components in the binary image can then be identified.

[0080] At block 502, the foreground elements are categorized as corner, horizontal, and vertical foreground elements. FIG. 13 depicts an implementation of a method for categorizing foreground elements according to an embodiment of the present disclosure. Briefly, the method classifies and sub-classifies each element based on its relative position and proximity to the various borders on the original page.

[0081] Turning to FIG. 13, an element is classified as a corner element if any pair of adjacent element borders is close enough to the nearest pair of adjacent image borders. Depending on the pair of borders that satisfy this criterion, the corner element is sub-classified as a bottom-left, bottom-right, top-left, or top-right corner element.

[0082] At block 600 widthSym and heightSym are calculated. As stated above, these measurements determine the symmetry of the element in the horizontal and vertical directions, respectively.

[0083] At block 602 the method determines whether the bottom and left borders of the element are within a given distance of the bottom left corner of the original image page. If it is, the element is classified as a bottom-left corner element (604). It will be understood that the specific distance requirement may depend upon the specific types of images that are being re-sized and that the depicted example is being shown as a non-limiting example. An example of a bottom-left corner element is shown at 35 in FIG. 4.

[0084] Still referring to FIG. 13, at block 606 the method determines whether the bottom and right borders of the element are within a given distance of the bottom right corner of the original image page. If it is, the element is classified as a bottom-right corner element (608). An example of a bottom-right corner element is shown at 36 in FIG. 4.

[0085] Still referring to FIG. 13, at block 610 the method determines whether the top and left borders of the element are within a given distance of the top left corner of the original image page. If it is, the element is classified as a top-left corner element (612). An example of a top-left corner element is shown at 37 in FIG. 4.

[0086] Still referring to FIG. 13, at block 614 the method determines whether the top and right borders of the element are within a given distance of the top right corner of the original image page. If it is, the element is classified as a top-right corner element (616). An example of a top-right corner element is shown at 38 in FIG. 4.

[0087] Still referring to FIG. 13, at block 618, the method determines if the element is a vertical or horizontal element by determining whether the element is place more symmetrically in the horizontal direction or the vertical direction. This can be calculated by determining if widthSym is greater than heightSym. If it is, the element is classified as a horizontal

element (620); if it isn't, the element is classified as a vertical element (628). Examples of horizontal foreground elements are shown at 31 and 32 on FIG. 4. Examples of vertical foreground elements are shown at 33 and 34 of FIG. 4.

[0088] Still referring to FIG. 13, if the element is a horizontal element, at block 622, the method further determines if the element is a top or bottom horizontal element by determining if dtop is smaller than dBottom. If it is, the element is sub-classified as a top element, if it isn't, the element is sub-classified as a bottom element. In FIG. 4, element 31 (which is itself made up of a repetitive pattern of individual elements) is a top horizontal element and element 32 is a bottom horizontal element.

[0089] Returning to block 630 of FIG. 13, the method further determines if the element is a right or left vertical element by determining if dLeft is less than dRight. If it is, the element is sub-classified as a left vertical element and, if it isn't, the element is sub-classified as a right vertical element. In FIG. 4, element 33 is a left vertical element and element 34 is a right vertical element.

[0090] Returning to FIG. 12, if the element is determined to be a horizontal element, the method moves to block 504 to determine if the element is a top or bottom horizontal element, for example, by using the methods described above. The elements are then scaled maintaining the original aspect ratio and repositioned using the top (block 508) or bottom (block 512) of the original image as a reference. According to one embodiment, the aspect ratio of all the elements in the foreground layer is preserved by selecting a

Appendix: Chao (Ex. 1008)

• Ex. 1008, ¶¶90-107

uniform scaling factor along both the horizontal and vertical directions. The uniform scaling factor may be, for example, the minimum of the page scaling factors in the horizontal and vertical directions.

[0091] For example, in FIG. 14, page 30*h*' is wider than page 30*h*, but the same height. Accordingly, in the depicted embodiment, the uniform scaling factor was selected to be 1, to reflect the scaling factor in the vertical direction. As shown, top horizontal element 31 on page 30*h* is made up of a plurality of duck images. On page 30*h*', which is wider than page 30*h*, the ducks are the same size as the ducks on page 30*h*, but have been spread out so that the ducks on page 30*h*' occupy the same relative space as the ducks on page 30*h*. Conversely, page 30*h*" is taller than page 30*h*, but the same width. Accordingly, the scaling factor is still 1, which reflects the horizontal scaling factor. Moreover, because the width of page 30*h*" is the same as that of page 30*h*, the ducks occupy the same position relative to each other that they had on page 30*h*. In this case, the horizontal elements 31" and 32" have simply been moved to the top and bottom edges of page 30*h*", respectively, while maintaining the same relative spacing between the elements and the edge of page 30*h*".

[0092] FIG. 15 depicts the calculations for determining how to position an element on a new page while maintaining the same relative spacing between an element and the edge of the page. As shown, original page 40 includes an element 42. For element 42, the distance from each edge of the element to each corresponding edge of the page can be calculated. For example, the distance from the right edge of the element to the right edge of the original page is dR_old , the distance from the left edge of the element to the left edge of the original page is dL_old , the distance from the top edge of the element to the top edge of the original page is dT_old , and the distance from the bottom edge of the element to the bottom edge of the original page is dB_old . Similarly,

dL_new , dR_new , dT_new , and dB_new can be calculated. In order for an element to maintain the same relative horizontal spacing, dL_new/dR_new must be equal dL_old/dR_old . Similarly, in order for an element to maintain the same relative vertical spacing dT_new/dB_new must be equal to dT_orig/dB_orig .

[0093] Returning to FIG. 12, if the element is determined to be a vertical element, the method moves to block 514 to determine if the element is a left or right vertical element. The elements are then scaled maintaining the original aspect ratio and repositioned using the left (block 518) or right (block 522) side of the original image as a reference. Again, the aspect ratio of all the elements in the foreground layer may be preserved by selecting a uniform scaling factor along both the horizontal and vertical directions.

[0094] For example, in FIG. 16, page 30*v*' is twice as wide as page 30*v*, and three times the height. Accordingly, elements 33' and 34' are scaled by a factor of 2x (the minimum of the vertical and horizontal scaling factors) while maintaining the original aspect ratio. Moreover, elements 33' and 34' are repositioned so that they maintain the same relative vertical and horizontal spacing as elements 33 and 34.

[0095] Returning to FIG. 12, if the element is determined to be a corner element, the method moves to block 524 to determine if the element is a bottom-left, bottom-right, top-left, or top-right element. At 526, each element is then scaled, for example, by using the uniform scaling factor described above, while maintaining the original aspect ratio. At 528, the elements are then repositioned relative to the

original corner while maintaining the same relative vertical and horizontal spacing as the original element.

[0096] For example, in FIG. 17, page 30*c*' is half as wide and twice as tall as page 30*c*. Accordingly, each of corner elements 35', 36', 37', and 38' are scaled by a factor of 1/2 (the smaller of the vertical and horizontal scaling factors), while still maintaining the original aspect ratio. Moreover, each element is repositioned in its respective corner while maintaining the same relative vertical and horizontal spacing as elements 35, 36, 37, and 38, respectively.

[0097] According to another embodiment, the present disclosure provides image elements which can be combined, for example, with their original layer order to form a final image that can be used, for example, as a border for a photograph, newsletter, postcard, greeting card, or the like. According to one embodiment, each image element includes metadata associated with the element that indicates the element's position on the page in the final image and instructions related to how the image should be scaled to accommodate various page sizes. The instructions could take the form of simply an indication as to whether the image is a background or foreground image, or any other information such as that disclosed herein as being helpful for scaling.

[0098] Furthermore, individual image elements may or may not be stored together. For example, an art design for a final image may be composed of individual elements that are stored in various places—e.g., on the user's computer, on the internet, and/or on a disk. As such the art design could be expressed in an XML or other suitable format that describes the content and type of elements (top right corner, left vertical, etc.) The content for each element could be a file name or a URL address.

[0099] The disclosure may be further understood by reference to the Example provided below.

EXAMPLE 1

Automatically Resizing SnapFish Photo Borders

[0100] Digital borders for use with the Snapfish photo-sharing software were resized as described below. The code takes as input a set of PNG images that are rendered from the individual layers. The input layers and their attributes are specified by an input text file that looks like:

[0101] Background bk0 bk1

[0102] Foreground fg0 fg1

[0103] The text file states that bk0.png, bk1.png are the background layers and fg0.png, fg1.png are the foreground layers of the multilayer image. Each layer is processed separately using a set of rules to render a scaled version of that layer.

[0104] The application uses a hierarchy of classes to accomplish its goals. TransformEngine (TransformEngine.h) is at the top of the hierarchy and uses services provided by the other classes. It is also the main interface class. The main() function (SnapFish.cpp) instantiates an object named autoImageRescaler of type TransformEngine and calls all the public member functions of this class. The order of function calls roughly follows the order of the various steps of the scaling algorithm.

[0105] 1. autoImageRescaler.LoadInputPNGImage()

[0106] 2. autoImageRescaler.BinarizeAlphaChannel()

[0107] 3. autoImageRescaler.LabelConnectedComponents()

Appendix: Chao (Ex. 1008)

• Ex. 1008, ¶¶108-35

[0108] 4. autoImageRescaler.WriteLabelsPNGImage()
[0109] 5. autoImageRescaler.ComputeElementProperties()
[0110] 6. autoImageRescaler.ClassifyElements()
[0111] 7. autoImageRescaler.ComputeElementTransformationMatrices()
[0112] 8. autoImageRescaler.TransformElements()
[0113] 9. autoImageRescaler.WriteOutputPNGImage()
[0114] This set of calls writes scaled versions of the various layers to the disk. The scaled versions are named by appending “_scaled” to the filename. For example, the scaled version of bk0.png will be written into bk0_scaled.png. Similarly connected component labeling result with a bounding box around each connected region is written to a PNG file with suffix “_labeled” added to the original filename.
[0115] The above functions in turn use public member functions of other classes in the hierarchy. A listing of these other classes with a rough grouping based on the services provided by them is given below:
[0116] 1. Element (TransformEngine.h)
[0117] 2. ElemType (TransformEngine.h)
[0118] 3. BkgdElemType (TransformEngine.h)
[0119] 4. FrgdElemType (TransformEngine.h)
[0120] Compute and store attributes of the various elements (connected regions) output by the connected components algorithm.
[0121] 5. Image (Image.h)
[0122] 6. BinaryImage (BinaryImage)

[0123] These encapsulate data elements to hold image data and functions to do image manipulation. Image is a template class and BinaryImage is derived as a boolean version of the generic Image class. The connected component labeling algorithm is part of the BinaryImage class implemented as a LabelRegions() public member function.

[0124] 7. PNGImage (PNGImage.h)

[0125] This is a wrapper class for the libpng (<http://www.libpng.org>) library and provides data elements and functions to hold and manipulate PNG images. It instantiates objects of type Image to handle the 4 different channels (R, G, B and Alpha) of the PNG image. Currently the interface to libpng is only for reading and writing PNG images through the wrapper functions LoadPNGFromFile() and WritePNGToFile().

[0126] 8. Matrix (Matrix.h)

[0127] 9. Vector (Vector.h)

[0128] These two classes facilitate matrix and vector operations.

[0129] In addition to the above classes, the code in the following files does the image scaling with bilinear interpolation when scaling up and low pass filtering followed by sub-sampling when scaling down.

[0130] 10. scale.h and scale.cpp

[0131] 11. scale16.h and scale16.cpp

[0132] 12. ScaleImage.h and ScaleImage.cpp

[0133] A rough representation of SnapFish class hierarchy is shown in FIG. 18. An arrow indicates that the class at the tail of the arrow is using the services of the class at the head of the arrow.

Classes and Methods

[0134] It should be noted that most of the functions in the application return an error code that is an integer value. It could be one of PG_ERROR or PG_NO_ERROR depending on whether the function has successfully completed its task. These error codes are defined in the stdafx.h file. In order to make error checking manageable without writing a lot of code a macro called ERE is also defined which can be used as ERE(function_name()). This macro returns from the function where it is residing with the error code PG_ERROR if PG_ERROR is returned by the call to function_name(). Otherwise it has no effect. The following is a brief description of the various classes and the most prominent member functions in the major classes.

1. TransformEngine

[0135]

```
int LoadInputPNGImage(const char* fileName);
    Load input PNG image from the specified filename.
int BinarizeAlphaChannel(const BYTE& threshold);
    Binarize the alpha channel using the specified threshold.
int LabelConnectedComponents( );
    Label the connected components in the binary alpha channel.
int ComputeElementProperties( );
    Compute element properties.
int ClassifyElements(const int& newPageNRows,
                    const int& newPageNCols,
                    const BOOL& frgdLayer = FALSE);
    Classify elements in order to compute appropriate element
    specific transformation matrix. The default assumption is that
    the elements belong to a background layer.
int ComputeElementTransformationMatrices( );
    Compute element specific transformation matrices.
int TransformElements( );
```

```
    Transform elements and write to output image.
int WriteLabelsPNGImage(const char* fileName);
    Write output labels PNG image to the specified filename.
int WriteOutputPNGImage(const char* fileName);
    Write output PNG image to the specified filename.
int InitializeOutputPNGImage(const BYTE& R = 0,
                            const BYTE& G = 0,
                            const BYTE& B = 0,
                            const BYTE& alpha = 0);
    Initialize output PNG image channels to the specified values.
    Default = 0 for all the channels.
int AddPNGFileToComposite(const char* fileName);
    Overlay the PNG image from input file on the composite image
    stored in m_PNG_outputPNGImage obeying the following rules:
    1. Ignore all pixels at which the alpha channel value is 0.
    2. Overwrite any previously stored RGB values at the pixels
    with alpha channel value not equal to 0.
int SplitHollowRectAreaElements( );
    Split hollow rectangular area elements into sub-elements.
int
ComputeSplitElementProperties(const Vector<int>& topPeakIndexVector,
                            const Vector<int>& bottomPeakIndexVector,
                            const Vector<int>& leftPeakIndexVector,
                            const Vector<int>& rightPeakIndexVector,
                            const Vector<long>& regionLabelsVector);
    Compute split element properties. The input is a set of 4
    vectors. Each of these vectors is used to specify the coordinates
    of the 4 corners for splitting the elements in the background
    layer.
int ClassifySplitElements( );
    Classify split elements in order to compute appropriate element
    specific transformation matrix.
int ComputeSplitElementTransformationMatrices( );
    Compute split element specific transformation matrices.
int PGLib::TransformEngine::TransformSplitElements( );
    Transform split elements and write to output image.
```

Appendix: Chao (Ex. 1008)

• Ex. 1008, ¶¶136-39

2. BinaryImage

[0136]

```
int CountTruePixels( );
    Count number of (BOOL) 1 pixels.
long LabelConnectedComponents(Image<Long>& labeledImage);
    Connected component labeling.
long LabelRegions(Region regions_list[],
    Image<long>& labeledImage,
    long& totalNumConnectedRegions,
    const int& neighborhoodSize = 1,
    const BOOL& useFourConnectedness = FALSE,
    const BOOL& mergeRegionLabels = FALSE);
    Region coloring algorithm with the following features:
    1. Option of using four or eight connectedness.
    2. Variable neighborhood size for defining connectivity.
```

3. Image

[0137]

```
Pixel<T>*& GetBuffer( ) const;
    Access the pixel buffer associated with "this" image.
int NumRows( ) const;
    Access the number of rows of "this" image.
int NumCols( ) const;
    Access the number of columns of "this" image.
int CopyFromBuffer(const T* inputBuf,
    const int& nRows,
    const int& nCols);
    Copy the input buffer into "this" image.
int CopyToBuffer(T* targetBuf);
    Copy "this" image into a buffer (buffer must be previously
    allocated).
```

```
int CopyToByteImage(Image<BYTE>& targetByteImage,
    const int& ROIStartRow,
    const int& ROIStartCol,
    const int& ROIHeight,
    const int& ROIWidth,
    const BYTE& bgVal = 0);
    Copy a region of interest (ROI) from "this" image into a byte
    image. The scaling of pixel values is determined by computing the
    maximum and minimum of the pixel values within the ROI. The
    region outside the ROI is filled with bgVal.
int AllocateImage(const int& nRows, const int& nCols);
    Generic allocation routine.
int DeAllocateImage( );
    Generic de-allocation routine.
int InitializeImage(const T& defaultValue);
    Generic initialization routine.
int CreateBinaryImage(const Image<BYTE>& inputImage,
    const BYTE& threshold,
    const BOOL& invert = FALSE);
    Convert an input BYTE image into a binary image by thresholding.
    Store the thresholded image into "this" image. By default (invert =
    FALSE), values below the threshold are set to 0, others are set
    to 1. If invert == TRUE, values below the threshold are set to 1,
    others are set to 0.
```

4. Matrix

[0138]

```
int InitializeMatrix(const T& defaultValue);
    Generic initialization routine.
T& MatrixElement(const int& rowIndex, const int& colIndex) const;
```

-continued

```
    Access an element of "this" Matrix.
int CopyFromMatrix(const Matrix<T>& inputMatrix);
    Copy the contents of the inputMatrix into "this" Matrix.
int CreateIdentityMatrix(const int& size);
    Make "this" Matrix an identity matrix of the specified size.
int Create2DAffineMatrix(const T& xScale,
    const T& yShear,
    const T& xShear,
    const T& yScale,
    const T& xTranslate,
    const T& yTranslate);
    Make "this" Matrix a 2D affine transformation matrix.
int MultiplyMatrixAndVector(Vector<T>& outputVector,
    const Vector<T>& inputVector);
    Post-multiply "this" matrix with a column vector.
int MultiplyVectorAndMatrix(Vector<T>& outputVector,
    const Vector<T>& inputVector);
    Pre-multiply "this" matrix with a row vector.
```

5. Vector

[0139]

```
int NumElements( ) const;
    Access the number of elements of "this" Vector.
int AllocateVector(const int& nElements);
    Generic allocation routine.
int DeAllocateVector( );
    Generic deallocation routine.
int InitializeVector(const T& defaultValue);
    Generic initialization routine.
T& VectorElement(const int& elementIndex) const;
    Access an element of "this" Vector.
```

```
double DotProduct(Vector<T>& inputVector);
    Dot product of "this" Vector with another.
int SortVectorElements(const int& startIndex = 0,
    const int& endIndex = m_i_NElements);
    Sort vector elements within an index range.
int WriteColVector(const char* fileName);
    Write vector as a column vector to an output file. If the output
    filename specifier string is NULL, then write to STDOUT.
int WriteRowVector(const char* fileName);
    Write vector as a row vector to an output file. If the output
    filename specifier string is NULL, then write to STDOUT.
int ComputeFirstOrdFwdDiffVector<T>(Vector<T>& outputVector);
    Compute first order forward difference approximation to
    derivative, "this" Vector forms the input. The result is put into
    outputVector. It is allocated or resized to be the same size as
    "this" Vector. The first element of the outputVector is always set
    to 0 since the number of elements in the first order difference
    is always one less than in the input vector.
int Absolute(Vector<T>& outputVector);
    Compute the absolute values of "this" Vector elements. Store the
    result in outputVector. It is allocated or resized to be the same
    size as "this" Vector.
int Maximum(T& maxValue,
    int& maxIndex);
    Compute the maximum among all the elements of "this" Vector. The
    maximum value is returned in maxValue and its index is returned
    in maxIndex. In the case of repeated maximum values, the index of
    the first occurrence in the Vector is returned.
int Minimum(T& minValue,
    int& minIndex);
    Compute the minimum among all the elements of "this" Vector. The
    minimum value is returned in minValue and its index is returned
    in minIndex. In the case of repeated minimum values, the index of
    the first occurrence in the Vector is returned.
```

Appendix: Chao (Ex. 1008)

- Ex. 1008, ¶¶140-42, cl. 1

6. PNGImage

[0140]

```
int NumRows( ) const;
    Access the number of rows of "this" image.
int NumCols( ) const;
    Access the number of columns of "this" image.
int AllocatePNGImage(const int& nRows, const int& nCols);
    Generic allocation routine.
int DeAllocatePNGImage( );
    Generic de-allocation routine.
Image<T>& GetRedChannelImage( );
    Access the red channel image.
Image<T>& GetGreenChannelImage( );
    Access the green channel image.
Image<T>& GetBlueChannelImage( );
    Access the blue channel image.
Image<T>& GetAlphaChannelImage( );
    Access the alpha channel image.
    int CopySubRowToRGBBuffer(T* targetBuf,
        const int& rowIndex,
        const int& startColIndex,
        const int& endColIndex);
    Copy RGB data from a subrow of "this" PNG image into a RGB
    buffer.
int CopySubRowToAlphaBuffer(T* targetBuf,
    const int& rowIndex,
    const int& startColIndex,
    const int& endColIndex);
    Copy alpha channel data from a subrow of "this" PNG image into a
    buffer.
int CopySubRowToRGBBuffer(T* targetBuf,
    const int& rowIndex,
    const int& startColIndex,
    const int& endColIndex,
    const Region* regionsList,
    const Image<long>& labelsImage,
    const long& matchLabel,
    const BOOL& serialLabels);
```

Overloaded version of the generic CopySubRowToRGBBuffer() function. Copies pixels whose labels in the labelsImage match the matchLabel value following the conventions below:
If serialLabels is TRUE, pixels whose labels equal matchLabel are copied. If serialLabels is FALSE, pixels whose root label in the regionsList data structure equals matchLabel are copied.

```
int CopySubRowToAlphaBuffer(T* targetBuf,
    const int& rowIndex,
    const int& startColIndex,
    const int& endColIndex,
    const Region* regionsList,
    const Image<long>& labelsImage,
    const long& matchLabel,
    const BOOL& serialLabels);
```

Overloaded version of the generic CopySubRowToAlphaBuffer() function. Copies pixels whose labels in the labelsImage match the matchLabel value following the conventions below:
If serialLabels is TRUE, pixels whose labels equal matchLabel are copied. If serialLabels is FALSE, pixels whose root label in the regionsList data structure equals matchLabel are copied.

```
int CopyRGBBufferToSubRow(const T* inputBuf,
    const int& rowIndex,
    const int& startColIndex,
    const int& endColIndex,
    const T& alpha = 255);
    Copy data from a RGB buffer into a subrow of "this" PNG image.
    The alpha channel pixels are all set to the specified value
    (default = 255).
```

```
int CopyRGBBufferToSubRow(const T* inputBuf,
    Image<long>& labelsImage,
    const int& rowIndex,
    const int& startColIndex,
    const int& endColIndex,
    const long& regionLabel);
```

-continued

Copy data from a RGB buffer into a subrow of "this" PNG image. The alpha channel pixels are ignored. Only those pixels in the subrow are considered that have not been previously assigned a RGB value. This is determined by checking for the existence of a non-zero label at the corresponding pixel location in the labelsImage.

If RGB value is written at a pixel, then the corresponding pixel in the labelsImage is assigned the value of regionLabel.

```
int CopyAlphaBufferToSubRow(const T* inputBuf,
    const int& rowIndex,
    const int& startColIndex,
    const int& endColIndex);
```

Copy data from an alpha channel buffer into a subrow of "this" PNG image.

```
int CopyNZAlphaBufferToSubRow(const T* inputBuf,
    const int& rowIndex,
    const int& startColIndex,
    const int& endColIndex);
```

Copy data from an alpha channel buffer into a subrow of "this" PNG image. Only the non-zero values from the alpha buffer are copied.

```
int CreatePNGImage(const Image<T>& singleChannelImage,
    const T& alpha = 255,
    const BOOL& scaleRange = FALSE,
    const T& maxScaledOutputValue = 255);
```

Create a gray level PNG image from a single channel image. The alpha channel pixels are all set to the specified value (default = 255).

```
int LoadPNGFromFile(const char* fileName);
    Load PNG image from file.
```

```
int WritePNGToFile(const char* fileName);
    Write PNG image to file.
```

```
int DrawRegionProperties(const Region regionsList[],
    const long& maxRegionLabel,
    const T& boundingBoxR,
    const T& boundingBoxG,
    const T& boundingBoxB,
    const T& centroidR,
    const T& centroidG,
    const T& centroidB);
```

[0141] using information about labeled regions from a regionsList array, draw bounding box and center of mass location of each region.

[0142] While the invention has been described with reference to the exemplary embodiments thereof, those skilled in the art will be able to make various modifications to the described embodiments without departing from the true spirit and scope of the disclosure. Accordingly, the terms and descriptions used herein are set forth by way of illustration only and are not meant as limitations.

What is claimed is:

1. A method for auto-resizing an original digital image from an original image to a resized image, where the original image includes at least one of a foreground element or a background element, the method comprising:

- separating the foreground elements from the background elements;
- determining an original aspect ratio of the foreground elements;
- creating resized foreground elements by scaling the foreground elements while maintaining the original aspect ratio;
- creating resized background elements by stretching or shrinking the background elements to fit the dimensions of the resized image; and
- repositioning the resized foreground and background elements on the resized image while maintaining the design intention of the original image.

Appendix: Chao (Ex. 1008)

• Ex. 1008, cls. 2-24

2. The method of claim 1 further comprising:
extracting the foreground elements by binarizing the alpha channel or luminance of a foreground layer to create a binary image; and
determining connected components in the binary image.

3. The method of claim 1 further comprising:
extracting the background elements by binarizing the alpha channel of a background layer to create a binary image; and
determining connected components in the binary image.

4. The method of claim 1 wherein separating the foreground elements from the background elements comprises measuring the symmetry, size, and complexity of each element.

5. The method of claim 4 wherein elements having high symmetry, high size and low complexity are identified as background elements.

6. The method of claim 1 wherein creating resized foreground elements comprises:
determining a horizontal scaling factor between the first size and the second size;
determining a vertical scaling factor between the first size and the second size; and
scaling the foreground element by the minimum of the horizontal and vertical scaling factors.

7. The method of claim 1 further comprising identifying a rectangular background element and separating the rectangular background element into horizontal and vertical elements.

8. A digital image comprising:
multiple elements wherein each element comprises:
metadata indicating whether the element is a foreground or background element;
metadata indicating whether the aspect ratio of the element should be maintained when the image is resized; and
metadata indicating how the element should be repositioned when the image is resized.

9. The digital image of claim 8 wherein the image is expressed as an XML document.

10. The digital image of claim 9 wherein the XML document identifies the location of each element.

11. The digital image of claim 10 wherein the metadata indicating whether the element is a foreground or background element is identified in the XML document.

12. A computer-readable medium having computer-executable instructions for auto-resizing an original digital image from an original image to a resized image, where the original image includes at least one foreground element and at least one background element, the instructions comprising:

separating the foreground elements from the background elements;
determining an original aspect ratio of the foreground elements;
creating resized foreground elements by scaling the foreground elements while maintaining the original aspect ratio;
creating resized background elements by stretching or shrinking the background elements to fit the dimensions of the resized image; and
repositioning the resized foreground and background elements on the resized image while maintaining the design intention of the original image.

13. The computer readable medium of claim 12 further comprising instructions for:

determining a horizontal scaling factor between the first size and the second size;
determining a vertical scaling factor between the first size and the second size; and
scaling the foreground element by the minimum of the horizontal and vertical scaling factors.

14. The computer readable medium of claim 12 wherein the original image is rectangular, the computer readable medium further comprising instructions for:

determining whether each foreground element is a corner, horizontal, or vertical element.

15. The computer readable medium of claim 12 further comprising instructions for:

measuring the symmetry, size, and complexity of each element in order to determine if each element is a foreground element or a background element.

16. The computer readable medium of claim 12 wherein the original image is rectangular and repositioning the resized foreground elements comprises, for each foreground element, determining whether the element is a corner, horizontal, or vertical element.

17. The computer readable medium of claim 12 where the foreground element is determined to be a corner element if any pair of adjacent element borders is within a given threshold distance from a pair of adjacent borders in the original image

18. The computer readable medium of claim 17 further comprising instructions for classifying a corner element as a bottom-left, bottom-right, top left, or top right element.

19. The computer readable medium of claim 18 wherein repositioning the element comprises positioning the corner elements so that they are at a similar relative position with respect to the corners of the resized image as the original image by using the corresponding corner in the original image as a reference scaling axis.

20. The computer readable medium of claim 12 where the foreground element is determined to be a horizontal element if the element bounding box center lies closer to the left or right border of the original image than to the side borders of the original image.

21. The computer readable medium of claim 20 further comprising instructions for classifying a horizontal element as a top horizontal element if it lies closer to the top border of the original image than the bottom and as a bottom horizontal element if it lies closer to the bottom border of the original image than the top.

22. The computer readable medium of claim 12 where the foreground element is determined to be a vertical element if the element bounding box center lies closer to the left or right borders of the original image than the top or bottom borders.

23. The computer readable medium of claim 22 comprising instructions for classifying the vertical element as a left side element if it lies closer to the left border of the original image than the right border and as a right side element if it lies closer to the right border of the original image than the left border.

24. The computer readable medium of claim 12 wherein if an element is determined to be a horizontal or vertical element, repositioning the resized foreground element comprises shifting the element so that it occupies a similar relative position in the scaled image as it did in the original image.

* * * * *

Appendix: Fleming (Ex. 1004)

Appendix: Fleming (Ex. 1004)

- Ex. 1004, Fig. 1

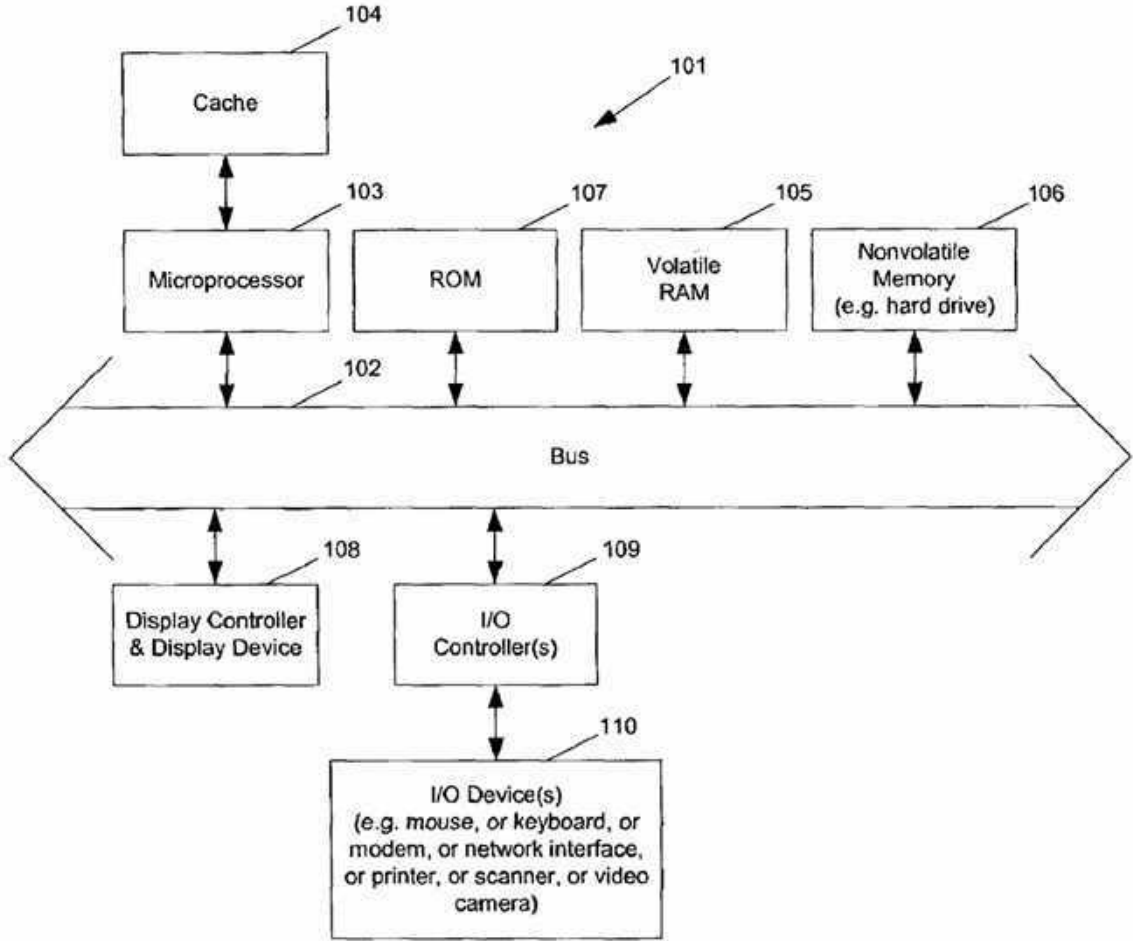


Fig. 1

Appendix: Fleming (Ex. 1004)

- Ex. 1004, Figs. 2-3

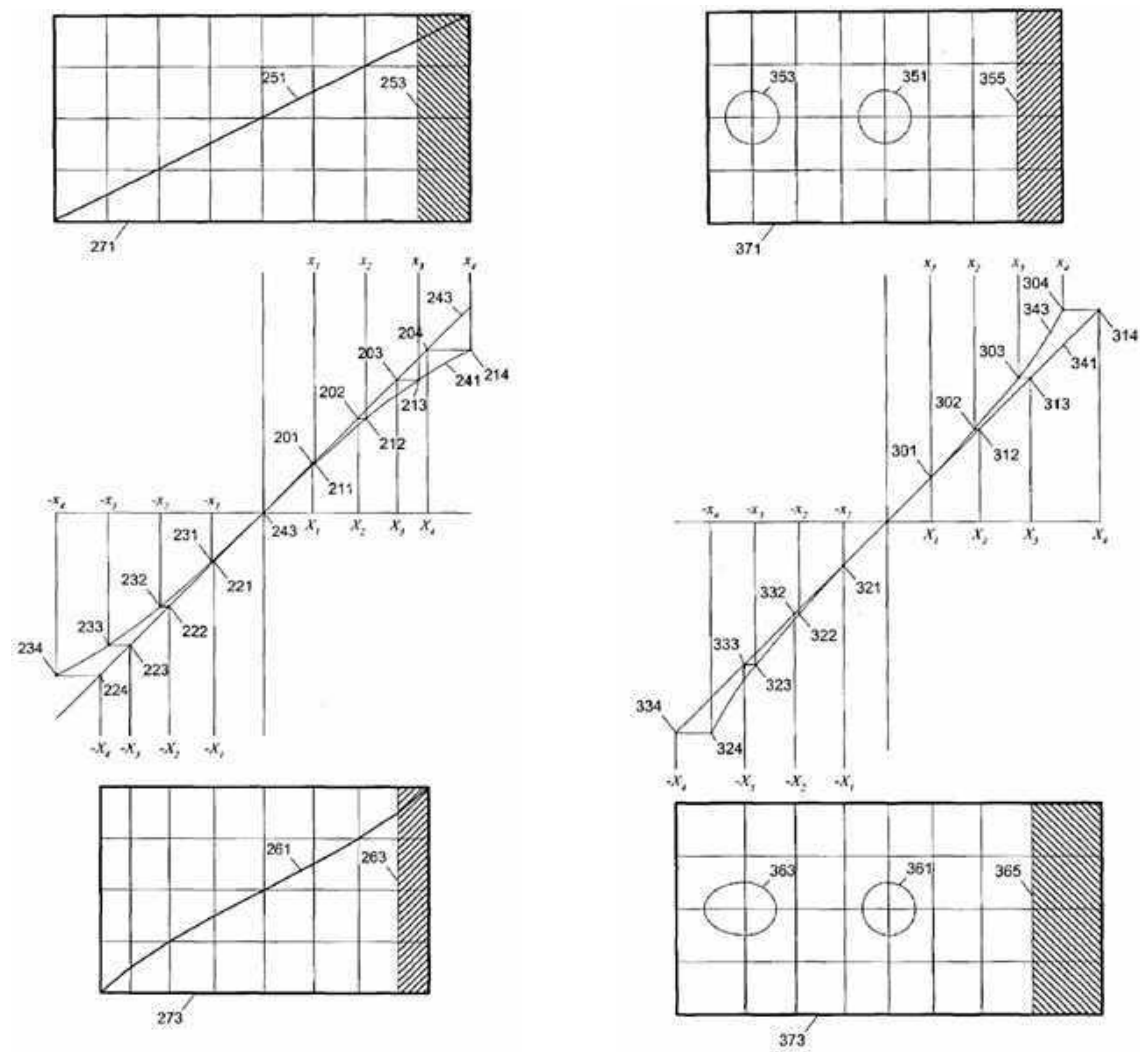


Fig. 2

Fig. 3

Appendix: Fleming (Ex. 1004)

- Ex. 1004, Figs. 4-5

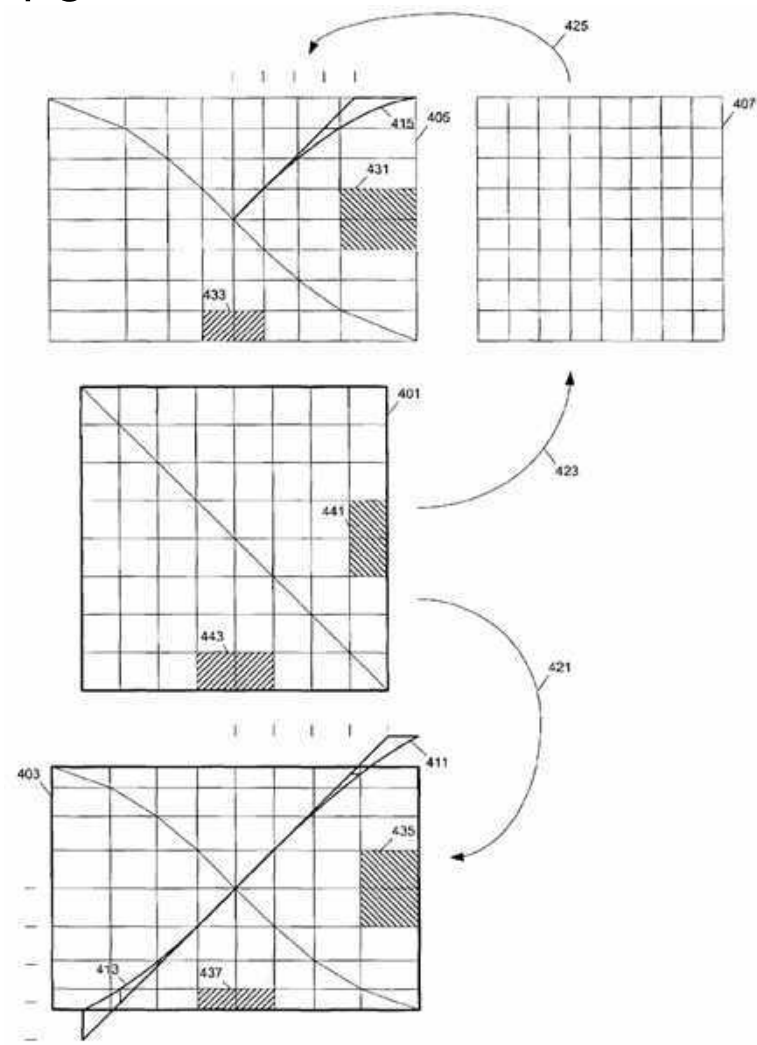


Fig. 4

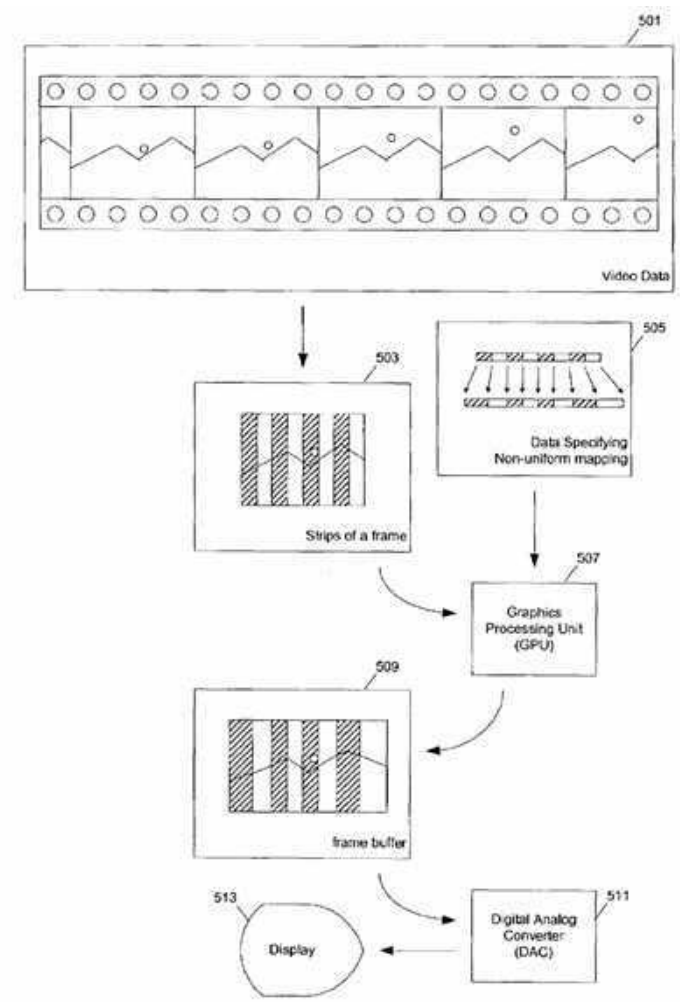


Fig. 5

Appendix: Fleming (Ex. 1004)

- Ex. 1004, Figs. 6-7

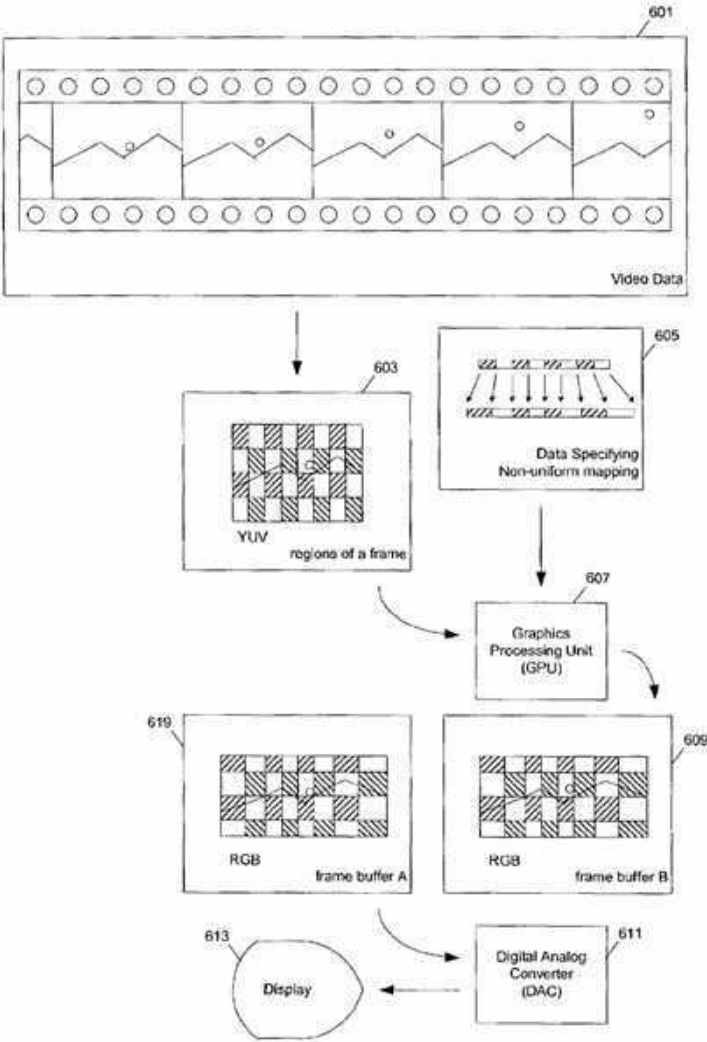


Fig. 6

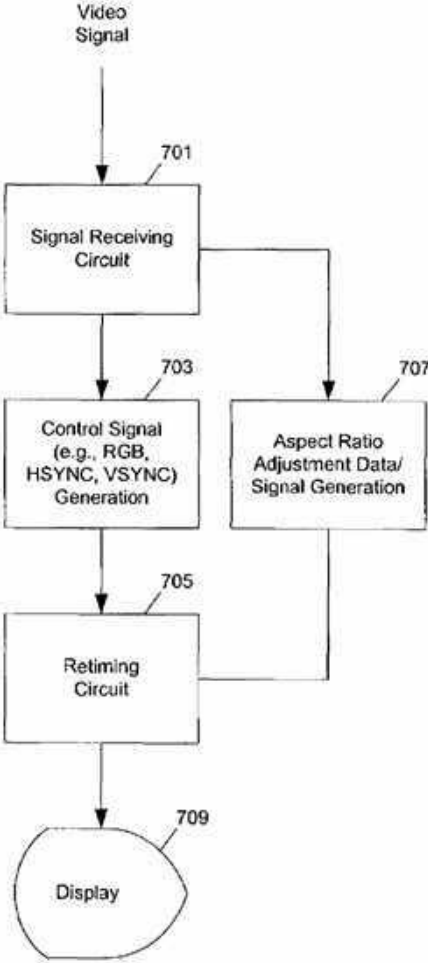


Fig. 7

Appendix: Fleming (Ex. 1004)

- Ex. 1004, Figs. 8-10

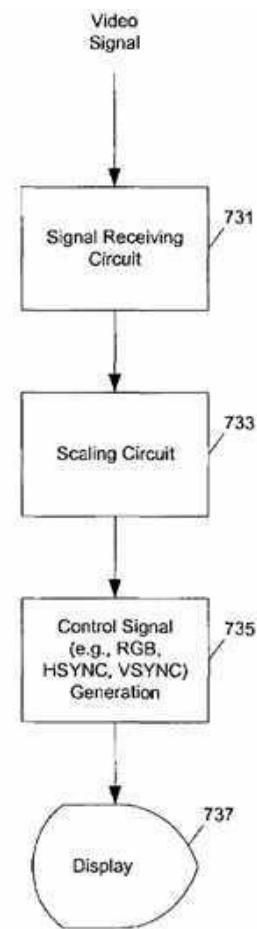


Fig. 8

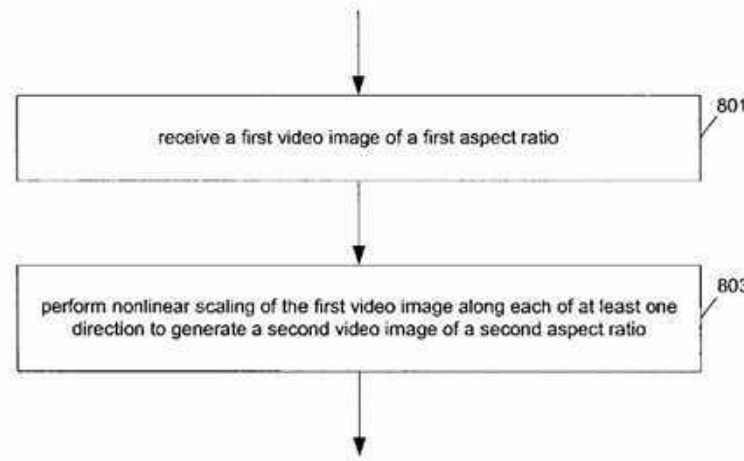


Fig. 9

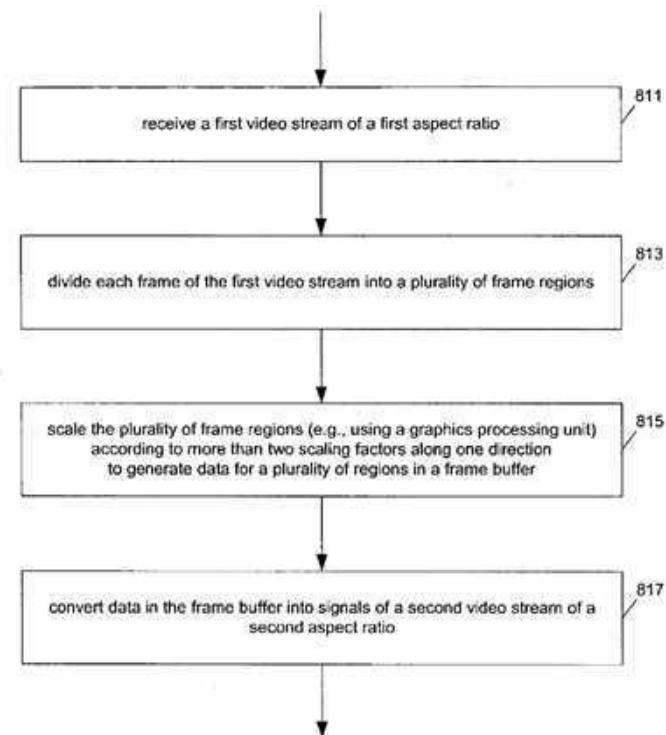


Fig. 10

Appendix: Fleming (Ex. 1004)

- Ex. 1004, Figs. 11-12

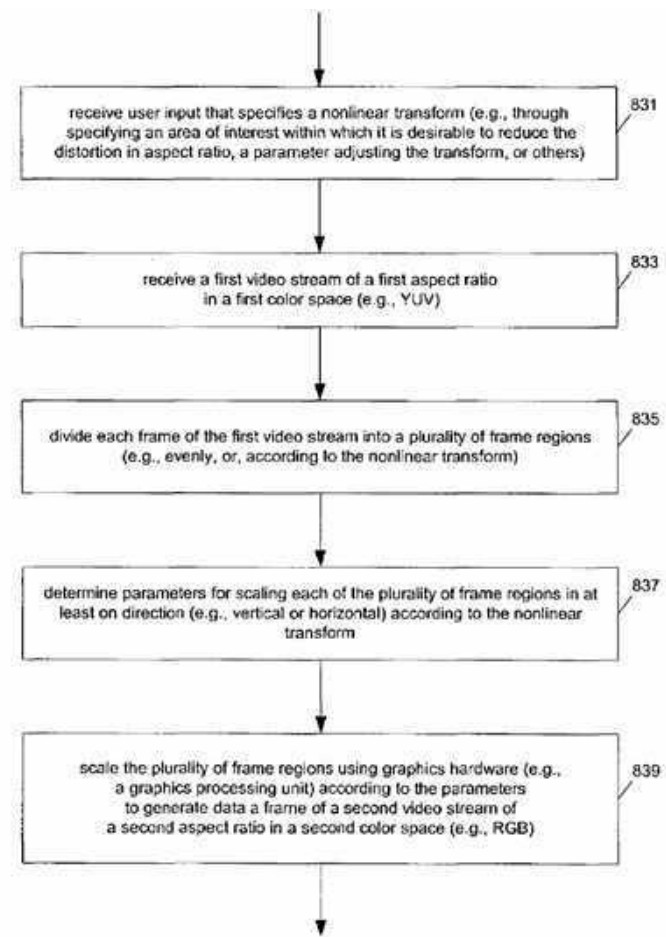


Fig. 11

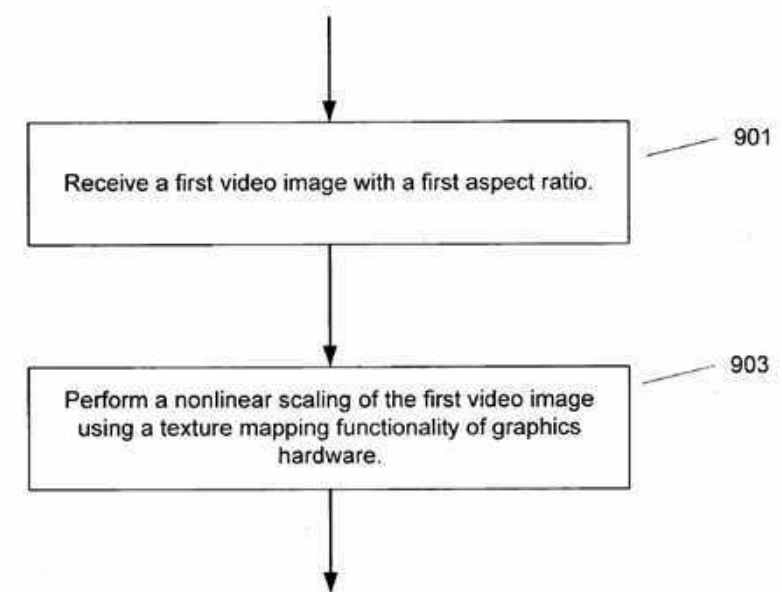


FIG. 12

Appendix: Fleming (Ex. 1004)

- Ex. 1004, cols. 1-2

1
METHOD AND APPARATUS FOR
NONLINEAR ANAMORPHIC SCALING OF
VIDEO IMAGES

FIELD OF THE INVENTION

The invention relates to display of video images, and more particularly to scaling of video images.

BACKGROUND OF THE INVENTION

A frame of a video image is typically represented by a rectangular array of pixels. The numbers of rows and columns of the rectangular array of pixels define the aspect ratio of the video image. A stream of video data typically contains a number of frames of video images, to be displayed at a given display area one frame after another. When the target display area is capable displaying a different number of rows or a different number of columns of pixels than those of the given video image, a portion of the target display area may be used to display the entire video image, or only a portion of the video image may be displayed on the target display area, without scaling.

To fit the entire video image onto the target display area, the video image is typically scaled linearly in the vertical direction and/or the horizontal direction. Since different scaling factors may be required to fit the entire video image onto the target display area, the aspect ratio of the original video image may be distorted after the scaling operation. Thus, a circle in the original video image may be displayed as an ellipse in the target display area.

SUMMARY OF THE DESCRIPTION

Methods and apparatuses for nonlinear scaling of video images are described here.

To match the aspect ratios of a video image and the target display area, at least one embodiment of the present invention scales the video image according to one or more nonlinear functions along the horizontal direction and/or the vertical direction. In one embodiment, the nonlinear functions are such that the original aspect ratio of the video image is preserved near the center region (or strip) of the image and the image is gradually stretched (or compressed) as it is mapped to the edges. In one example, the scaling is implemented by the texture mapping functionality of OpenGL using graphics hardware. In one embodiment of the present invention, the nonlinear mapping is constructed according to a polynomial mapping; and, the coefficients of the polynomial are adjustable by a user to trade off distortion between the image center and the image edges, giving the user control over the location and the amount of distortion.

In at least one embodiment of the present invention, a method to scale a first video image having a first aspect ratio includes: performing nonlinear scaling of the first video image along each of at least one direction (e.g., a direction along the horizontal or vertical pixel lines of the first video image), to generate a second video image of a second aspect ratio that is different from the first aspect ratio. In one example, a first rectangular region of the first video image is scaled as a corresponding first rectangular region of the second video image; the aspect ratios of the first rectangular region of the first video image and the first rectangular region of the second video image are substantially the same; a second rectangular region of the first video image is scaled as a corresponding second rectangular region of the second video image; and, the aspect ratios of the second rectangular

2

region of the first video image and the second rectangular region of the second video image are substantially different. In one example, user input is received to specify one of: a) the first rectangular region of the first video image; and, b) the first rectangular region of the second video image. In one example, the first video image has a first rectangular boundary of the first aspect ratio; and, the second video image has a second rectangular boundary of the second aspect ratio. In one example, the nonlinear scaling of the first video image includes piecewise linear scaling of a plurality of regions according to different scaling factors; and, graphics hardware (e.g., a graphics processing unit (GPU)) of a data processing system is instructed (e.g., by the central processing unit (CPU) of the data processing system) to perform the piecewise linear scaling of the plurality of regions. In one example, user input is received to specify the nonlinear scaling; and, the plurality of regions for nonlinear scaling are determined according to the user input. In one example, the graphics hardware also converts the first video image from a first color space (e.g., YUV) to the second video image in a second color space (RGB). In one example, the second video image is stored in a frame buffer for display on a display device of the data processing system. In another example, a circuit is integrated with a display device for scaling different video signals of different aspect ratios.

The present invention includes methods and apparatuses which perform these methods, including data processing systems which perform these methods, and computer readable media which when executed on data processing systems cause the systems to perform these methods.

Other features of the present invention will be apparent from the accompanying drawings and from the detailed description which follows.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example and not limitation in the figures of the accompanying drawings in which like references indicate similar elements.

FIG. 1 shows a block diagram example of a data processing system which may be used with the present invention.

FIGS. 2-4 show examples of nonlinear anamorphic scaling of video images according to embodiments of the present invention.

FIGS. 5-6 show examples of nonlinear anamorphic scaling of video images using graphics hardware of a data processing system according to embodiments of the present invention.

FIGS. 7-8 show block diagram examples of display devices for nonlinear anamorphic scaling of video images according to embodiments of the present invention.

FIG. 9 shows a method to scale video images according to one embodiment of the present invention.

FIG. 10 shows a detailed method to piecewisely scale video images according to one embodiment of the present invention.

FIG. 11 shows a method to adjust the aspect ratio of a video image according to one embodiment of the present invention.

FIG. 12 shows a method according to one embodiment of the present invention.

DETAILED DESCRIPTION

The following description and drawings are illustrative of the invention and are not to be construed as limiting the invention. Numerous specific details are described to pro-

Appendix: Fleming (Ex. 1004)

- Ex. 1004, cols. 3-4

3
vide a thorough understanding of the present invention. However, in certain instances, well known or conventional details are not described in order to avoid obscuring the description of the present invention. References to an or one embodiment in the present disclosure are not necessary to the same embodiment; and, such references means at least one.

FIG. 1 shows one example of a typical computer system which may be used with the present invention. Note that while FIG. 1 illustrates various components of a computer system, it is not intended to represent any particular architecture or manner of interconnecting the components as such details are not germane to the present invention. It will also be appreciated that network computers and other data processing systems which have fewer components or perhaps more components may also be used with the present invention. The computer system of FIG. 1 may, for example, be an Apple Macintosh computer.

As shown in FIG. 1, the computer system 101, which is a form of a data processing system, includes a bus 102 which is coupled to a microprocessor 103 and a ROM 107 and volatile RAM 105 and a non-volatile memory 106. The microprocessor 103, which may be, for example, a G3 or G4 microprocessor from Motorola, Inc. or IBM is coupled to cache memory 104 as shown in the example of FIG. 1. The bus 102 interconnects these various components together and also interconnects these components 103, 107, 105, and 106 to a display controller and display device 108 and to peripheral devices such as input/output (I/O) devices which may be mice, keyboards, modems, network interfaces, printers, scanners, video cameras and other devices which are well known in the art. Typically, the input/output devices 110 are coupled to the system through input/output controllers 109. The volatile RAM 105 is typically implemented as

dynamic RAM (DRAM) which requires power continually in order to refresh or maintain the data in the memory. The non-volatile memory 106 is typically a magnetic hard drive or a magnetic optical drive or an optical drive or a DVD RAM or other type of memory systems which maintain data even after power is removed from the system. Typically, the non-volatile memory will also be a random access memory although this is not required. While FIG. 1 shows that the non-volatile memory is a local device coupled directly to the rest of the components in the data processing system, it will be appreciated that the present invention may utilize a non-volatile memory which is remote from the system, such as a network storage device which is coupled to the data processing system through a network interface such as a modem or Ethernet interface. The bus 102 may include one or more buses connected to each other through various bridges, controllers and/or adapters as is well known in the art. In one embodiment the I/O controller 109 includes a USB (Universal Serial Bus) adapter for controlling USB peripherals, and/or an IEEE-1394 bus adapter for controlling IEEE-1394 peripherals.

It will be apparent from this description that aspects of the present invention may be embodied, at least in part, in software. That is, the techniques may be carried out in a computer system or other data processing system in response to its processor, such as a microprocessor, executing sequences of instructions contained in a memory, such as ROM 107, volatile RAM 105, non-volatile memory 106, cache 104 or a remote storage device. In various embodiments, hardwired circuitry may be used in combination with software instructions to implement the present invention. Thus, the techniques are not limited to any specific combination of hardware circuitry and software nor to any par-

4
ticular source for the instructions executed by the data processing system. In addition, throughout this description, various functions and operations are described as being performed by or caused by software code to simplify description. However, those skilled in the art will recognize what is meant by such expressions is that the functions result from execution of the code by a processor, such as the microprocessor 103.

A machine readable medium can be used to store software and data which when executed by a data processing system causes the system to perform various methods of the present invention. This executable software and data may be stored in various places including for example ROM 107, volatile RAM 105, non-volatile memory 106 and/or cache 104 as shown in FIG. 1. Portions of this software and/or data may be stored in any one of these storage devices.

Thus, a machine readable medium includes any mechanism that provides (i.e., stores and/or transmits) information in a form accessible by a machine (e.g., a computer, network device, personal digital assistant, manufacturing tool, any device with a set of one or more processors, etc.). For example, a machine readable medium includes recordable/non-recordable media (e.g., read only memory (ROM); random access memory (RAM); magnetic disk storage media; optical storage media; flash memory devices; etc.), as well as electrical, optical, acoustical or other forms of propagated signals (e.g., carrier waves, infrared signals, digital signals, etc.); etc.

At least one embodiment of the present invention seeks to preserve the original aspect ratio near an area of interest using nonlinear scaling of video images.

When a video image is scaled linearly by different scaling factors along the horizontal and vertical directions, the aspect ratio of the video image is changed uniformly across the video image. A circle at any location on the original video image is displayed as an ellipse of the same aspect ratio.

According to one embodiment of the present invention, it is desirable to maintain the original aspect ratio in an area of interest (e.g., at the center of the image) and gradually stretch the image in other areas. Thus, the image in the area of interest is presented in the target display area with reduced distortion (or with no distortion), while other areas are distorted. Nonlinear scaling is used according to one embodiment of the present invention to achieve such results.

FIGS. 2-4 show examples of nonlinear anamorphic scaling of video images according to embodiments of the present invention.

In the example of FIG. 2, image 271 is scaled nonlinearly in the horizontal direction to reduce the width to that of image 273. To preserve the original aspect ratio near the center of image 271, nonlinear mapping according to non-linear curve 241 is used to map the positions of the vertical lines along the horizontal direction. For example, the vertical line at $x=x_4$ (214) in image 271 is mapped to the vertical line at $X=X_4$ (204) in image 273. Similarly, horizontal positions represented by points 213, 212 and 211 (x_3, x_2, x_1) in image 271 are mapped to horizontal positions represented by points 203, 202 and 201 (X_3, X_2, X_1) in image 273; and, 231-234 to 221-224. Thus, evenly spaced vertical lines in image 271 are displayed as non-evenly spaced vertical lines in image 273. For example, shaded region 253 is displayed as region 263. Since the nonlinear scaling is applied only to the horizontal direction, the change of aspect ratio is uniform along the vertical direction, while non-uniform along the horizontal direction. The slope of curve 241 near the center (243) of the image is adjusted so that the original aspect ratio

Appendix: Fleming (Ex. 1004)

- Ex. 1004, cols. 5-6

5

near the center is preserved in image 273. Thus, a square region at the center of image 271 is mapped (or scaled) as a square (or a rectangle substantially close to a square) in image 273. While the straight horizontal and vertical lines of image 271 are mapped as the corresponding straight horizontal and vertical lines of image 273, a straight line of a slant angle (e.g., line 251) is displayed as a curve (e.g., curve 261), due to the nonlinear scaling in the horizontal direction. It will be appreciated that "linear" means that a change in the x coordinate value (e.g., x changes from 1 to 2) produces a constant multiple of that change in the y coordinate value (e.g., y changes from 2 to 4 if $y=2x$). Non-linear normally means that a change in x produces a change in y which is other than a constant multiple of that change (or x value) along at least a portion of the function which describes the nonlinear relationship or curve.

In the example of FIG. 3, image 371 is scaled nonlinearly in the horizontal direction to increase the width to that of image 373. Nonlinear curve 343 is used to map the vertical lines of image 371 from positions 302-304 to positions 312-314 (on line 341), and 322-324 to 332-334. The positions of the vertical lines at $x=\pm x_1$ (301 and 321) remain unchanged. Thus, the center strip of image 371 is displayed without distortion as the center strip of image 373, while the side strips of image 371 (e.g., strip 355) are stretched horizontally (e.g., as strip 365) to fill the display region. Circle 351 in the center strip of image 371 is displayed as circle 361 in the center strip of image 373. However, circle 353 is distorted as curve 363. Since the detail at the center strip is typically of more interest to a viewer, the nonlinear scaling in the example preserves the original feature of the video image in the area of interest, the center strip.

FIGS. 2 and 3 illustrate the examples of nonlinear scaling in the horizontal direction, while the vertical direction is not scaled. In general, linear scaling in both directions can also be combined with nonlinear scaling (either in the vertical direction or in the horizontal direction) so that the aspect ratio of the area of interest (e.g., the center strip) is preserved. Typically, the linear scaling and the nonlinear scaling are performed in one combined operation.

According to one embodiment of the present invention, nonlinear scaling is applied in both the vertical direction and the horizontal direction. In FIG. 4, image 401 is stretched in the horizontal direction and compressed in the vertical direction to generate image 403. Nonlinear scaling is applied in both directions in operation 421. Curve 411 is used to map the horizontal positions of the vertical lines; and, curve 413 is used to map the vertical positions of the horizontal lines. Thus, while the aspect ratio of the portions of the image near the boundary is distorted, the aspect ratio of the center region of the image is preserved. Alternatively, linear scaling can be combined with nonlinear scaling in operations 423 and 425 to generate image 405. Curve 415 is used for the nonlinear horizontal scaling. In image 405, the aspect ratio of the center strip is preserved. For example, region 433 near the bottom of image 405 has the same aspect ratio as region 443 of image 401. However, when region 431 is compared to region 441, it is seen that the vertical boundary strips are severely distorted. In image 403, while both regions 437 and 435 are distorted in aspect ratio, region 435 has less distortion than region 431. Thus, the distortion of aspect ratio is distributed around the boundary region in image 403, while the distortion is concentrated in the boundary strips in image 405.

In FIG. 4, it is seen that image 407 can also be scaled nonlinearly in both directions to generate image 403. In one embodiment of the present invention, when a video image is

0

required to fill a display area with a linear horizontal scaling factor A and a linear vertical scaling factor B, where $A \neq B$, nonlinear scaling can be constructed to have a scaling factor C at the center of interest (e.g., the center of the image) to preserve the aspect ratio at the center of interest. In one embodiment of the present invention, C is chosen so that C is between A and B; and, the aspect ratio of the image is compressed nonlinearly along one direction and stretched nonlinearly along the other direction. When C approaches A, the horizontal scaling approaches from nonlinear scaling to linear scaling; when C approaches B, the vertical scaling approaches from nonlinear scaling to linear scaling. In one embodiment of the present invention, a user specifies the area of interest; and, the nonlinear scaling is constructed according to the location and the size of the area of interest.

Various nonlinear curves can be used for the construction of nonlinear scaling. For example, a cubical polynomial can be used to map the coordinates. For instance, when a video image is required to fill a display area with a linear horizontal scaling factor A and a linear vertical scaling factor B, the image can be linearly scaled by factor A in the horizontal direction using mapping $X=Ax$ and be nonlinearly scaled in the vertical direction using mapping $Y=y(A+(B-A)(y/H)^2)$, where H is the half height of the video image, (x, y) are the coordinates of a point in the video image, and (X, Y) is the coordinates of the corresponding point in the scaled image. Thus, the region near $y=0$ is scaled by a factor close to A in both the horizontal and vertical direction, while the regions near $y \pm H$ are scaled by a factor of A in the horizontal direction, and a factor of $(3B-2A)$. Functions other than the cubical polynomial can also be used. For example, a curve-fit function generated based on a number of control points can be used. The control points can be predefined or received from user input devices (e.g., through a control button, or a graphical user interface, or a voice recognition system). Piecewise functions (e.g., piecewise linear functions) can also be used.

FIGS. 5-6 show examples of nonlinear anamorphic scaling of video images using graphics hardware of a data processing system according to embodiments of the present invention.

In FIG. 5, a computer program processes video data 501 having separate frames (e.g., stored on memory of a data processing system, such as volatile RAM 105 or nonvolatile memory 106 in FIG. 1). Each frame may be part of a sequence of related images, as in a movie. Each frame of video image is nonlinearly scaled according to embodiments of the present invention using graphics hardware (e.g., display controller 108 in FIG. 1). For example, a frame is divided into a number of strips (503). The data specifying the non-uniform mapping (505) is used to define the scaling of each of strips using the graphics hardware (e.g., graphics processing unit (GPU) 507). Various graphics hardware known in the art can be used to efficiently scale the strips of images. The graphics hardware linearly scales each strip according to the instructions (e.g., sent from the central processing unit (CPU)). Through graphics function calls (e.g., OpenGL, or DirectX), the graphics hardware can be instructed to scale each strips linearly. Thus, each frame of image (503) is piecewise linearly scaled and stored in a frame buffer (509). Typically, a digital analog converter (DAC) 511 converts the data in the frame buffer into signals for controlling display 513. The target area in the frame buffer can be a portion of the frame buffer so that the video is displayed on only a portion of a display device (e.g., a Cathode Ray Tube (CRT) monitor, a Liquid Crystal Display (LCD) panel, or others). The target area can also be the

Appendix: Fleming (Ex. 1004)

- Ex. 1004, cols. 7-8

7

entire frame buffer so that the video is displayed on the entire display area of the display device.

Different schemes can be used in determining the strips and coordinates for instructing the graphics hardware to perform the scaling. For example, the original video image can be divided into evenly spaced strips so that the graphics hardware can scale them into strips of different widths for the frame buffer. Alternatively, the original video image may be so divided that, after the scaling, the corresponding strips have the same width. The nonlinear scaling in one direction can be to either stretch or compress the aspect ratio along this direction.

In FIG. 6, both the horizontal and the vertical directions are scaled piecewise linearly using a graphics processing unit (607). A frame of video image from video data 601 is divided into a number of rectangular regions (603). The graphics processing unit linearly scales each of the rectangular regions to achieve the overall nonlinear scaling in both directions (e.g., by using different scaling factors for different regions according to the data specifying the non-uniform mapping (605)). In one embodiment of the present invention, the graphics hardware also converts the image data from one color space (e.g., YUV of the video data) to another (e.g., RGB suitable for controlling display device). In one embodiment of the present invention, double buffering is used for the frame buffer so that when the graphics processing unit is generating one frame of video image in one of the frame buffers (e.g., 609), the digital analog converter (DAC) (611) generates display signals for the previous frame in the other frame buffer (e.g., 619) to control display (613). After the graphics processing unit finishing scaling the current frame, the frame buffers are switched in roles.

In one embodiment of the present invention, the nonlinear scaling (e.g., the location and size of the area of interest) is adjustable by a user so that the user can adjust the scaling in real time according to the user's preferences.

At least some embodiments of the nonlinear scaling of the present inventions can also be applied to display devices, such as monitors, display panels, television sets or high definition television sets. The display devices according to one embodiment of the present invention contain circuits for nonlinearly scaling video signals to compensate distortions in aspect ratio due to linear scaling at areas of interest (typically the center region or strip of the display device).

FIGS. 7-8 show block diagram examples of display devices for nonlinear anamorphic scaling of video images according to embodiments of the present invention.

In FIG. 7, after the video signal is received from a source (e.g., an external connection, or a tuner) at signal receiving circuit 701, the control signals (e.g., RGB color signals, horizontal and vertical synchronization signals (HSYNC and VSYNC)) are generated (703) (e.g., according to linear scaling in both directions). Under control of aspect ratio adjustment data/signal (e.g., received from a control button of the display device, or determined from the input video signal and the aspect ratio of the display device), the retiming circuit (705) adjusts the timing of the horizontal (and/or vertical) scanline control signal to scale the image in the horizontal (or vertical) direction nonlinearly. For example, the horizontal scanline control signal of a CRT monitor can be changed from a linear function of time to a nonlinear function of time so that the timing of the activation of the pixels on the scanline is remapped to effectively perform nonlinear scaling along the horizontal direction. In one embodiment of the present invention, the information about the original aspect ratio of the video signal (701) is

8

combined with the information of the aspect ratio of the display device (709) and the user input about the nonlinear scaling (e.g., the area of interest, or, the degree of nonlinearity) to generate the aspect ratio adjustment data. From this description, a person skilled in the art will understand that various circuits can be used to as the retiming circuit (705) to perform the nonlinear scaling. In addition to analog circuits for performing the nonlinear scaling (e.g., using the nonlinear scanline control signals), digital circuits can also be used, when analog to digital and/or digital to analog converters are used.

For example, in FIG. 8, video signal (e.g., signals from an external source, a television tuner, a high definition television tuner, a digital video source, such as digital cable television, digital satellite television) is received at the signal receiving circuit 731 for display on device 737. To fit the video image into the display area of display device 737, scaling circuit 733 nonlinearly scales the video image (e.g., to preserve the original aspect ratio of the video image at an area of interest, such as the center region or center strip of the display device). Then, control signals 735 (e.g., RGB signals and horizontal and vertical synchronization signals) are generated from displaying the video image on display device 737. In one embodiment of the present invention, scaling circuit 733 includes a buffer for pipelining the process of nonlinear scaling and the generation of control signals.

A typical CRT computer monitor contains buttons for adjusting various properties of the CRT monitor, such as the brightness, contrast, horizontal/vertical position or size of the display area, and the shape of the display area (e.g., trapezoid correction). According to one embodiment of the present invention, a display device also contains buttons for adjusting the nonlinear scaling.

FIG. 9 shows a method to scale video images according to one embodiment of the present invention. After operation 801 receives a first video image of a first aspect ratio, operation 803 performs nonlinear scaling of the first video image along each of at least one direction to generate a second video image of a second aspect ratio. In one embodiment of the present invention, horizontal and vertical lines of a video image are scaled as horizontal and vertical lines in the scaled image so that rectangular regions of the video image remains as a rectangular region (or a square, a special case of a rectangular region) after the nonlinear scaling.

FIG. 10 shows a detailed method to piecewisely scale video images according to one embodiment of the present invention. Operation 811 receives a first video stream of a first aspect ratio. Operation 813 divides each frame of the first video stream into a plurality of frame regions. Operation 815 scales the plurality of frame regions (e.g., using a graphics processing unit) according to more than two scaling factors along one direction to generate data for a plurality of regions in a frame buffer. Operation 817 converts data in the frame buffer into signals of a second video stream of a second aspect ratio.

FIG. 11 shows a method to adjust the aspect ratio of a video image according to one embodiment of the present invention. Operation 831 receives user input that specifies a nonlinear transform (e.g., through specifying an area of interest within which it is desirable to reduce the distortion in aspect ratio, a parameter adjusting the transform, or others). Operation 833 receives a first video stream of a first aspect ratio in a first color space (e.g., YUV). Operation 835 divides each frame of the first video stream into a plurality of frame regions (e.g., evenly, or, according to the nonlinear transform). For example, the frame of the first video stream

Appendix: Fleming (Ex. 1004)

• Ex. 1004, cols. 9-10

9

can be so divided that these regions, after the scaling, will be of even size on the target display area. Operation 837 determines parameters for scaling each of the plurality of frame regions in at least one direction (e.g., vertical or horizontal) according to the nonlinear transform. Operation 839 scales the plurality of frame regions using graphics hardware (e.g., a graphics processing unit) according to the parameters to generate data a frame of a second video stream of a second aspect ratio in a second color space (e.g., RGB).

FIG. 12 shows a method to scale video images according to one embodiment of the present invention. In this embodiment, the nonlinear scaling is performed by the texture mapping functionality of graphics systems, such as OpenGL or DirectX, implemented in graphics hardware. After receiving a first video image of a first aspect ratio in block 901, a nonlinear mapping function is constructed which scales the first video image to generate a second video image of a second aspect ratio. Then, the nonlinear scaling based on this mapping function is performed using texture functionalities of graphics hardware, as indicated in block 903.

In the foregoing specification, the invention has been described with reference to specific exemplary embodiments thereof. It will be evident that various modifications may be made thereto without departing from the broader spirit and scope of the invention as set forth in the following claims. The specification and drawings are, accordingly, to be regarded in an illustrative sense rather than a restrictive sense.

What is claimed is:

1. A method to scale a video image, the method comprising:

determining a nonlinear function from a position coordinate of a first video image having a first aspect ratio to a position coordinate of a second video image having a second aspect ratio, the second aspect ratio being different from the first aspect ratio, wherein the nonlinear function comprises piecewise linear scaling of a plurality of regions according to different scaling factors along at least one direction;

performing nonlinear scaling of the first video image along the at least one direction according to the nonlinear function to generate the second video image of the second aspect ratio, the second aspect ratio being different from the first aspect ratio; and

displaying the second video image on a display device.

2. A method as in claim 1 wherein the first aspect ratio is preserved at a center portion of the second video image.

3. A method as in claim 1 wherein the first video image comprises a video image frame from a video stream comprising a plurality of frames.

4. A method as in claim 1 wherein the determining comprises determining the nonlinear function based on user input.

5. A method as in claim 1 wherein a texture mapping functionality of graphics hardware is used to perform the nonlinear scaling.

6. A method as in claim 5 wherein the graphics hardware is instructed by graphics functions of one of: a) OpenGL; and, b) DirectX.

7. A method as in claim 1 wherein the second video image has at least two different aspect ratios within the second video image.

8. A method as in claim 1 wherein a first rectangular region of the first video image is scaled as a first rectangular region of the second video image; aspect ratios of the first rectangular region of the first video image and the first rectangular region of the second video image are substan-

10

tially the same; a second rectangular region of the first video image is scaled as a second rectangular region of the second video image; and aspect ratios of the second rectangular region of the first video image and the second rectangular region of the second video image are substantially different.

9. A method as in claim 8 further comprising: receiving input specifying one of:

a) the first rectangular region of the first video image; and
b) the first rectangular region of the second video image.

10. A method as in claim 1 wherein the first video image has a first rectangular boundary of the first aspect ratio; and, the second video image has a second rectangular boundary of the second aspect ratio.

11. A method as in claim 10 wherein the at least one direction comprises at least one of:

a) a direction along horizontal pixel lines of the first video image; or
b) a direction along vertical pixel lines of the first video image.

12. A method as in claim 11 wherein the at least one direction comprises:

a) a direction along horizontal pixel lines of the first video image; and
b) a direction along vertical pixel lines of the first video image.

13. A method as in claim 1 wherein the plurality of regions are equally divided along the at least one direction.

14. A method as in claim 1 further comprising: instructing graphics hardware of a data processing system to perform the piecewise linear scaling of the plurality of regions.

15. A method as in claim 14 wherein the graphics hardware comprises a graphics processing unit (GPU); and, the data processing system comprises a central processing unit (CPU).

16. A method as in claim 15 further comprising: receiving input specifying the nonlinear scaling.

17. A method as in claim 16 further comprising: determining the plurality of regions according to the input.

18. A method as in claim 14 wherein the graphics hardware converts the first video image from a first color space to the second video image in a second color space.

19. A method as in claim 18 wherein the second video image is stored in a frame buffer for display on a display device of the data processing system.

20. A computer readable medium containing executable computer program instructions which when executed by a digital processing system cause said system to perform a method to scale a video image, the method comprising:

determining a nonlinear function from a position coordinate of a first video image having a first aspect ratio to a position coordinate of a second video image having a second aspect ratio, the second aspect ratio being different from the first aspect ratio, wherein the nonlinear function comprises piecewise linear scaling of a plurality of regions according to different scaling factors along at least one direction;

performing nonlinear scaling of the first video image along the at least one direction according to the nonlinear function to generate the second video image of the second aspect ratio, the second aspect ratio being different from the first aspect ratio; and

displaying the second video image on a display device.

21. A medium as in claim 20 wherein the first aspect ratio is preserved at a center portion of the second video image.

Appendix: Fleming (Ex. 1004)

• Ex. 1004, cols. 11-12

11

22. A medium as in claim 20 wherein the first video image comprises a video image frame from a video stream comprising a plurality of frames.

23. A medium as in claim 20 wherein the determining comprises determining the nonlinear function based on user input.

24. A medium as in claim 20 wherein a texture mapping functionality of graphics hardware is used to perform the nonlinear scaling.

25. A medium as in claim 24 wherein the graphics hardware is instructed by graphics functions of one of: a) OpenGL; and, b) DirectX.

26. A medium as in claim 20 wherein the second video image has at least two different aspect ratios within the second video image.

27. A medium as in claim 20 wherein a first rectangular region of the first video image is scaled as a first rectangular region of the second video image; aspect ratios of the first rectangular region of the first video image and the first rectangular region of the second video image are substantially the same; a second rectangular region of the first video image is scaled as a second rectangular region of the second video image; and aspect ratios of the second rectangular region of the first video image and the second rectangular region of the second video image are substantially different.

28. A medium as in claim 27 wherein the method further comprises:

receiving input specifying one of:

- a) the first rectangular region of the first video image; and
- b) the first rectangular region of the second video image.

29. A medium as in claim 20 wherein the first video image has a first rectangular boundary of the first aspect ratio; and, the second video image has a second rectangular boundary of the second aspect ratio.

30. A medium as in claim 29 wherein the at least one direction comprises at least one of:

- a) a direction along horizontal pixel lines of the first video image; or
- b) a direction along vertical pixel lines of the first video image.

31. A medium as in claim 30 wherein the at least one direction comprises:

- a) a direction along horizontal pixel lines of the first video image; and
- b) a direction along vertical pixel lines of the first video image.

32. A medium as in claim 20 wherein the plurality of regions are equally divided along the at least one direction.

33. A medium as in claim 20 wherein the method further comprises:

instructing graphics hardware of the digital processing system to perform the piecewise linear scaling of the plurality of regions.

34. A medium as in claim 33 wherein the graphics hardware comprises a graphics processing unit (GPU); and, the digital processing system comprises a central processing unit (CPU).

35. A medium as in claim 34 wherein the method further comprises:

receiving input specifying the nonlinear scaling.

36. A medium as in claim 35 wherein the method further comprises:

determining the plurality of regions according to the input.

12

37. A medium as in claim 33 wherein the graphics hardware converts the first video image from a first color space to the second video image in a second color space.

38. A medium as in claim 37 wherein the second video image is stored in a frame buffer for display on a display device of the digital processing system.

39. A data processing system to scale a video image, the data processing system comprising:

means for storing a nonlinear function from a position coordinate of a first video image having a first aspect ratio to a position coordinate of a second video image having a second aspect ratio, the second aspect ratio being different from the first aspect ratio, wherein the nonlinear function comprises piecewise linear scaling of a plurality of regions according to different scaling factors along at least one direction; and

means for performing nonlinear scaling of the first video image along the at least one direction according to the nonlinear function to generate the second video image of the second aspect ratio, the second aspect ratio being different from the first aspect ratio.

40. A data processing system as in claim 39 wherein the first aspect ratio is preserved at a center portion of the second video image.

41. A data processing system as in claim 39 wherein the first video image comprises a video image frame from a video stream comprising a plurality of frames.

42. A data processing system as in claim 39 wherein the determining comprises determining the nonlinear function based on user input.

43. A data processing system as in claim 39 wherein the means for performing the nonlinear scaling comprises graphics hardware; and, a texture mapping functionality of the graphics hardware is used to perform the nonlinear scaling.

44. A data processing system as in claim 43 wherein the graphics hardware is instructed by graphics functions of one of: a) OpenGL; and, b) DirectX.

45. A data processing system as in claim 39 wherein the second video image has at least two different aspect ratios within the second video image.

46. A data processing system as in claim 39 wherein a first rectangular region of the first video image is scaled as a first rectangular region of the second video image; aspect ratios of the first rectangular region of the first video image and the first rectangular region of the second video image are substantially the same; a second rectangular region of the first video image is scaled as a second rectangular region of the second video image; and aspect ratios of the second rectangular region of the first video image and the second rectangular region of the second video image are substantially different.

47. A data processing system as in claim 46 further comprising:

means for receiving input specifying one of:

- a) the first rectangular region of the first video image; and
- b) the first rectangular region of the second video image.

48. A data processing system as in claim 39 wherein the first video image has a first rectangular boundary of the first aspect ratio; and, the second video image has a second rectangular boundary of the second aspect ratio.

49. A data processing system as in claim 48 wherein the at least one direction comprises at least one of:

- a) a direction along horizontal pixel lines of the first video image; or
- b) a direction along vertical pixel lines of the first video image.

Appendix: Fleming (Ex. 1004)

• Ex. 1004, cols. 13-14

13

50. A data processing system as in claim 49 wherein the at least one direction comprises:
a) a direction along horizontal pixel lines of the first video image; and
b) a direction along vertical pixel lines of the first video image.
51. A data processing system as in claim 39 wherein the plurality of regions are equally divided along the at least one direction.
52. A data processing system as in claim 39 further comprising:
graphics hardware to perform the piecewise linear scaling of the plurality of regions.
53. A data processing system as in claim 52 wherein the graphics hardware comprises a graphics processing unit (GPU); and, the data processing system comprises a central processing unit (CPU).
54. A data processing system as in claim 53 further comprising:
means for receiving input specifying the nonlinear scaling.
55. A data processing system as in claim 54 further comprising:
means for determining the plurality of regions according to the input.
56. A data processing system as in claim 52 wherein the graphics hardware converts the first video image from a first color space to the second video image in a second color space.
57. A data processing system as in claim 56 wherein the second video image is stored in a frame buffer for display on a display device of the data processing system.

58. A digital processing system to scale a video image, the digital processing system comprising:
a bus;
memory coupled to the bus, the memory storing a first video image having a first aspect ratio and the memory storing a nonlinear function from a position coordinate of the first video image to a position coordinate of a second video image having a second aspect ratio, the second aspect ratio being different from the first aspect ratio, wherein the nonlinear function comprises piecewise linear scaling of a plurality of regions according to different scaling factors along at least one direction; and
a processor coupled to the bus, the processor performing nonlinear scaling of the first video image along the at least one direction according to the nonlinear function to generate the second video image of the second aspect ratio.
59. A digital processing system as in claim 58 wherein the first aspect ratio is preserved at a center portion of the second video image.
60. A digital processing system as in claim 58 wherein the first video image comprises a video image frame from a video stream comprising a plurality of frames.
61. A digital processing system as in claim 58 wherein the determining comprises determining the nonlinear function based on user input.
62. A digital processing system as in claim 58 further comprising:
graphics hardware coupled to the bus;
wherein a texture mapping functionality of the graphics hardware is used to perform the nonlinear scaling.
63. A digital processing system as in claim 62 wherein the graphics hardware is instructed by graphics functions of one of: a) OpenGL; and, b) DirectX.

14

64. A digital processing system as in claim 58 wherein the second video image has at least two different aspect ratios within the second video image.
65. A digital processing system as in claim 58 wherein a first rectangular region of the first video image is scaled as a first rectangular region of the second video image; aspect ratios of the first rectangular region of the first video image and the first rectangular region of the second video image are substantially the same; a second rectangular region of the first video image is scaled as a second rectangular region of the second video image; and aspect ratios of the second rectangular region of the first video image and the second rectangular region of the second video image are substantially different.
66. A digital processing system as in claim 65 further comprising:
an input device coupled to the bus to receive input specifying one of:
a) the first rectangular region of the first video image; and
b) the first rectangular region of the second video image.
67. A digital processing system as in claim 58 wherein the first video image has a first rectangular boundary of the first aspect ratio; and, the second video image has a second rectangular boundary of the second aspect ratio.
68. A digital processing system as in claim 67 wherein the at least one direction comprises at least one of:
a) a direction along horizontal pixel lines of the first video image; or
b) a direction along vertical pixel lines of the first video image.
69. A digital processing system as in claim 68 wherein the at least one direction comprises:
a) a direction along horizontal pixel lines of the first video image; and
b) a direction along vertical pixel lines of the first video image.

70. A digital processing system as in claim 58 wherein the plurality of regions are equally divided along the at least one direction.
71. A digital processing system as in claim 58 further comprising:
graphics hardware coupled to the bus, the graphics hardware performing the piecewise linear scaling of the plurality of regions in response to instructions from the processor.
72. A digital processing system as in claim 71 wherein the graphics hardware comprises a graphics processing unit (GPU).
73. A digital processing system as in claim 72 further comprising:
an input device coupled to the bus, the input device receiving input that specifies the nonlinear scaling.
74. A digital processing system as in claim 73 wherein the processor determines the plurality of regions according to the input.
75. A digital processing system as in claim 71 wherein the graphics hardware converts the first video image from a first color space to the second video image in a second color space.
76. A digital processing system as in claim 75 wherein the memory comprises a frame buffer; and, the second video image is stored in the frame buffer for display on a display device of the digital processing system.
77. An apparatus for displaying video images, the apparatus comprising:
a scaling circuit for scaling a received video image of a first aspect ratio nonlinearly along at least one direction

Appendix: Fleming (Ex. 1004)

- Ex. 1004, cols. 15-16

15

to generate an output video image of a second aspect ratio, the second aspect ratio being different from the first aspect ratio, wherein the scaling is performed according to a nonlinear function from a position coordinate of the received video image to a position coordinate of the output video image and wherein the nonlinear function comprises piecewise linear scaling of a plurality of regions according to different scaling factors along the at least one direction; and
a display device coupled to the scaling circuit to display the output video image.

78. An apparatus as in claim 77 wherein the first aspect ratio is preserved at a center portion of the second video image.

79. An apparatus as in claim 77 wherein the first video image comprises a video image frame from a video stream comprising a plurality of frames.

80. An apparatus as in claim 77 wherein the determining comprises determining the nonlinear function based on user input.

81. An apparatus as in claim 77 wherein the second video image has at least two different aspect ratios within the second video image.

16

82. An apparatus as in claim 77 wherein a first rectangular region of the first video image is scaled as a first rectangular region of the second video image; aspect ratios of the first rectangular region of the first video image and the first rectangular region of the second video image are substantially the same; a second rectangular region of the first video image is scaled as a second rectangular region of the second video image; and aspect ratios of the second rectangular region of the first video image and the second rectangular region of the second video image are substantially different.

83. An apparatus as in claim 77 further comprising:
a first signal receiving circuit coupled to the scaling circuit, the first signal receiving circuit receiving the first video image from a source external to the apparatus.

84. An apparatus as in claim 77 wherein the apparatus comprises a television set.

85. An apparatus as in claim 77 wherein the apparatus comprises computer monitor.

* * * * *

Appendix: Obara (Ex. 1006)

Appendix: Obara (Ex. 1006)

- Ex. 1006, Figs. 1-2

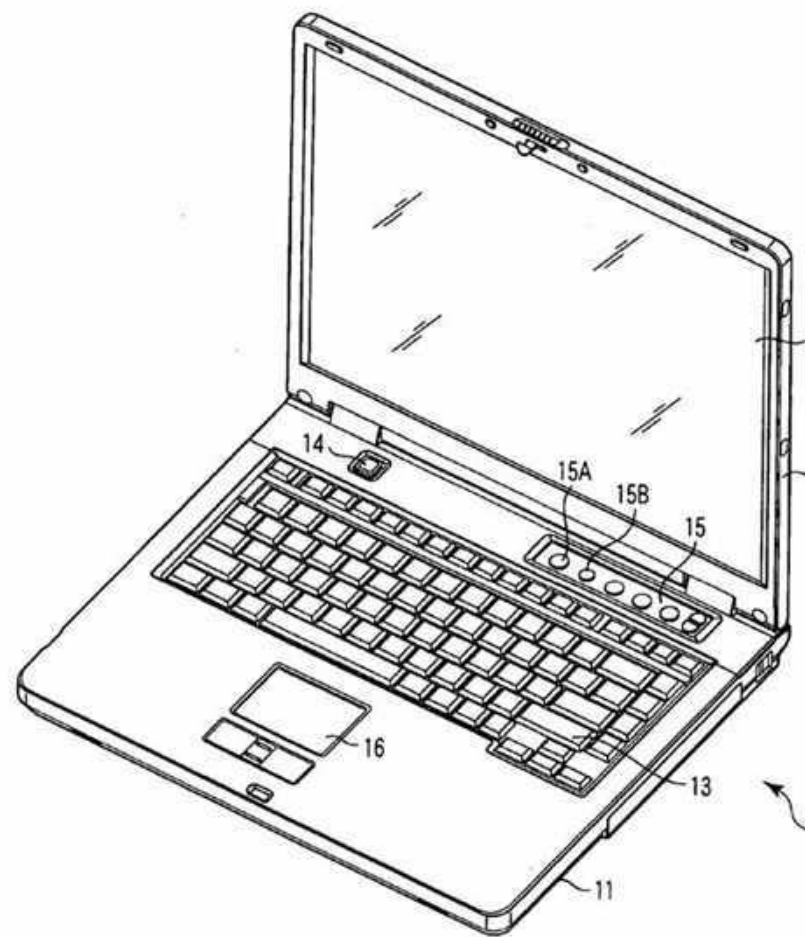


FIG. 1

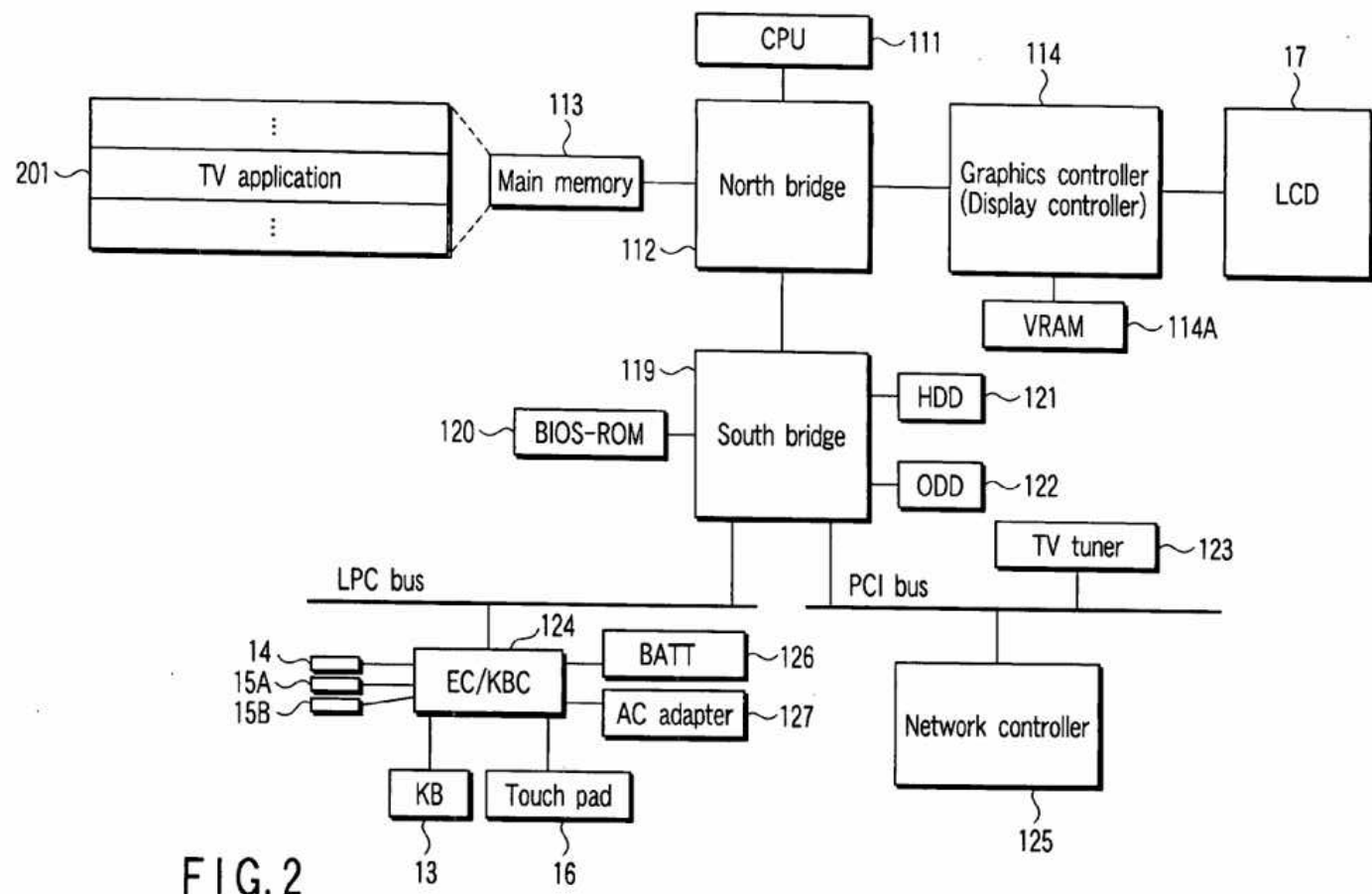


FIG. 2

Appendix: Obara (Ex. 1006)

- Ex. 1006, Fig. 3

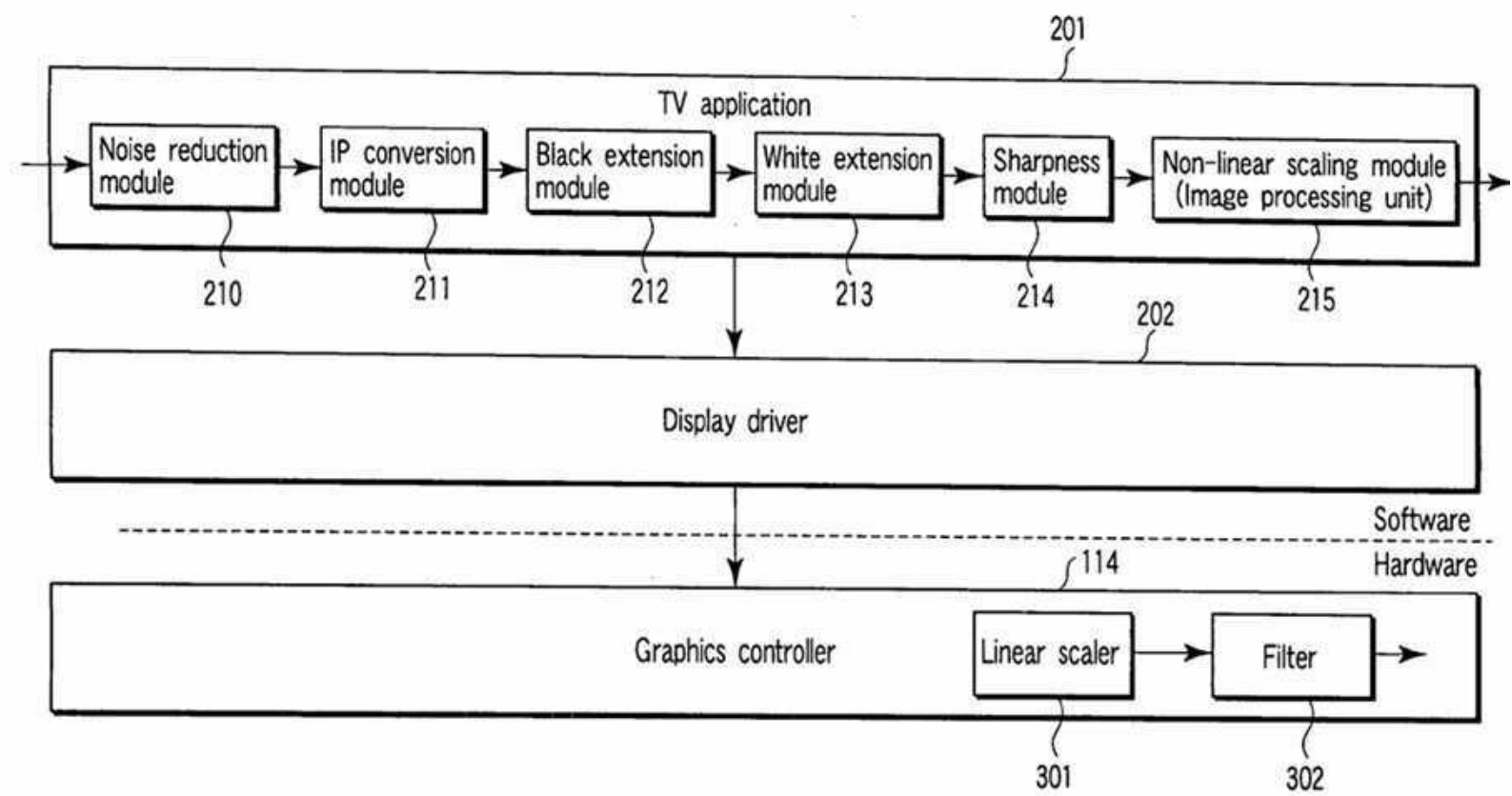


FIG. 3

Appendix: Obara (Ex. 1006)

- Ex. 1006, Fig. 4

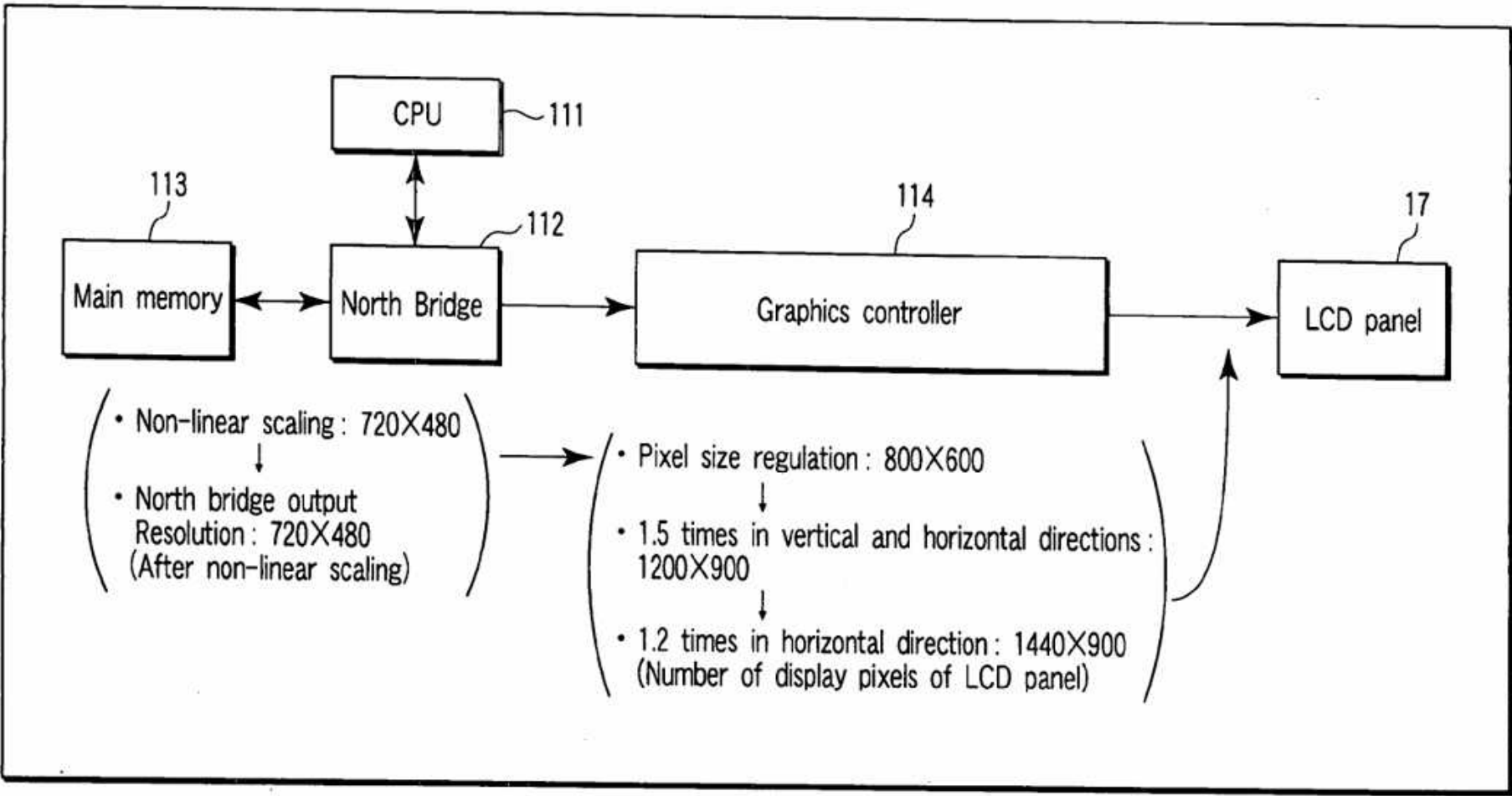


FIG. 4

Appendix: Obara (Ex. 1006)

- Ex. 1006, Figs. 5-6

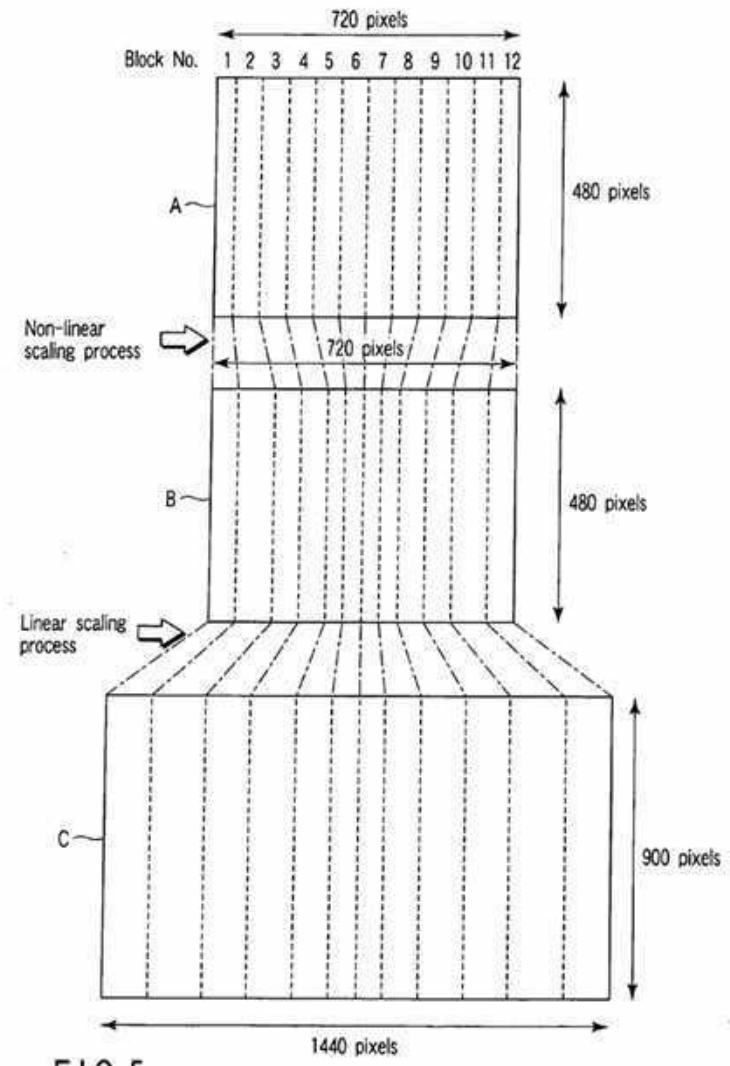


FIG. 5

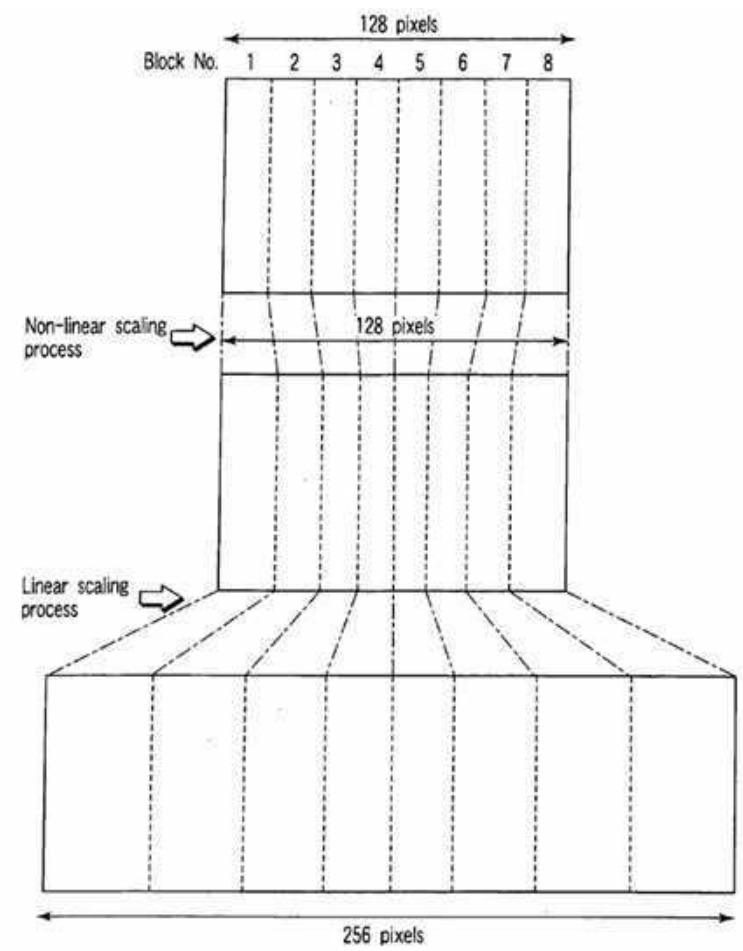


FIG. 6

Appendix: Obara (Ex. 1006)

- Ex. 1006, Fig. 7

Example in which enlargement ratio of each block is integer number of times of 1/(power of 2)

Block	1	2	3	4	5	6	7	8	Total
Horizontal pixel number	16	16	16	16	16	16	16	16	128
After non-linear scaling	19	17	15	13	13	15	17	19	128
Enlargement ratio	1.1875	1.0625	0.9375	0.8125	0.8125	0.9375	1.0625	1.1875	1
Coefficient	19/16	17/16	15/16	13/16	13/16	15/16	17/16	19/16	1
Horizontal double	38	34	30	26	26	30	34	38	256
Enlargement ratio relative to original pixels	2.375	2.125	1.875	1.625	1.625	1.875	2.125	2.375	2

FIG.7

Appendix: Obara (Ex. 1006)

- Ex. 1006, Fig. 8

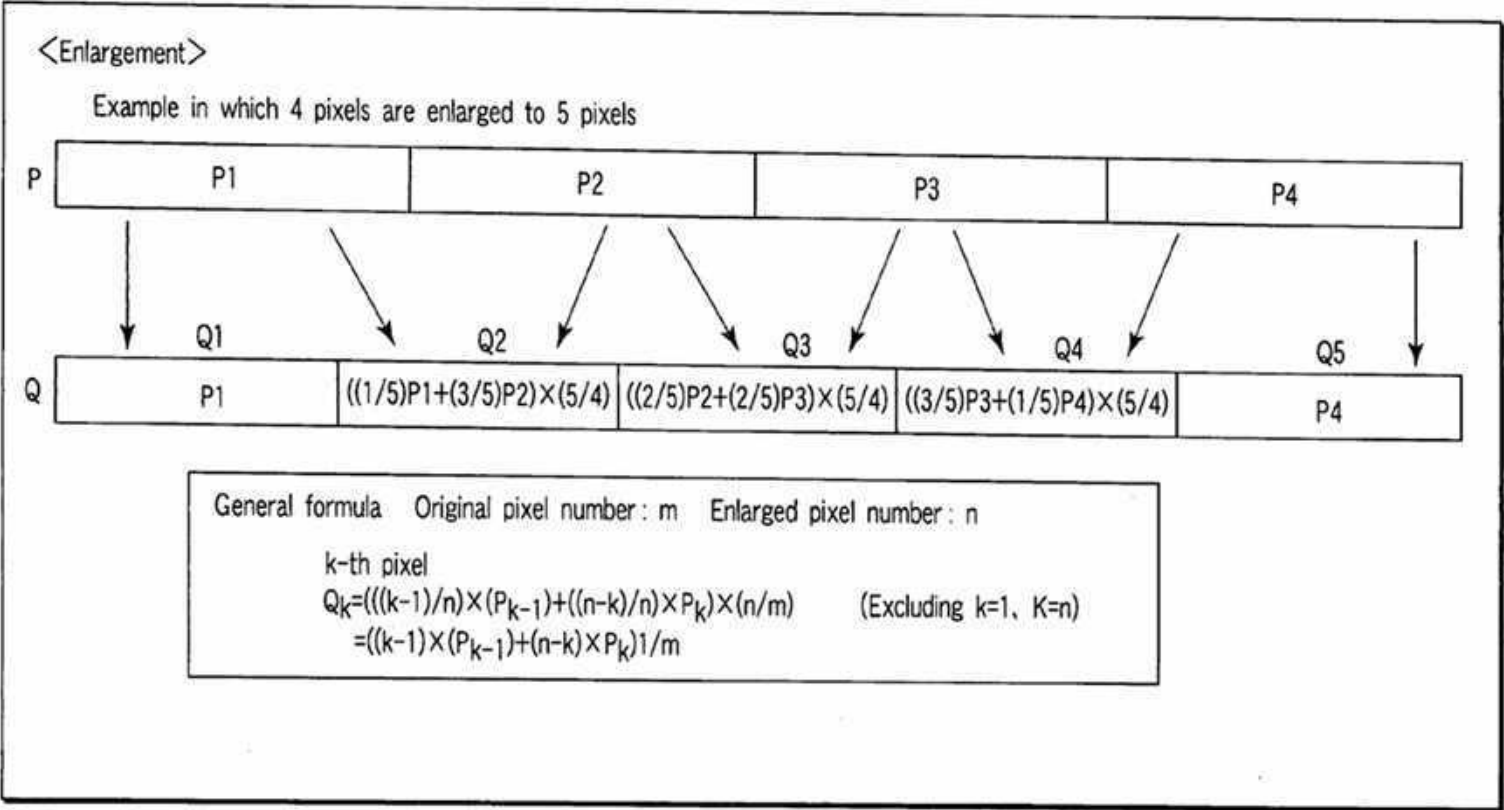


FIG. 8

Appendix: Obara (Ex. 1006)

- Ex. 1006, Fig. 9

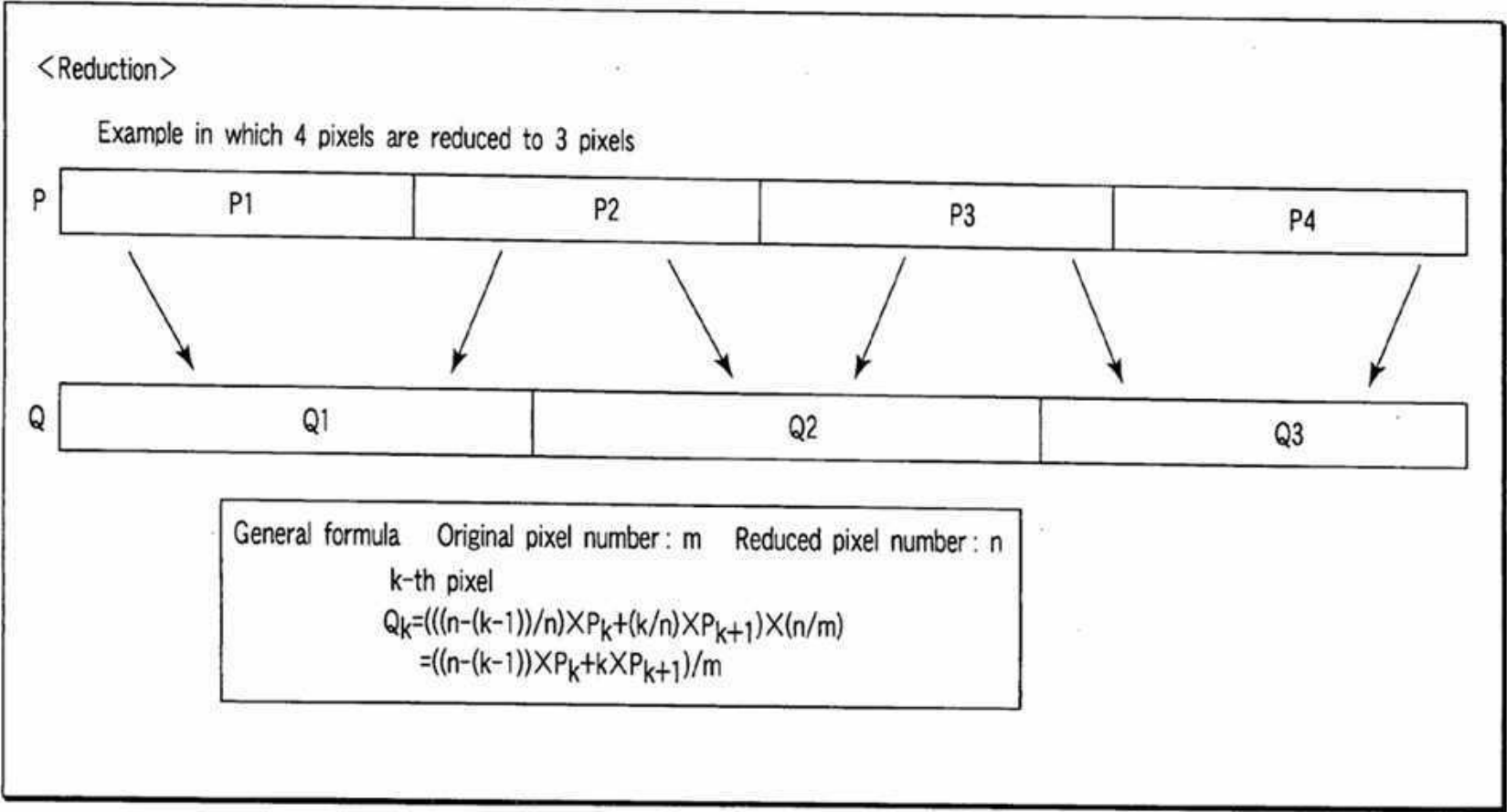


FIG. 9

Appendix: Obara (Ex. 1006)

- Ex. 1006, Fig. 10

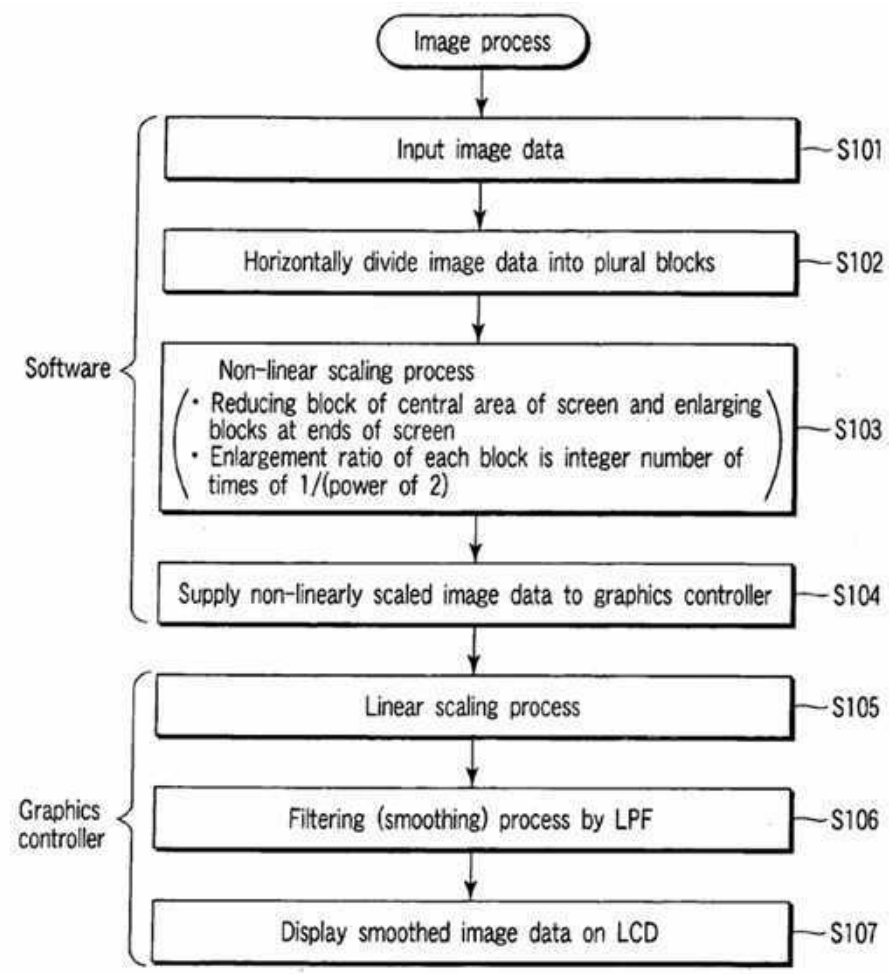


FIG. 10

Appendix: Obara (Ex. 1006)

• Ex. 1006, ¶¶1-24

INFORMATION PROCESSING APPARATUS AND IMAGE PROCESSING METHOD

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application is based upon and claims the benefit of priority from Japanese Patent Application No. 2005-172587, filed Jun. 13, 2005, the entire contents of which are incorporated herein by reference.

BACKGROUND

[0002] 1. Field

[0003] One embodiment of the invention relates to an information processing apparatus such as a personal computer, and an image processing method for use in the apparatus.

[0004] 2. Description of the Related Art

[0005] In general, in a TV apparatus, a scaling technology is used in order to display image data with an aspect ratio of, e.g. 4:3, on a wide screen having an aspect ratio of 16:9. Non-linear scaling is known as an example of the scaling technology.

[0006] Jpn. Pat. Appln. KOKAI Publication No. 2000-148128 discloses an image enlargement process circuit which enlarges image data by non-linear scaling.

[0007] Using the non-linear scaling, it becomes possible to display image data on the entire wide screen without causing horizontal enlargement distortion of an image of a central area of the image data.

[0008] In general cases, an image of motion video, which is displayed on a central part of the screen, is important. Thus, the non-linear scaling is an effective technique for enlarging motion video.

[0009] In recent years, personal computers, which have the same AV reproducing functions as audio/video (AV) apparatuses such as DVD (Digital Versatile Disc) players and TV apparatuses, have been developed. The non-linear scaling is also suitable for displaying motion video on display devices of personal computers.

[0010] However, in general, a display controller, which is used in the information processing apparatus such as the personal computer, is provided with only a linear scaling function which uniformly enlarges image data with a fixed enlargement ratio. It is thus necessary to provide the personal computer with a dedicated circuit for non-linearly enlarging image data that is output from the display controller. The provision of such a dedicated circuit, however, is a large factor that leads to an increase in manufacturing cost of the personal computer.

[0011] Under the circumstances, there is a need to realize a novel function for displaying non-linearly enlarged image data on a display device without using a dedicated circuit.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

[0012] A general architecture that implements the various feature of the invention will now be described with reference to the drawings. The drawings and the associated descrip-

tions are provided to illustrate embodiments of the invention and not to limit the scope of the invention.

[0013] FIG. 1 is an exemplary perspective view showing an example of the external appearance of a computer according to an embodiment of the present invention;

[0014] FIG. 2 is an exemplary block diagram showing an example of the system configuration of the computer shown in FIG. 1;

[0015] FIG. 3 is an exemplary block diagram showing an example of the functional configuration of a TV application program which is used in the computer shown in FIG. 1;

[0016] FIG. 4 is an exemplary view for explaining an example of the flow of image data in the computer shown in FIG. 1;

[0017] FIG. 5 is an exemplary view for describing a first example of an image data enlargement process which is executed by the computer shown in FIG. 1;

[0018] FIG. 6 is an exemplary view for describing a second example of the image data enlargement process which is executed by the computer shown in FIG. 1;

[0019] FIG. 7 is an exemplary view for explaining an example of an enlargement ratio which is applied to each of image blocks in the image data enlargement process illustrated in FIG. 6;

[0020] FIG. 8 is an exemplary view for explaining an example of a pixel interpolation process for horizontally enlarging image blocks by using enlargement ratios, each of which is an integral multiple of $1/2^n$, in the computer shown in FIG. 1;

[0021] FIG. 9 is an exemplary view for explaining an example of a pixel interpolation process for horizontally reducing image blocks by using enlargement ratios, each of which is an integral multiple of $1/2^n$, in the computer shown in FIG. 1; and

[0022] FIG. 10 is an exemplary flow chart illustrating an example of the procedure of an image process which is executed by the computer shown in FIG. 1.

DETAILED DESCRIPTION

[0023] Various embodiments according to the invention will be described hereinafter with reference to the accompanying drawings. In general, according to one embodiment of the invention, an information processing apparatus includes an image processing unit which executes a non-linear scaling process of vertically dividing input image data having a first size into a plurality of image blocks and horizontally reducing at least one of the image blocks, which belongs to a central area of the input image data, and a display controller which enlarges the input image data, which is subjected to the non-linear scaling process, by linear scaling, generates output image data having a second size greater than the first size, and displays the generated output image data on a display device.

[0024] To begin with, referring to FIG. 1 and FIG. 2, the structure of an information processing apparatus according to an embodiment of the invention is described. The information processing apparatus is realized, for example, as a notebook portable personal computer 10.

Appendix: Obara (Ex. 1006)

• Ex. 1006, ¶¶25-38

[0025] FIG. 1 is a perspective view that shows the state in which a display unit of the personal computer 10 is opened. The computer 10 comprises a computer main body 11 and a display unit 12. A display device that is composed of a TFT-LCD (Thin Film Transistor Liquid Crystal Display) 17 is built in the display unit 12. The display screen of the LCD 17 is positioned at an approximately central part of the display unit 12. The LCD 17 has a horizontally elongated wide display screen. The size (resolution) of the display screen is, for example, 1440×900 pixels.

[0026] The display unit 12 is attached to the computer main body 11 such that the display unit 12 is freely rotatable between an open position and a closed position. The computer main body 11 has a thin box-shaped casing. A keyboard 13, a power button 14 for powering on/off the computer 10, an input operation panel 15 and a touch pad 16 are disposed on the top surface of the computer main body 11.

[0027] The input operation panel 15 is an input device that inputs an event corresponding to a pressed button. The input operation panel 15 has a plurality of buttons for activating a plurality of functions. The buttons include a TV activation button 15A and a DVD/CD activation button 15B. The TV activation button 15A is a button for activating a TV function for reproducing, viewing/listening to, and recording TV broadcast program data. When the TV activation button 15A is pressed by the user, a TV application program for executing the TV function is automatically started.

[0028] In this computer 10, a general-purpose main operating system and a purpose-specific sub-operating system for processing AV (audio/video) data are installed. The TV application program is a program that runs on the sub-operating system.

[0029] When the power button 14 is pressed by the user, the main operating system is booted up. On the other hand, when the TV activation button 15A is pressed by the user, the sub-operating system, and not the main operating system, is booted up, and the TV application program is automatically executed. The sub-operating system has only a minimum function for executing the AV function. The time that is needed to boot up the sub-operating system is much shorter than the time that is needed to boot up the main operating system. Thus, only by pressing the TV activation button 15A, the user can immediately execute TV viewing/recording.

[0030] The DVD/CD activation button 15B is a button for reproducing video content that is recorded on a DVD or a CD. When the DVD/CD activation button 15B is pressed by the user, a video reproduction application program for reproducing video content is automatically started. The video reproduction application program, too, is an application program that runs on the sub-operating system. When the DVD/CD activation button 15B is pressed by the user, the sub-operating system, and not the main operating system, is booted up, and the video reproduction application program is automatically executed.

[0031] Next, referring to FIG. 2, the system configuration of the computer 10 is described.

[0032] The computer 10, as shown in FIG. 2, comprises a CPU 111, a north bridge 112, a main memory 113, a graphics controller 114, a south bridge 119, a BIOS-ROM 120, a hard disk drive (HDD) 121, an optical disc drive (ODD) 122, a

TV tuner 123, an embedded controller/keyboard controller IC (EC/KBC) 124, and a network controller 125.

[0033] The CPU 111 is a processor that is provided in order to control the operation of the computer 10. The CPU 111 executes the main operating system/sub-operating system and various application programs such as a TV application program 201, which are loaded from the hard disk drive (HDD) 121 into the main memory 113. The CPU 111 can execute a plurality of processes in parallel by using a plurality of pipelines.

[0034] The TV application program 201 has a function of enhancing the image quality of image data (motion video data) included in TV broadcast program data that is received by the TV tuner 123. The size (resolution) of the image data included in the TV broadcast program data is, e.g. 720×480 pixels. As shown in FIG. 3, the TV application program 201 includes, as video processing functions for enhancing the image quality of image data, a noise reduction module 210, an IP (Interlace/Progressive) conversion module 211, a black extension module 212, a white extension module 213, a sharpness module 214 and a non-linear scaling module 215. The noise reduction module 210, IP (Interlace/Progressive) conversion module 211, black extension module 212, white extension module 213, sharpness module 214 and non-linear scaling module 215 are software modules.

[0035] The noise reduction module 210 executes a process for removing noise from image data which is included in broadcast program data that is received by the TV tuner 123. The IP conversion module 211 executes a progressive conversion process for converting image data from interlace video to progressive video which has double the data amount of the interlace video. The black extension module 212 and white extension module 213 execute processes for extension correction of black and white levels, respectively. The sharpness module 214 executes a sharpness process for, e.g. outline emphasis.

[0036] The non-linear scaling module 215 is an image process unit which executes a non-linear scaling process for generating image data in which an image of a central area is horizontally reduced. Specifically, the non-linear scaling module 215 executes the non-linear scaling process of vertically dividing image data of, e.g. 720×480 pixels into a plurality of image blocks, and horizontally reducing at least one of the divided image blocks, which belongs to the central area of the image data. In the non-linear scaling process, the non-linear scaling module 215 executes a process of horizontally altering the size of at least one of image blocks, which belongs to the central area, with an enlargement ratio that is less than 1, and a process of horizontally altering the sizes of the other image blocks with enlargement ratios that gradually increase toward the end portions of the image data.

[0037] Image data (720×480 pixels), which has been subjected to the non-linear scaling process by the non-linear scaling module 215, is written in the video memory 114A of the graphics controller 114 via a display driver 202. The display driver 202 is software for controlling the graphics controller 114.

[0038] A linear scaler 301 and a filter 302 are provided in the graphics controller 114. The linear scaler 301 uniformly enlarges image data, which has been subjected to the non-

Appendix: Obara (Ex. 1006)

• Ex. 1006, ¶¶38-49

linear scaling process, with a fixed enlargement ratio by linear scaling, and generating output image data having a size (1440×900 pixels) corresponding to the size of the display screen of the LCD 17. The filter 302 is a filtering process unit which executes a filtering process for smoothing output image data that has been obtained by the linear scaling. The output image data, which has been subjected to the filtering process, is displayed on the LCD 17. The above-described process makes it possible to display image data, such as TV broadcast program data, on the entire wide screen of the LCD 17 with high image quality by a software process, without providing a dedicated circuit for executing a non-linear scaling process at the rear stage of the graphics controller 114.

[0039] The CPU 111 executes a system BIOS (Basic Input/Output System) that is stored in the BIOS-ROM 120. The system BIOS is a program for hardware control.

[0040] The north bridge 112 is a bridge device that connects a local bus of the CPU 111 and the south bridge 119. The north bridge 112 includes a memory controller that access-controls the main memory 113. The north bridge 112 has a function of executing communication with the graphics controller 114 via, e.g., an AGP (Accelerated Graphics Port) bus.

[0041] The graphics controller 114 is a display controller for controlling the LCD 17 that is used as a display monitor of the computer 10. The graphics controller 114 causes the LCD 17 to display image data, which is written in the video memory (VRAM) 114A. The graphics controller 114, as has been described with reference to FIG. 3, includes the linear scaler 301 and filter 302.

[0042] The south bridge 119 controls the devices on an LPC (Low Pin Count) bus, and the devices on a PCI (Peripheral Component Interconnect) bus. In addition, the south bridge 119 includes an IDE (Integrated Drive Electronics) controller for controlling the HDD 121 and ODD 122. The south bridge 119 also includes a function for access-controlling the BIOS-ROM 120.

[0043] The HDD 121 is a storage device that stores various software and data. The optical disc drive (ODD) 123 is a drive unit for driving storage media, such as a DVD or a CD, on which video content is stored. The TV tuner 123 is a receiving device for receiving broadcast program data such as TV broadcast program data. The TV tuner 123 includes an encoder for compression-encoding broadcast program data. When a recording process is executed to store received broadcast program data in the HDD 121, the received broadcast program data is compression-encoded by the encoder, and the compression-encoded broadcast program data is transferred to the main memory 113 via the PCI bus, south bridge 119 and north bridge 112. On the other hand, when received broadcast program data is displayed on the LCD 17, the received broadcast program data is not compression-encoded and is transferred to the main memory 113 via the PCI bus, south bridge 119 and north bridge 112.

[0044] The embedded controller/keyboard controller IC (EC/KBC) 124 is a 1-chip microcomputer in which an embedded controller for power management and a keyboard controller for controlling the keyboard (KB) 13 and touch pad 16 are integrated. The embedded controller/keyboard controller IC (EC/KBC) 124 has a function of powering

on/off the computer 10 in response to the user's operation of the power button 14. Operation power, which is supplied to the respective components of the computer 10, is generated from a battery 126 which is built in the computer 10 or from an external power supply which is supplied via an AC adapter 127.

[0045] Further, the embedded controller/keyboard controller IC (EC/KBC) 124 is capable of powering on/off the computer 10 in response to the user's operation of the TV activation button 15A or DVD/CD activation button 15B. The network controller 125 is a communication device that executes communication with an external network such as the Internet.

[0046] Next, referring to FIG. 4, the flow of image data in the case of executing non-linear scaling by a software process is described.

[0047] In the case where an image process is executed by software, if the load on the CPU 111 by the image process exceeds a predetermined level, other functions that are executed by the CPU 111 would be affected or real-time processing of image data would be disabled. Thus, when an image process is executed by software, it is very important to reduce the load on the CPU 111. In the present embodiment, in order to realize a size altering process (enlargement/reduction) of image data, which is necessary for a non-linear scaling process, by an arithmetic process with a relatively light load, the size altering process is executed with enlargement ratios each corresponding to an integer number of times of $1/(a \text{ power of } 2)$, i.e. an integral multiple of $1/2^n$ (n is an integer). Thereby, a pixel interpolation process for altering the size of an image in the horizontal direction can be realized by only an arithmetic process including bit shift and multiplication of integers, and the load on the CPU 111, which is necessary for the size altering process, can greatly be reduced.

[0048] By the non-linear scaling process which is executed by the CPU 111 on the main memory 113, a central part of image data with a resolution (number of pixels) of 720×480 is reduced in the horizontal direction, while the total number of pixels is unchanged in the horizontal and vertical directions. Further, the image data is horizontally altered in size with enlargement ratios that increase toward both ends of the image data. The image data (720×480 pixels), which is obtained by the non-linear scaling process, is transferred from the main memory 113 to the graphics controller 114 via the north bridge 112.

[0049] In the graphics controller 114, the image data (720×480 pixels), which is transferred from the north bridge 112 to the graphics controller 114, is subjected to pixel size regulation (i.e. a process for altering the pixel size to 800×600 pixels) and, at last, is linearly enlarged to image data of 1440×900 pixels. The enlarged image data is sent to the LCD 17. As a result, the image data, which is sent to the LCD 17, is a so-called non-linearly scaled image, which is obtained by horizontally enlarging the original image with enlargement ratios that gradually increase from a central area toward both ends of the image data. By the above process, image data with an aspect ratio of, e.g. 4:3 can be displayed on the entire wide screen of the LCD 17, which has an aspect ratio of, e.g. 16:9 or 16:10, without causing horizontal enlargement distortion of an image at the central area of the screen.

Appendix: Obara (Ex. 1006)

• Ex. 1006, ¶¶50-70

[0050] Next, referring to FIG. 5, a first example of the image data enlarging process, which is executed in the present embodiment, is described.

[0051] (1) Non-linear Scaling Process by Software

[0052] The size of original image data (A) with 720×480 pixels is altered such that a central area of the original image data (A) is horizontally reduced while the size of the original image data (A) is horizontally altered with enlargement ratios that gradually increase toward both ends of the image data, with the total number of pixels being unchanged in the horizontal and vertical directions. Thereby, image data (B) is obtained.

[0053] The non-linear scaling module 215 vertically divides original image data (A) with, e.g. 720×480 pixels into, e.g. 12 image blocks. The non-linear scaling module 215 horizontally reduces image blocks 6 and 7 which belong to the central area. Further, the non-linear scaling module 215 horizontally alters the sizes of the other image blocks with enlargement ratios that gradually increase toward both ends of the image data.

[0054] In this case, the enlargement ratio corresponding to each of the image blocks is set to be an integer number of times of $1/(a \text{ power of } 2)$, that is, $m/2^n$, where m and n are integers. Thereby, the arithmetic process for enlarging/reducing the image blocks can be executed by only bit shift and multiplication of integers.

[0055] In normal cases, image data that is non-linearly scaled needs to be subjected to a filtering process for smoothing. In the present embodiment, however, the non-linear scaling process is followed by the linear scaling process and filtering process in the graphics controller 114. Thus, the execution of a filtering process by a software process is omitted. The omission of the filtering process can further reduce the load on the CPU 111.

[0056] (2) Linear Scaling Process by Graphics Controller

[0057] In the graphics controller 114, for example, image data (B) is enlarged 1.5 times in the horizontal direction and vertical direction by the linear scaling process, and the image data (B) is further enlarged 1.2 times in the horizontal direction by the linear scaling process. Thereby, image data (C) of 1440×900 pixels, which is equal in size to the display screen of the LCD 17, is generated.

[0058] Next, referring to FIG. 6, a second example of the image data enlarging process, which is executed in the present embodiment, is described.

[0059] It is assumed that original image data is enlarged two times in the horizontal direction.

[0060] (1) Non-linear Scaling Process by Software

[0061] The total number of pixels of original image data (A) in the horizontal direction is 128. A central area of the original image data (A) is horizontally reduced without altering the total number of pixels in the horizontal and vertical directions, and the image data (A) is horizontally altered in size with enlargement ratios that gradually increase toward both ends. Thus, image data (B) is obtained.

[0062] The non-linear scaling module 215 vertically divides the original image data (A) into, for example, eight image blocks. The non-linear scaling module 215 horizontally

reduces image blocks 3, 4, 5 and 6 which belong to the central area. Further, the non-linear scaling module 215 horizontally alters (enlarges) the sizes of the other image blocks with enlargement ratios that gradually increase toward the blocks located at both ends of the image data.

[0063] (2) Linear Scaling Process by Graphics Controller

[0064] In the graphics controller 114, for example, the image data (B) is horizontally enlarged two times by the linear scaling process.

[0065] Referring now to FIG. 7, the enlargement ratios which are applied to the respective image blocks shown in FIG. 6 are explained.

[0066] An item "Horizontal Pixel Number" in FIG. 7 corresponds to the number of horizontal pixels of each of the blocks 1 to 8 of the original image data (A) shown in FIG. 6. An item "After Non-Linear Scaling" in FIG. 7 corresponds to the number of horizontal pixels of each of the blocks 1 to 8 of the image data (B) shown in FIG. 6. An item "Enlargement Ratio" in FIG. 7 indicates the value of the enlargement ratio which is applied to each of the blocks 1 to 8 of the original data (A). An item "Coefficient" in FIG. 7 expresses the enlargement ratio by a fraction. As is clear from the value of "Coefficient" in FIG. 7, the denominator of each of the coefficients is a power of 2. Thus, the enlargement ratio of each image block is an integral multiple of $1/2^n$. The enlargement ratio of each of the central blocks 4 and 5 is $0.8125 (=1^3/16)$. The enlargement ratio of each of

the blocks 3, 2 and 1, which are positioned on the left side of the central blocks 4 and 5, is greater than the enlargement ratio of each of the central blocks 4 and 5, and the enlargement ratios of the blocks 3, 2 and 1 increase in the named order. The enlargement ratio of each of the blocks 6, 7 and 8, which are positioned on the right side of the central blocks 4 and 5, is greater than the enlargement ratio of each of the central blocks 4 and 5, and the enlargement ratios of the blocks 6, 7 and 8 increase in the named order. An item "Horizontal Double" in FIG. 7 indicates the number of horizontal pixels of each of the blocks 1 to 8 after the number of horizontal pixels of each of the blocks 1 to 8 is increased two times in the horizontal direction by the linear scaling.

[0067] Next, a description is given of a pixel interpolation process which is executed by the non-linear scaling module 215 in order to alter (enlarge or reduce) the sizes of the image blocks with enlargement ratios each corresponding to an integral multiple of $1/2^n$.

[0068] As described above, by using the enlargement ratios each corresponding to an integral multiple of $1/2^n$, the arithmetic operation for executing the pixel interpolation process can greatly be simplified.

[0069] FIG. 8 shows an example of the pixel interpolation process in the case where the image block is horizontally enlarged with enlargement ratios each corresponding to an integral multiple of $1/2^n$.

[0070] For the purpose of simple description, assume the case in which the number of horizontal pixels of the image block is increased from four to five by the interpolation process (enlargement ratio= $5/4$). An original image block P is composed of four pixels P1 to P4. An enlarged image block Q is composed of five pixels Q1 to Q5.

Appendix: Obara (Ex. 1006)

• Ex. 1006, ¶¶71-87

[0071] The pixel value of the pixel Q1 is equal to that of the pixel P1. The pixel value of the pixel Q2 is generated from the pixel values of two pixels P1 and P2. The pixel value of the pixel Q3 is generated from the pixel values of two pixels P2 and P3. The pixel value of the pixel Q4 is generated from the pixel values of two pixels P3 and P4. The pixel value of the pixel Q5 is equal to that of the pixel P4.

[0072] A general formula of the pixel interpolation process in the case of enlargement is given by

$$Q_k = ((k-1)/n) \times (P_{k-1}) + ((n-k)/n) \times (P_k) \times (n/m)$$

(excluding

$$k = 1 \text{ and } k = n)$$
$$= ((k-1) \times (P_{k-1}) + ((n-k) \times P_k) / m$$

[0073] where m is the number of horizontal pixels of original image block P, n is the number of horizontal pixels of the enlarged image block Q, and Q_k is the pixel value of a k-th pixel within the enlarged image block Q.

[0074] As is clear from the formula (1), the value that appears in the denominator can be limited to m. Accordingly, by setting the enlargement ratio (n/m) at an integral multiple of 1/(a power of 2), the division of "1/m" can easily be executed by only bit shift.

[0075] FIG. 9 shows an example of the pixel interpolation process in the case where the image block is horizontally reduced with enlargement ratios each corresponding to an integral multiple of 1/2ⁿ.

[0076] For the purpose of simple description, assume the case in which the number of horizontal pixels of the image block is decreased from four to three by the interpolation process (enlargement ratio=3/4). An original image block P is composed of four pixels P1 to P4. A reduced image block Q is composed of three pixels Q1 to Q3.

[0077] The pixel value of the pixel Q1 is generated from the pixel values of two pixels P1 and P2. The pixel value of the pixel Q2 is generated from the pixel values of two pixels P2 and P3. The pixel value of the pixel Q3 is generated from the pixel values of two pixels P3 and P4.

[0078] A general formula of the pixel interpolation process in the case of reduction is given by

$$Q_k = ((n-(k-1))/n) \times (P_k) + ((k/n) \times (P_{k+1})) \times (n/m) = ((n-(k-1)) \times P_k + k \times P_{k+1}) / m$$

[0079] where m is the number of horizontal pixels of original image block P, n is the number of horizontal pixels of the reduced image block Q, and Q_k is the pixel value of a k-th pixel within the reduced image block Q.

[0080] As is clear from the formula (2), the value that appears in the denominator can also be limited to m in the arithmetic operation of the pixel interpolation process for reduction. Accordingly, by setting the enlargement ratio (n/m) at an integral multiple of 1/2ⁿ, the division of "1/m" can easily be executed by only bit shift.

[0081] Next, referring to a flow chart of FIG. 10, the procedure of the image process which is executed in the present embodiment is described.

[0082] To start with, the CPU 111, that is, the non-linear scaling module 215, inputs image data which is to be processed (block S101). The CPU 111, or the non-linear scaling module 215, divides input image data into a plurality of vertically extending image blocks (block S102), and subjects the input image data to a non-linear scaling process (block S103). In block S103, the CPU 111, or the non-linear scaling module 215, horizontally alters the size of at least one image block, which belongs to a central area of the input image data, with an enlargement ratio less than 1, thereby horizontally reducing the at least one image block. Further, in block S103, the CPU 111, or the non-linear scaling module 215, horizontally alters the sizes of the other image blocks with enlargement ratios which have values greater than the enlargement ratio corresponding to the image block of the central block and gradually increase toward end portions of the input image data. As regards any one of the image blocks of the input image data, a value corresponding to an integral multiple of 1/2ⁿ is chosen as the enlargement ratio. In the size altering process, the CPU 111, or the non-linear scaling module 215, executes an arithmetic process including multiplication of positive integers and bit shift.

[0083] Subsequently, the CPU 111, or the non-linear scaling module 215, sends the non-linearly scaled image data to the graphics controller 114 (block S104).

[0084] The graphics controller 114 horizontally enlarges the non-linearly scaled image data by linear scaling with a fixed horizontal enlargement ratio (or horizontally and vertically enlarges the non-linearly scaled image data by linear

scaling with a fixed horizontal enlargement ratio and a fixed vertical enlargement ratio), thereby generating output image data of a size corresponding to the screen size of the LCD 17 (block S105). Then, the graphics controller 114 executes a filtering process for smoothing the output image data by a filter circuit (LPF: Low-Pass Filter) 302 (block S106). The graphics controller 114 displays the smoothed output image data on the LCD 17 (block S107).

[0085] As has been described above, according to the present embodiment, the central area of the image data is once horizontally reduced by the software process, and the reduced image data is uniformly enlarged by the graphics controller 114 and displayed on the wide screen of the LCD 17. Thus, the non-linear scaling function can be realized by using an existing graphics controller, without providing a dedicated non-linear scaler circuit at a rear stage of the graphics controller 114. Therefore, the manufacturing cost can be reduced, compared to the case of providing a dedicated non-linear scaler circuit.

[0086] In the non-linear scaling process of the present embodiment, the enlargement ratio of each image block is an integral multiple of 1/2ⁿ. Hence, the pixel interpolation process for enlargement/reduction is realized by multiplication of integers and bit shift, and there is no need to execute a floating point arithmetic operation with a heavy load. Therefore, the non-linear scaling process can be executed by software without causing an increase of the load on the CPU 111.

[0087] In the graphics controller 114, the filter 302 is present at the rear stage of the linear scaler 301. Therefore, the execution of a filtering process after the non-linear scaling can advantageously be omitted.

Appendix: Obara (Ex. 1006)

- Ex. 1006, ¶¶88-90, cls. 1-12

[0088] In the non-linear scaling process by the software, it is possible to execute only the process of horizontally reducing the central area of the image data. In this case, too, it is possible to prevent horizontal enlargement distortion of a central area of the enlarged image.

[0089] The present embodiment is directed, by way of example, to the case in which image data, which is included in broadcast program data, is enlarged and displayed. The image processing method of this embodiment, however, is also applicable to the case in which image data or the like, which is read out of a storage medium such as a DVD, is enlarged and displayed. Furthermore, the non-linear scaling module 215 may be realized by hardware.

[0090] While certain embodiments of the inventions have been described, these embodiments have been presented by way of example only, and are not intended to limit the scope of the inventions. Indeed, the novel methods and systems described herein may be embodied in a variety of other forms; furthermore, various omissions, substitutions and changes in the form of the methods and systems described herein may be made without departing from the spirit of the inventions. The accompanying claims and their equivalents are intended to cover such forms or modifications as would fall within the scope and spirit of the inventions.

What is claimed is:

1. An information processing apparatus comprising:

an image processing unit which executes a non-linear scaling process of vertically dividing input image data having a first size into a plurality of image blocks and horizontally reducing at least one of the image blocks, which belongs to a central area of the input image data; and

a display controller which enlarges the input image data, which is subjected to the non-linear scaling process, by linear scaling, generates output image data having a second size greater than the first size, and displays the generated output image data on a display device.

2. The information processing apparatus according to claim 1, wherein the non-linear scaling process includes a process of horizontally altering a size of the at least one of the image blocks, which belongs to the central area, with an enlargement ratio which is less than 1, and a process of horizontally altering sizes of the other image blocks with enlargement ratios which increase toward ends of the input image data.

3. The information processing apparatus according to claim 1, wherein the non-linear scaling process includes a process of horizontally altering a size of the at least one of the image blocks, which belongs to the central area, with an enlargement ratio which is an integral multiple of $1/2^n$ and is less than 1, n being an integer.

4. The information processing apparatus according to claim 1, wherein the non-linear scaling process includes a process of horizontally altering sizes of the plurality of image blocks with a plurality of enlargement ratios having values, each of which is an integral multiple of $1/2^n$, n being an integer.

5. The information processing apparatus according to claim 1, wherein the non-linear scaling process includes a pixel interpolation process for horizontally altering sizes of the plurality of image blocks with a plurality of enlargement

ratios having values, each of which is an integral multiple of $1/2^n$, the n is an integer, and the image processing unit executes the pixel interpolation process by an arithmetic process including bit shift and multiplication of integers.

6. The information processing apparatus according to claim 1, wherein the display controller includes a filtering process unit which executes a filtering process of smoothing the generated output image data, and the output image data, which is subjected to the filtering process, is displayed on the display device.

7. The information processing apparatus according to claim 1, further comprising a receiving device which receives broadcast program data, wherein the input image data is the broadcast program data which is received by the receiving device.

8. An image processing method which processes image data using a display controller having a linear scaling function, comprising:

executing a non-linear scaling process of vertically dividing input image data having a first size into a plurality of image blocks and horizontally reducing at least one of the image blocks, which belongs to a central area of the input image data; and

sending the input image data, which is subjected to the non-linear scaling process, to the display controller, and causing the display controller to execute a process of enlarging the input image data, which is subjected to the non-linear scaling process, by linear scaling and generating output image data having a second size greater than the first size; and

displaying the generated output image data on a display device.

9. The image processing method according to claim 8, wherein the non-linear scaling process includes a process of horizontally altering a size of the at least one of the image blocks, which belongs to the central area, with an enlargement ratio which is less than 1, and a process of horizontally altering sizes of the other image blocks with enlargement ratios which increase toward ends of the input image data.

10. The image processing method according to claim 8, wherein the non-linear scaling process includes a process of horizontally altering a size of the at least one of the image blocks, which belongs to the central area, with an enlargement ratio which is an integral multiple of $1/2^n$ and is less than 1, n being an integer.

11. The image processing method according to claim 8, wherein the non-linear scaling process includes a process of horizontally altering sizes of the plurality of image blocks with a plurality of enlargement ratios having values, each of which is an integral multiple of $1/2^n$, n being an integer.

12. The image processing method according to claim 8, wherein the non-linear scaling process includes a pixel interpolation process for horizontally altering sizes of the plurality of image blocks with a plurality of enlargement ratios having values, each of which is an integral multiple of $1/2^n$, n being an integer, and said executing the non-linear scaling process includes executing the pixel interpolation process by an arithmetic process including bit shift and multiplication of integers.

Appendix: Obara (Ex. 1006)

- Ex. 1006, ¶¶88-90, cls. 1-12

[0088] In the non-linear scaling process by the software, it is possible to execute only the process of horizontally reducing the central area of the image data. In this case, too, it is possible to prevent horizontal enlargement distortion of a central area of the enlarged image.

[0089] The present embodiment is directed, by way of example, to the case in which image data, which is included in broadcast program data, is enlarged and displayed. The image processing method of this embodiment, however, is also applicable to the case in which image data or the like, which is read out of a storage medium such as a DVD, is enlarged and displayed. Furthermore, the non-linear scaling module 215 may be realized by hardware.

[0090] While certain embodiments of the inventions have been described, these embodiments have been presented by way of example only, and are not intended to limit the scope of the inventions. Indeed, the novel methods and systems described herein may be embodied in a variety of other forms; furthermore, various omissions, substitutions and changes in the form of the methods and systems described herein may be made without departing from the spirit of the inventions. The accompanying claims and their equivalents are intended to cover such forms or modifications as would fall within the scope and spirit of the inventions.

What is claimed is:

1. An information processing apparatus comprising:

an image processing unit which executes a non-linear scaling process of vertically dividing input image data having a first size into a plurality of image blocks and horizontally reducing at least one of the image blocks, which belongs to a central area of the input image data; and

a display controller which enlarges the input image data, which is subjected to the non-linear scaling process, by linear scaling, generates output image data having a second size greater than the first size, and displays the generated output image data on a display device.

2. The information processing apparatus according to claim 1, wherein the non-linear scaling process includes a process of horizontally altering a size of the at least one of the image blocks, which belongs to the central area, with an enlargement ratio which is less than 1, and a process of horizontally altering sizes of the other image blocks with enlargement ratios which increase toward ends of the input image data.

3. The information processing apparatus according to claim 1, wherein the non-linear scaling process includes a process of horizontally altering a size of the at least one of the image blocks, which belongs to the central area, with an enlargement ratio which is an integral multiple of $1/2^n$ and is less than 1, n being an integer.

4. The information processing apparatus according to claim 1, wherein the non-linear scaling process includes a process of horizontally altering sizes of the plurality of image blocks with a plurality of enlargement ratios having values, each of which is an integral multiple of $1/2^n$, n being an integer.

5. The information processing apparatus according to claim 1, wherein the non-linear scaling process includes a pixel interpolation process for horizontally altering sizes of the plurality of image blocks with a plurality of enlargement

ratios having values, each of which is an integral multiple of $1/2^n$, the n is an integer, and the image processing unit executes the pixel interpolation process by an arithmetic process including bit shift and multiplication of integers.

6. The information processing apparatus according to claim 1, wherein the display controller includes a filtering process unit which executes a filtering process of smoothing the generated output image data, and the output image data, which is subjected to the filtering process, is displayed on the display device.

7. The information processing apparatus according to claim 1, further comprising a receiving device which receives broadcast program data, wherein the input image data is the broadcast program data which is received by the receiving device.

8. An image processing method which processes image data using a display controller having a linear scaling function, comprising:

executing a non-linear scaling process of vertically dividing input image data having a first size into a plurality of image blocks and horizontally reducing at least one of the image blocks, which belongs to a central area of the input image data; and

sending the input image data, which is subjected to the non-linear scaling process, to the display controller, and causing the display controller to execute a process of enlarging the input image data, which is subjected to the non-linear scaling process, by linear scaling and generating output image data having a second size greater than the first size; and

displaying the generated output image data on a display device.

9. The image processing method according to claim 8, wherein the non-linear scaling process includes a process of horizontally altering a size of the at least one of the image blocks, which belongs to the central area, with an enlargement ratio which is less than 1, and a process of horizontally altering sizes of the other image blocks with enlargement ratios which increase toward ends of the input image data.

10. The image processing method according to claim 8, wherein the non-linear scaling process includes a process of horizontally altering a size of the at least one of the image blocks, which belongs to the central area, with an enlargement ratio which is an integral multiple of $1/2^n$ and is less than 1, n being an integer.

11. The image processing method according to claim 8, wherein the non-linear scaling process includes a process of horizontally altering sizes of the plurality of image blocks with a plurality of enlargement ratios having values, each of which is an integral multiple of $1/2^n$, n being an integer.

12. The image processing method according to claim 8, wherein the non-linear scaling process includes a pixel interpolation process for horizontally altering sizes of the plurality of image blocks with a plurality of enlargement ratios having values, each of which is an integral multiple of $1/2^n$, n being an integer, and said executing the non-linear scaling process includes executing the pixel interpolation process by an arithmetic process including bit shift and multiplication of integers.

Appendix: First Katsaggelos Transcript (Ex. 2026)

Appendix: First Katsaggelos Transcript (Ex. 2026)

- Ex. 2026, p. 14

1 many examples, and then -- Yeah, just three examples out
2 of the many examples that one can present.

3 Q You've mentioned in the deposition that Meulen
4 discloses that the wireless device 100 can include a
5 digital camera, correct?

6 A Yes, it did state that -- in Meulen that the
7 wireless device may comprise a camera, yes.

8 Q There's no discussion in Meulen about how the
9 aspect ratio of that camera compares to the display on
10 Meulen, is there?

11 MS. NGUYEN: Objection, form.

12 THE WITNESS: Meulen discloses that the wireless
13 device may include the display as well and discloses
14 ways to improve the performance of the device to display
15 content from the media source. So a disclosure is
16 there, but the device is capable of changing the aspect
17 ratio of the media.

18 MR. SCHMIDT: Q But you would agree with me that
19 if a digital camera on a wireless device has an image
20 sensor with an aspect ratio equal to the aspect ratio of
21 the display, then there's no need to change the aspect
22 ratio, right?

23 A Well, this is a hypothetical, and it might be
24 the case that there's no need for that. But Meulen

Appendix: First Katsaggelos Transcript (Ex. 2026)

- Ex. 2026, p. 15

1 discloses that the capability of the wireless device is
2 there to perform this conversion of the aspect ratio.

3 Q But Meulen doesn't disclose that -- the
4 capabilities needed in order to display an image
5 captured by the digital camera that's part of the
6 wireless device; you'd agree with that, correct?

7 MS. NGUYEN: Objection, form.

8 THE WITNESS: Meulen discloses ways to do this
9 transformation. So that's the main -- the main thesis
10 of Meulen's disclosure. So it has the capability to do
11 this aspect ratio conversion of the images acquired by
12 the camera and the display device.

13 MR. SCHMIDT: Q But I'm asking you a very specific
14 question. Meulen doesn't teach that the capability of
15 changing aspect ratios is necessary for displaying
16 digital images captured by the digital camera that's
17 part of the wireless device, right?

18 A Meulen discloses improving the performance of
19 the wireless device to display content from a media
20 source. The media source being the camera is
21 therefore -- That's what Meulen is teaching us, how to
22 change aspect ratio data supplied by the camera in order
23 to be displayed on the display of the phone.

24 Q What's the aspect ratio of the image sensor of

Appendix: First Katsaggelos Transcript (Ex. 2026)

- Ex. 2026, p. 16

1 the digital camera on Meulen?

2 A It may or may not be disclosed, but -- the exact
3 aspect ratio, but, again, the capability's there of
4 Meulen to improve the performance of the wireless device
5 display content of the media source by performing this
6 aspect ratio conversion.

7 Q Now, the display on Meulen is a 1 by 1 aspect
8 ratio, correct?

9 A Can you point me to where this is disclosed in
10 Meulen?

11 Q Column 5, at line 62, reading, in this
12 implementation, the display 138 may have an aspect ratio
13 of 1 by 1, with a resolution of 320 by 320. Do you see
14 that?

15 A Yes, I do. So that's a possible aspect ratio.
16 It doesn't have to be that. But as stated here in this
17 one implementation, yes, it is 1 to 1.

18 Q And, in fact, all the embodiments in Meulen
19 disclose a 1 by 1 aspect ratio display; isn't that
20 right?

21 A Well, I see implementation. I see embodiment.
22 But it doesn't mean that that's the only aspect ratio of
23 the display device that is accommodated by Meulen if we
24 look at the claims.

Page 16

Appendix: First Katsaggelos Transcript (Ex. 2026)

- Ex. 2026, p. 17

1 Q Let's just focus on the 1 by 1 implementation,
2 the implementation where Meulen has a 1 by 1 aspect
3 ratio display. Are you with me?

4 A Okay.

5 Q If Meulen also had an image sensor that was 1 by
6 1 aspect ratio, then you would agree that there would be
7 no purpose to the '413 aspect ratio functionality being
8 applied for displaying that image on Meulen's display,
9 correct?

10 MS. NGUYEN: Objection, form.

11 THE WITNESS: Well, to start with, most cameras
12 don't have a 1 by 1 aspect ratio. But if you think
13 about just about one implementation out of many, that's
14 not -- I mean, what Meulen discloses is how to perform
15 aspect ratio conversion, and just amount of
16 implementation -- any of the disclosures here of the
17 '413 or this or that would be needed. We are talking
18 about the situation where there is a need to change the
19 aspect ratio.

20 MR. SCHMIDT: Q Turn to column 13 of Meulen. That
21 would be column 13 at line 5. Meulen reads, in various
22 embodiments the processor 510 may be arranged to receive
23 information, such as media source content, through a
24 communication interface 512. Do you see that?

Page 17

Appendix: First Katsaggelos Transcript (Ex. 2026)

- Ex. 2026, p. 18

1 A Yes, I do.

2 Q And the communication interface 512 would be an
3 implementation where the media source content is being
4 received from an external source, correct?

5 A External source is not there. It's just
6 receiving information from a media source. That's what
7 it reads there, right?

8 Q But receiving information through a
9 communication interface implies an external source; you
10 would agree with that, right?

11 MS. NGUYEN: Objection, form.

12 THE WITNESS: It says that the processor may be
13 arranged to receive this information. And the
14 communications interface, that's always implying
15 external source. It's an interface where you connect
16 the processor to an input.

17 MR. SCHMIDT: Q Okay. Let's turn back to your
18 declaration, which is Exhibit 1, and I want to turn to
19 paragraph 253.

20 A Yes, I'm there.

21 Q And in this portion of the declaration you were
22 analyzing claim element 2C of the '413 patent. Do you
23 see that?

24 A Yes, I do.

Page 18

Appendix: First Katsaggelos Transcript (Ex. 2026)

- Ex. 2026, p. 19

1 Q That's the determining a portion requirement,
2 correct?

3 A Correct.

4 Q At paragraph 253 you write, Meulen discloses the
5 central display region 140-3 may comprise 200 pixels and
6 require the media source content to undergo no image
7 compression, and it continues from there. Do you see
8 that?

9 A Yes, I do.

10 Q Do I understand that it's your opinion that the
11 central display region 140-3 is the portion of element
12 2C of the '413 patent?

13 A Yes, in this example, this is the part of the
14 image that undergoes no compression and, therefore,
15 corresponds to the subset of the source pixels that are
16 then in the portion of the image.

17 Q And at the end of this sentence you cite to
18 Meulen at column 6, lines 27 through 31. Do you see
19 that?

20 A Yes, I do.

21 Q Okay. Let's turn to that citation in Meulen.
22 So Exhibit 3, column 6, lines 27 through 31.

23 In that passage of Meulen you're citing to the
24 central display region as described in table 1 of

Appendix: First Katsaggelos Transcript (Ex. 2026)

- Ex. 2026, p. 20

1 Meulen. Do you see that?

2 A Yes.

3 Q And in table 1 of Meulen, what table 1 is
4 describing is a pre-determined partition of the total
5 display area, correct?

6 MS. NGUYEN: Objection, form.

7 THE WITNESS: Yeah, it shows that there are five
8 regions there that -- again, it's an example of five
9 regions that you can output image. That's correct.

10 MR. SCHMIDT: Q And those five regions are all
11 allocated some number of pixels from the total display
12 width of Meulen, correct?

13 A Yeah. I mean, this is an example of five
14 regions. In other examples there may be seven or as
15 many regions as we want. And this is an example that
16 shows how 480 source pixels can be marked in the target
17 region pixels.

18 Q And in this approach of partitioning the display
19 into different regions, the regions are all pre-assigned
20 some compression ratio. Do you see that?

21 MS. NGUYEN: Objection, form.

22 THE WITNESS: Yeah, in this embodiment of the many
23 embodiments, it demonstrates how this change in aspect
24 ratio can take place.

Page 20

Appendix: Palm Treo 700w Manual (Ex. 1021)

Appendix: Palm Treo 700w Manual (Ex. 1021)

- Ex. 1021, p. 1



Using Your Palm® Treo™ 700w Smartphone

Appendix: Palm Treo 700w Manual (Ex. 1021)

- Ex. 1021, p. 5

Table of Contents

Welcome.....	1
What's in the box?.....	2
What you need to use your smartphone.....	3
Chapter 1: Setting up your smartphone and computer.....	5
Palm® Treo™ 700w smartphone overview.....	5
Installing the battery	8
Charging the battery	9
Turning your smartphone on/off	11
Making your first call	12
Setting up your computer	15
Chapter 2: Moving around on your smartphone	19
Moving around the screen	19
Using the keyboard	26
Opening and closing applications	30
Using your Today screen.....	31
Synchronizing contacts and other info.....	32
Battery life.....	34
Where to learn more	35

Appendix: Palm Treo 700w Manual (Ex. 1021)

- Ex. 1021, p. 6

Chapter 3: Using Your Phone..... 37

 Phone overview.....37

 Making calls.....41

 Receiving calls.....46

 Using voicemail.....47

 Managing active calls 48

 More ways to manage calls.....50

 Defining speed dial buttons 54

 Using a phone headset.....58

Chapter 4: Staying Organized..... 61

 Contacts 61

 Calendar 63

 Tasks 71

 Notes 74

 Calculator.....77

Chapter 5: Using Email and Other Wireless Services..... 79

 Sending and receiving messages and email 79

 Using Pocket MSN 103

 Browsing the web 104

 Connecting to devices with Bluetooth® wireless technology 110

 Beaming information with IR 114

Appendix: Palm Treo 700w Manual (Ex. 1021)

- Ex. 1021, p. 7

Chapter 6: Playing Media Files	117
Pictures & Videos	117
Windows Media Player Mobile	127
Chapter 7: Staying Productive	133
Word Mobile	133
PowerPoint Mobile	140
Excel Mobile	142
Chapter 8: Managing Files and Applications	155
Using Find	155
Installing applications	157
Removing applications	160
Setting synchronization options	161
Sharing info	165
Using expansion cards	166
Chapter 9: Customizing Your Smartphone	173
Today screen settings	173
Phone settings	175
System sound settings	180
Display and appearance settings	183
Application settings	185
Locking your smartphone and info	189
Using Auto-Keyguard and touchscreen lockout	190

Appendix: Palm Treo 700w Manual (Ex. 1021)

- Ex. 1021, p. 8

System settings	194
Connection settings	201
Chapter 10: Troubleshooting	211
Transferring info from another device	211
Trouble installing the desktop software?	212
Resetting your smartphone	213
Performance	217
Screen	218
Network connection	219
Synchronization (ActiveSync® software)	224
Email	230
Web	234
Camera	235
Third-party applications	236
Making room on your smartphone	238
Voice quality	239
Glossary	241
Regulatory Information	245
FCC	247
Specifications	253
Index	257

viii

- Ex. 1021, p. 125



Playing Media Files

Pictures & Videos

Your Palm® Treo™ 700w smartphone comes with an easy-to-use, built-in, 1.3-megapixel camera with 2x digital zoom (camera not included with all smartphones). You can use the camera to take and view pictures and videos and send them to your friends and family. To add a personal touch to your smartphone, use your pictures as your Today screen background, and as caller ID images, or use your videos to create video ringtones.

Even if your smartphone does not have a camera, you can still receive and view pictures and videos, and send pictures and videos as attachments or multimedia messages. You can also personalize your smartphone by using a picture as a background or by adding a picture to a contact; see "Assigning a picture and ringtone ID to a contact" on page 176 for details). You can also move pictures and videos to a computer by synchronizing your smartphone with your computer.

Taking a picture

By default, pictures are stored in the My Pictures folder on your smartphone. If you want to store your pictures on an expansion card, see "Customizing your Camera settings" on page 124 to change where pictures are stored. For information on accessing your pictures on your computer, see "Viewing pictures and videos on your computer" on page 126.

+ Did You Know?

Pictures are captured and stored in 16-bit color, JPG format. Resolution settings range from 1280 x 1024 to 160 x 120 pixels (1.3-megapixels to VGA). You can change the default setting. See "Customizing your Camera settings" on page 124 for details.

Appendix: Palm Treo 700w Manual (Ex. 1021)

- Ex. 1021, p. 126

CHAPTER 6

Playing Media Files

* Tip

If you see a camcorder icon below the preview image, video mode is on. To turn on the still camera, press Menu and select Still Mode.

* Tip

You can also adjust the zoom and brightness by pressing Menu and selecting Zoom or Brightness.

+ Did You Know?

Videos are captured and stored in 3G2 format. Video resolution settings range from 352 x 288 to 176 x 144 pixels.

+ Did You Know?

After three minutes of inactivity when previewing an image, the camera goes into standby mode. Press any key or tap the screen to return to the image preview.

1. Press Start  and select Pictures & Videos.
2. Select Camera .
3. Adjust the position of your smartphone until you see the subject you want to photograph on the screen.
4. (Optional) Adjust any of the following:
 - **Zoom:** Press Up  to zoom in or Down  to zoom out.
 - **Brightness:** Press Right  to increase the brightness or Left  to decrease the brightness.
 - **Resolution:** Press Menu  (right action key) and select Resolution to change the image quality.
5. (Optional) Set a 5-second timer: Press Menu  (right action key), select Mode, and then select Timer.
6. Press Center  to capture the picture or start the timer.
7. Hold your smartphone still until the picture renders, and then do one of the following:
 - Press Camera  (left action key) to take another picture.
 - Press OK  to return to the Thumbnails View.



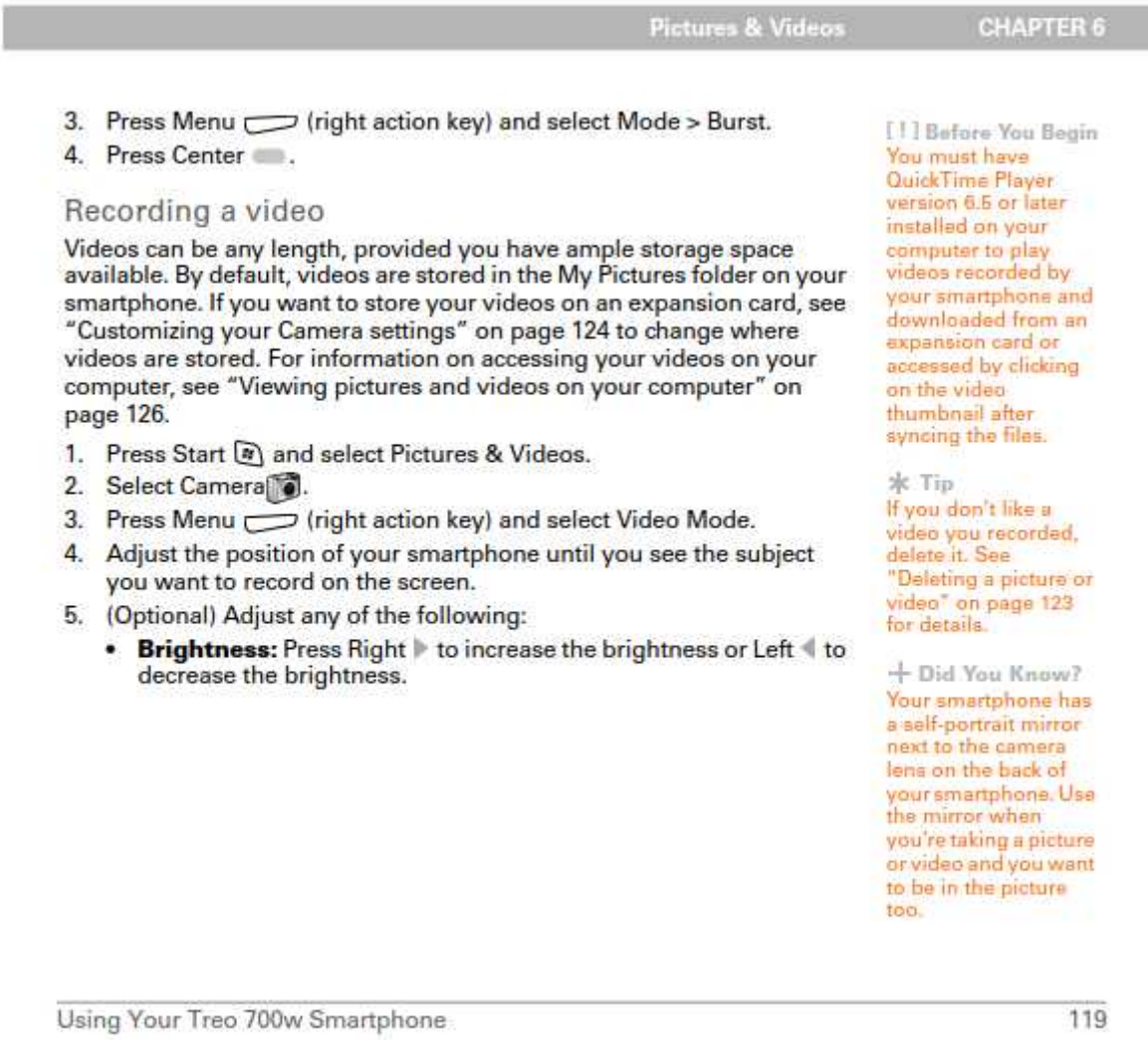
Taking pictures in burst mode

Burst mode takes five pictures in quick succession, with one button press.

1. Press Start  and select Pictures & Videos.
2. Press Camera  (left action key).

Appendix: Palm Treo 700w Manual (Ex. 1021)

- Ex. 1021, p. 127



Appendix: Palm Treo 700w Manual (Ex. 1021)

- Ex. 1021, p. 128

CHAPTER 6

Playing Media Files

* Tip

To view pictures or videos in a different folder, select the Show list in the upper-left, and then select the album you want to view.

* Tip

You can also press Right and Left on the 5-way to move forward or backward through the slides.

+ Did You Know?

You cannot send copyrighted pictures or videos that appear with a Lock icon in the Thumbnail View.

* Tip

You can send pictures and videos as email attachments.

- **Resolution:** Press Menu  and select Quality to change the video resolution.

6. Press Center  to start recording.
7. When you're finished recording, press Stop  (left action key) or Center  to stop recording.
8. (Optional) To review the video in Windows Media Player Mobile, press Thumbnails  (left action key), highlight the video, and then press Center . Press OK  to return to the Thumbnail View.



Viewing a picture or video

1. Press Start  and select Pictures & Videos.
2. Select the picture or video you want to view. For more info on viewing videos, see "Playing media files on your smartphone" on page 128.
3. Press OK  to return to the Thumbnail View.

Viewing a slide show

1. Press Start  and select Pictures & Videos.
2. Press Menu  (right action key) and select Play Slide Show. The slide show plays automatically.
3. Press Center  to display the slide show toolbar, which you can use to Rotate , Play , Pause , Reverse , Forward , and Stop  the slide show.

120

Using Your Treo 700w Smartphone

Insta360 Exhibit 1021 Page 128
IPR2024-00984 Insta360 v. GoPro

Appendix: Palm Treo 700w Manual (Ex. 1021)

- Ex. 1021, p. 129

Sending a picture or video

You can send a picture or video to other picture-enabled mobile smartphones or to an email address.

1. Press Start  and select Pictures & Videos.
2. Highlight the picture or video you want to send.
3. Press Menu  (right action key) and select Send.
4. Select the email or MMS account you want to use to send the picture or video.
5. When the Messaging application opens, address and send the message. (See "Creating and sending a multimedia message" on page 81 for details.)

Creating a video ringtone

You can save a video that you record as a ringtone.

1. Press Start  and select Pictures & Videos.
2. Highlight the video you want to use as a ringtone.
3. Press Menu  (right action key) and select Save to Contact Ring Tone.
4. Select the contact to whom you want to assign the ringtone.

Organizing pictures and videos

You can move or copy pictures and videos to other folders or between your smartphone and an expansion card.

1. Press Start  and select Pictures & Videos.
2. Highlight the picture or video you want to move or copy.

Appendix: Palm Treo 700w Manual (Ex. 1021)

- Ex. 1021, p. 130

CHAPTER 6

Playing Media Files

3. Do one of the following:
 - To move the picture or video to another location, press Menu  and select Edit > Cut.
 - To keep the picture or video in two locations, press Menu  and select Edit > Copy.
4. Select the Show list in the upper-left, and select the location where you want to place the picture or video you selected in step 3.
5. Press Menu  (right action key) and select Edit > Paste.

You can also view a specific folder or expansion card and arrange the pictures and videos by name, date, or size.

1. Press Start  and select Pictures & Videos.
2. Select the Show list in the upper-left, and then select the folder you want to view.
3. Select the Sort By list in the upper-right, and then select the sort method: Name, Date, or Size.

Appendix: Palm Treo 700w Manual (Ex. 1021)

- Ex. 1021, p. 131

Using a picture as the Today screen background

1. Press Start  and select Pictures & Videos.
2. Highlight the picture you want to use.
3. Press Menu  (right action key) and select Set as Today Background.
4. Select the Transparency level list and select the appropriate level. Use a higher percentage for a more transparent picture and a lower percentage for a more opaque picture.
5. Press OK  to return to the Thumbnail View.

Editing a picture

1. Press Start  and select Pictures & Videos.
2. Highlight the picture you want to edit.
3. Press Menu  (right action key) and select Edit.
4. Do any of the following:
 - To rotate a picture 90 degrees counterclockwise, select Rotate.
 - To crop a picture, press Menu  (right action key) and select Crop. Tap and drag the stylus to highlight the area to crop. Tap outside the box to stop cropping.
 - To adjust the brightness and contrast levels of a picture, press Menu  (right action key) and select AutoCorrect.

Deleting a picture or video

1. Press Start  and select Pictures & Videos.
2. Highlight the picture or video you want to delete.
3. Press Menu  (right action key) and select Delete.
4. Select Yes.

* Tip

To undo an edit, press Menu and select Undo. To cancel all unsaved edits you made to the picture, select Revert to Saved.

* Tip

For more extensive edits, just download your picture or video to your computer and edit it in your favorite graphics program. Then sync the picture or video back on your smartphone.

Appendix: Palm Treo 700w Manual (Ex. 1021)

- Ex. 1021, p. 132

CHAPTER 6

Playing Media Files

* Tip
If your smartphone does not have a built-in camera, the Camera and Video options do not apply to you.

Customizing your Camera settings

1. Press Start [] and select Pictures & Videos.
2. Press Menu [] (right action key) and select Options.
3. On the General tab, set any of the following options:
 - **Use this picture size:** Sets the size of pictures you send with the Messaging application.
 - **When rotating a picture, rotate 90 degrees:** Sets the direction in which pictures rotate.
4. Select the Slide Show tab and set any of the following options:
 - **During slide shows, optimize for viewing:** Sets whether pictures are optimized for portrait or landscape format during slide shows.
 - **Play screensaver when connected to my PC and idle for 2 minutes:** Sets whether the pictures in your My Pictures folder are used as a screensaver when your smartphone is connected to your computer and ActiveSync® software is not running.



Appendix: Palm Treo 700w Manual (Ex. 1021)

- Ex. 1021, p. 133

5. Select the Camera tab and set any of the following options:

- **Save files to:** Determines if pictures and videos are stored on your smartphone or on an expansion card.
- **Type filename prefix:** Assigns a name to a series of pictures to be captured, such as Seattle001, Seattle002, and so on.
- **Still image compression level:** Sets the default size for newly captured pictures.



6. Select the Video tab and set any of the following options:

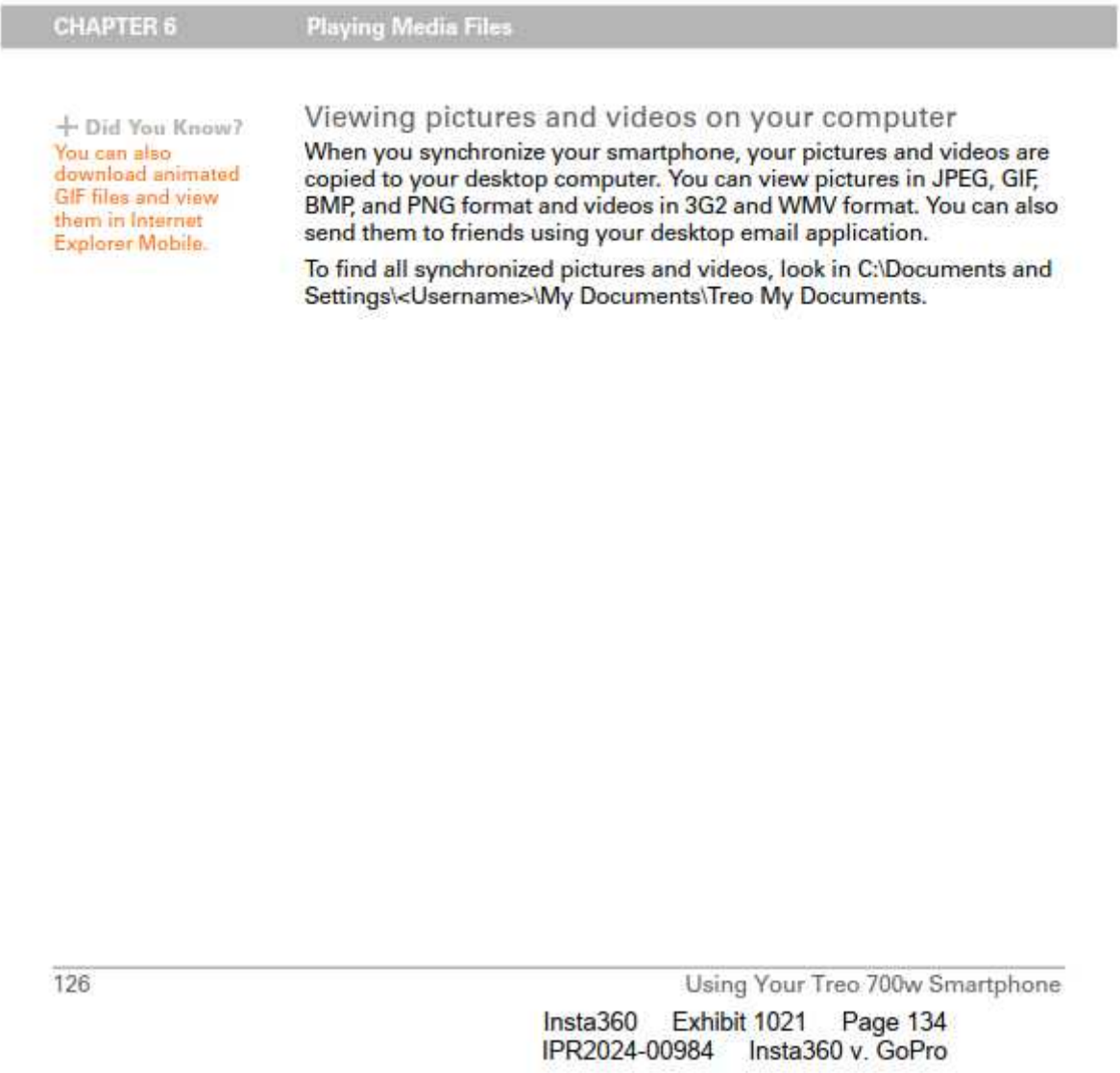
- **Include audio when recording video files:** Turns the microphone on and off so that you can record videos with or without sound.
- **Time limit for videos:** Limits the length of videos you record. You can also select the No limit option.



7. Press OK (OK) to return to the Thumbnail View.

Appendix: Palm Treo 700w Manual (Ex. 1021)

- Ex. 1021, p. 134



Appendix: Palm Treo 700w Manual (Ex. 1021)

- Ex. 1021, p. 135

Windows Media Player Mobile

CHAPTER 6

Windows Media Player Mobile

Windows Media Player Mobile can play music, audio, and video files that are stored on your smartphone or on an expansion card (sold separately) in any of the following file formats:

- WMA
- WMV
- MP3
- 3G2

You can listen to these music, audio, and video files through the speaker on the back of your smartphone or through a stereo headphone (3.5mm stereo headphone adapter or 2.5mm stereo headphone required, sold separately).

Transferring media files to your smartphone

Use the Sync feature in the desktop version of Windows Media Player 10 to transfer digital music, audio, video, and playlist files from your computer to an expansion card or your smartphone. Using Sync ensures that the files transfer correctly.

1. On your computer, open Windows Media Player 10.
2. (Optional) Insert a 32MB or larger expansion card into your smartphone.
3. Connect your smartphone to your computer with the USB sync cable.
4. When the Device Setup Wizard opens on your computer, click Automatic.
5. Check the Customize the playlists that will be synchronized box.

!!! Before You Begin
Do the following:

- You must have Windows Media Player 10 or later installed on your computer. See "My media files won't sync" on page 226.
- To sync media files with your computer, you must set the Media sync option. See "Selecting which info to sync" on page 161.

*** Tip**
For tips on using the desktop version of Windows Media Player, go to the Help menu in Windows Media Player 10 on your computer.

*** Tip**
Be patient; transferring media files to an expansion card can take several minutes.

Using Your Treo 700w Smartphone

127

Appendix: Palm Treo 700w Manual (Ex. 1021)

- Ex. 1021, p. 136

CHAPTER 6

Playing Media Files

* Tip

If you can't find a media file on your expansion card, update the library (see "Working with libraries" on page 129).

* Tip

To play a file that is not in a library, go to the Library screen, press Menu, and select Open File.

* Tip

To play a file from the web, press Menu on the Library screen, and then select Open URL. Select URL and enter the website address, or select History and select a site you've visited before.

* Tip

To repeat the current song, press Menu and select Repeat Song.

6. Select the playlists you want to sync.

7. Click Finish to begin the transfer. The next time you connect your smartphone to your computer while the desktop version of Windows Media Player 10 is running, synchronization starts automatically.

Playing media files on your smartphone

1. Press Start  and select Windows Media.

2. Press Menu  (right action key) and select Library.

3. Select the Library list in the upper-left, and then select the library you want to use.

4. Select the item you want to play (such as a song, album, or artist name).

5. Press Play  (left action key).

6. Use any of the following onscreen controls during playback:

Center  or  Plays the current file.

Center  or  Pauses the current file.

Left  Skips to the beginning of the current file or to the previous file.

Right  Skips to the next file.

 Sets the point from which playback begins. Tap and drag the slider to change the current position.

Volume button or  Increases the volume level.

Volume button or  Decreases the volume level.

 Turns the sound on or off.

 Displays a video using the full screen.

 Displays a website where you can find music and videos to play.

 Indicates the rating of the current file. Select the star to change the rating.

128

Using Your Treo 700w Smartphone

Insta360 Exhibit 1021 Page 136
IPR2024-00984 Insta360 v. GoPro

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

244

Appendix: Palm Treo 700w Manual (Ex. 1021)

- Ex. 1021, p. 137

Windows Media Player Mobile

CHAPTER 6

7. Press Menu  (right action key) and select any of the following during playback:

- **Library:** Displays the Library screen so you can select a different song to play.
- **Shuffle/Repeat > Shuffle:** Plays the Now Playing playlist in random order. A check appears next to this command when it is on.
- **Shuffle/Repeat > Repeat:** Plays the Now Playing playlist repeatedly. A check appears next to this command when it is on.
- **Stop:** Ends playback.

Working with libraries

A library represents each of the storage locations available to Windows Media Player Mobile, so you should have two libraries: My Device and My Storage Card. Each library contains links to the media files in that location. Windows Media Player Mobile usually updates the My Device library automatically, but you must manually update the My Storage Card library.

1. Press Start  and select Windows Media.

2. Press Menu  (right action key) and select Library.

3. Press Menu  (right action key) and select Update Library.

4. Wait for the files to be added, and then select Done.

+ Did You Know?

When you update a library, it automatically updates video files (WMV and 3G2) and audio files (WMA, MP3, OCP, and WAV).

* Tip

To delete an item from a library: Highlight the item and then press and hold Center on the 5-way to open the shortcut menu. Then select Delete from Library. Select Yes to confirm the deletion.

* Tip

If you don't see a media file that you added, manually update the library.

Appendix: Palm Treo 700w Manual (Ex. 1021)

- Ex. 1021, p. 138

CHAPTER 6

Playing Media Files

+ Did You Know?

You can also move a file in the Now Playing playlist by tapping and dragging it to a new position.

* Tip

If you move files between your smartphone and an expansion card, be sure to update your libraries, or you won't be able to see the files in their new location.

Working with playlists

A playlist is a list of media files that play in a specific order. You can use playlists to group audio files together or video files together for convenient playback. For example, in the desktop Player, you can create a playlist of upbeat songs for when you exercise and a playlist of soothing songs for a long flight. When you synchronize, your favorite playlists are automatically copied to your smartphone. Your playlists appear in your libraries (in the My Playlists category).

A temporary playlist, called Now Playing, appears on the Now Playing menu. It lists the currently playing file, as well as any files that are queued up to play next. You can add to, modify, or clear the files on the Now Playing playlist.

1. Go to the Now Playing screen:
 - If you are on the Library screen, select the Now Playing category.
 - If you are on the Playback screen, select Now Playing.
2. Do any of the following:
 - To move a file up or down one slot, highlight the file, and then select Move Up  or Move Down .
 - To add a file, select Add , press Menu  (right action key), and then select Queue Up.
 - To delete a file from the playlist, highlight the file, and then select Remove .



Appendix: Palm Treo 700w Manual (Ex. 1021)

- Ex. 1021, p. 139

- To view more info about a file, highlight the file and select Properties .
- To remove all items from the Now Playing playlist, press Menu  and select Clear Now Playing.

Customizing Windows Media Player Mobile

1. If necessary, go to the Playback screen by pressing OK  to close the current screen.
2. Press Menu  (right action key) and select Options.
3. On the Playback tab, set any of the following options:
 - **Show time as:** Sets whether the time remaining or time elapsed appears in the Playback screen.
 - **Pause playback while using another program:** Sets whether playback continues if you switch to another application.
 - **Resume playback after a phone call:** Sets whether playback continues after you finish a phone call.
4. Select the Video tab and set any of the following options:
 - **Play video in full screen:** Sets whether videos automatically play in full screen format.
 - **Scale to fit window:** Sets whether videos are automatically scaled to fit the Playback screen.



+ Did You Know?
You can also customize your smartphone so that pressing and holding the Side button on the side of your smartphone opens Windows Media Player Mobile. See "Reassigning buttons" on page 185 for details.

Appendix: Palm Treo 700w Manual (Ex. 1021)

- Ex. 1021, p. 140

CHAPTER 6

Playing Media Files

+ Did You Know?
If you close the Windows Media Player Mobile window, your music continues to play in the background.

5. Select the Network tab and set the following options:
 - **Protocol:** Enables and disables the available protocols. You must select at least one protocol. You can also set a UDP Port.
 - **Internet Connection Speed:** Sets the speed of your network connection, and sets whether you want the device to detect connection speed.
6. Select the Library tab and set whether you want to see the Library or Playback screen when you open Windows Media Player Mobile.
7. Select the Skins tab and select Previous or Next to set the player's background.
8. Select the Buttons tab to change any of the available button settings:
 - To assign a button, highlight the item you want to set, select Assign, and then press the button you want to use for that item.
 - To restore an item's factory setting, highlight the item and select Reset.
 - To unassign an item, highlight the item and select None.
9. Press OK.

