

IMAGE DISTORTION CORRECTION

BACKGROUND:

FIELD OF THE INVENTION:

[0001] Embodiments of the present invention relate to correcting image distortion.

RELATED ART:

[0002] A continual issue when dealing with cameras and other optical devices is the distortion introduced by the lens of the camera itself. Many different kinds of distortion can occur, and are familiar problems for camera designers and photographers alike.

[0003] Several approaches are traditionally used, when correcting distortion. In more expensive cameras, such as single-lens reflex (SLR) cameras, combinations of lenses are used in sequence, with each additional piece of glass often designed to reduce or eliminate a particular type of distortion. Less expensive cameras offer correspondingly fewer hardware fixes for the distortion introduced by their lenses, with integrated solutions, such as mobile phone cameras, having almost no inherent distortion correction.

[0004] Distortion can also be corrected after an image has been captured. Digital imagery, such as the pictures and video captured by digital cameras and camcorders, can be manipulated after the image has been taken, and the distortion introduced by the camera itself can be nearly completely eliminated.



159159



G00002

SUMMARY:

[0005] Methods and systems for reducing or eliminating distortion in an image are described. The approach generally involves determining the distortion introduced by a lens, and modifying a captured image to reduce that distortion. In one embodiment, the distortion information associated with a lens is determined. The distortion information is stored. A captured image taken by that lens is processed, with reference to the distortion information. By quantifying image distortion in this manner, embodiments of the present invention can be used to improve image quality even in camera systems that utilized a simple lens system.

[0006] Another embodiment is described for reducing lens distortion in a captured image. A system is described, having a buffer, for storing information, and a texturing engine, for generating pixel data as part of a graphics processing unit (GPU). The texturing engine is configured to access a correction image stored in the buffer. The texturing engine uses the correction image to generate a corrected image from the captured image.

[0007] Another embodiment is also described for warping a captured image to reduce distortion. This approach involves generating distortion correction information. The distortion correction information is stored as a warping texture. The warping texture is loaded into a GPU, e.g., in a texture engine, and the GPU is used to warp the captured image, with reference to the warping texture, to produce a modified image.

BRIEF DESCRIPTION OF THE DRAWINGS:

[0008] The accompanying drawings, which are incorporated in and form a part of this specification, illustrate embodiments of the invention and, together with the description, serve to explain the principles of the invention:

[0009] Figure 1 is a block diagram of an exemplary computer system upon which embodiments of the present invention may be implemented.

[0010] Figure 2 is an illustration of several types of distortion, upon which embodiments of the present invention may be applied.

[0011] Figure 3 is a flowchart of an exemplary method of correcting distortion in an image, in accordance with one embodiment.

[0012] Figure 4 is a flowchart of an exemplary method of determining distortion for a lens, in accordance with one embodiment.

[0013] Figure 5 is a flowchart of an exemplary method of generating a corrected image, in accordance with one embodiment.

[0014] Figure 6 is an illustration of corrected image generation, in accordance with one embodiment.

[0015] Figure 7 is a block diagram of another system for reducing distortion in captured images, in accordance with one embodiment.

DETAILED DESCRIPTION:

[0016] Reference will now be made in detail to several embodiments of the invention. While the invention will be described in conjunction with the alternative embodiment(s), it will be understood that they are not intended to limit the invention to these embodiments. On the contrary, the invention is intended to cover alternative, modifications, and equivalents, which may be included within the spirit and scope of the invention as defined by the appended claims.

[0017] Furthermore, in the following detailed description, numerous specific details are set forth in order to provide a thorough understanding of the claimed subject matter. However, it will be recognized by one skilled in the art that embodiments may be practiced without these specific details or with equivalents thereof. In other instances, well-known methods, procedures, components, and circuits have not been described in detail as not to unnecessarily obscure aspects and features of the subject matter.

[0018] Portions of the detailed description that follows are presented and discussed in terms of a method. Although steps and sequencing thereof are disclosed in figures herein (e.g., Figure 5) describing the operations of this method, such steps and sequencing are exemplary. Embodiments are well suited to performing various other steps or variations of the steps recited in the flowchart of the figure herein, and in a sequence other than that depicted and described herein.

[0019] Some portions of the detailed description are presented in terms of procedures, steps, logic blocks, processing, and other symbolic representations of operations on data bits that can be performed on computer memory. These descriptions and representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. A procedure, computer-executed step, logic block, process, etc., is here, and generally, conceived to be a self-consistent sequence of steps or instructions leading to a desired result. The steps are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated in a computer system. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like.

[0020] It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the following discussions, it is appreciated that throughout, discussions utilizing terms such as “accessing,” “writing,” “including,” “storing,” “transmitting,” “traversing,” “associating,” “identifying” or the like, refer to the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer system's registers and memories into other data similarly represented as physical quantities within the computer

system memories or registers or other such information storage, transmission or display devices.

[0021] Computing devices typically include at least some form of computer readable media. Computer readable media can be any available media that can be accessed by a computing device. By way of example, and not limitation, computer readable medium may comprise computer storage media and communication media. Computer storage media includes volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules, or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile discs (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by a computing device. Communication media typically embodies computer readable instructions, data structures, program modules, or other data in a modulated data signals such as a carrier wave or other transport mechanism and includes any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared, and other wireless media. Combinations of any of the above should also be included within the scope of computer readable media.

[0022] Some embodiments may be described in the general context of computer-executable instructions, such as program modules, executed by one or more computers or other devices. Generally, program modules include routines, programs, objects, components, data structures, etc., that perform particular tasks or implement particular abstract data types. Typically the functionality of the program modules may be combined or distributed as desired in various embodiments.

[0023] Although embodiments described herein may make reference to a CPU and a GPU as discrete components of a computer system, those skilled in the art will recognize that a CPU and a GPU can be integrated into a single device, and a CPU and GPU may share various resources such as instruction logic, buffers, functional units and so on; or separate resources may be provided for graphics and general-purpose operations. Accordingly, any or all of the circuits and/or functionality described herein as being associated with GPU could also be implemented in and performed by a suitably configured CPU.

[0024] Further, while embodiments described herein may make reference to a GPU, it is to be understood that the circuits and/or functionality described herein could also be implemented in other types of processors, such as general-purpose or other special-purpose coprocessors, or within a CPU.

BASIC COMPUTING SYSTEM

[0025] Referring now to Figure 1, a block diagram of an exemplary computer system 112 is shown. It is appreciated that computer system 112 described herein illustrates an exemplary configuration of an operational platform upon which embodiments may be implemented to advantage. Nevertheless, other computer systems with differing configurations can also be used in place of computer system 112 within the scope of the present invention. That is, computer system 112 can include elements other than those described in conjunction with Figure 1. Moreover, embodiments may be practiced on any system which can be configured to enable it, not just computer systems like computer system 112. It is understood that embodiments can be practiced on many different types of computer system 112. System 112 can be implemented as, for example, a desktop computer system or server computer system having a powerful general-purpose CPU coupled to a dedicated graphics rendering GPU. In such an embodiment, components can be included that add peripheral buses, specialized audio/video components, IO devices, and the like. Similarly, system 112 can be implemented as a handheld device (e.g., cellphone, etc.) or a set-top video game console device such as, for example, the Xbox®, available from Microsoft Corporation of Redmond, Washington, or the PlayStation3®, available from Sony Computer Entertainment Corporation of Tokyo, Japan. System 112 can also be implemented as a "system on a chip", where the electronics (e.g., the components 101, 103, 105, 106, and the like) of a computing device are wholly contained within a single integrated circuit die. Examples include a hand-held instrument with a display, a car navigation system, a portable entertainment system, and the like.

[0026] Computer system 112 comprises an address/data bus 100 for communicating information, a central processor 101 coupled with bus 100 for processing information and instructions; a volatile memory unit 102 (e.g., random access memory [RAM], static RAM, dynamic RAM, etc.) coupled with bus 100 for storing information and instructions for central processor 101; and a non-volatile memory unit 103 (e.g., read only memory [ROM], programmable ROM, flash memory, etc.) coupled with bus 100 for storing static information and instructions for processor 101. Moreover, computer system 112 also comprises a data storage device 104 (e.g., hard disk drive) for storing information and instructions.

[0027] Computer system 112 also comprises an optional graphics subsystem 105, an optional alphanumeric input device 106, an optional cursor control or directing device 107, and signal communication interface (input/output device) 108. Optional alphanumeric input device 106 can communicate information and command selections to central processor 101. Optional cursor control or directing device 107 is coupled to bus 100 for communicating user input information and command selections to central processor 101. Signal communication interface (input/output device) 108, which is also coupled to bus 100, can be a serial port. Communication interface 108 may also include wireless communication mechanisms. Using communication interface 108, computer system 112 can be communicatively coupled to other computer systems over a communication network such as the Internet or an intranet (e.g., a local area network), or

can receive data (e.g., a digital television signal). Computer system 112 may also comprise graphics subsystem 105 for presenting information to the computer user, e.g., by displaying information on an attached display device 110, connected by a video cable 111. In some embodiments, graphics subsystem 105 is incorporated into central processor 101. In other embodiments, graphics subsystem 105 is a separate, discrete component. In other embodiments, graphics subsystem 105 is incorporated into another component. In other embodiments, graphics subsystem 105 is included in system 112 in other ways.

CORRECTING DISTORTION THROUGH THE USE OF THE TEXTURING ENGINE

[0028] Graphics processing units (GPUs) are increasingly common in many electronic devices. Dedicated computers, such as laptops or desktops, will almost always include a GPU, either as a discrete processor, or integrated with other functionality within the system. GPU functionality can be found in a much broader range of applications, however. Handheld devices, such as personal digital assistants (PDAs), mobile phones and smart phones, and even some cameras may include one or more GPUs to speed the display of information, or to provide additional functionality.

[0029] GPUs, and in particular pixel shaders, are ideally suited to the manipulation of images. In the embodiments that follow, approaches are described that harness the functionality of pixel shaders to manipulate images on a per pixel basis. In particular, in some embodiments, pixel shaders are used to correct distortion in a digital image, such as the distortion introduced by a camera lens.

DISTORTION TYPES

[0030] Many different types of distortion may be introduced, when capturing an image. Causes of distortion may vary, across different embodiments. In some embodiments, for example, distortion is introduced as a result of differences in focal length through different portions of the camera lens or due to imperfections in the optics' manufacture or design. These are often referred to as *geometric distortion*. In another embodiment, distortion may result from a camera having different focal lengths across different wavelengths of light, e.g., the camera has different focal lengths for red, green, and blue light. Such distortions caused by differing wavelengths of light are often referred to as *chromatic distortion*. Similarly, the effect that distortion will have upon an image may vary, across different types of distortion and across different embodiments.

[0031] With reference now to Figure 2, several common distortion types are illustrated. While several types of distortion are illustrated, it is understood that embodiments are well suited to applications involving many different types of distortion. Figure 2 is intended to be illustrative, not exhaustive. Image 200 depicts a picture of an undistorted grid of lines, intersecting at right angles. Image 210 shows the same grid, subjected to "pincushion" distortion. Image 220 shows the grid of image 200, subjected to "barrel" distortion. Image 250 depicts 3 cylindrical objects, viewed from above. Image 260 shows the same objects, subjected to "relief" distortion, wherein objects directly below the camera lens will have only their tops visible, while all other objects will appear to lean away from the center of the image such that their tops and sides are visible. Additional,

important lens distortions include but are not limited to: perspective distortion, caused by misalignment of the focal plane relative to the chief ray's axis; and asymmetric distortions, due to "ripples" in the focal or sensor plane or due to gross imperfection in the lens manufacture.

[0032] Distortion is not limited to any particular application. Distortion effects occur in video applications, much as they occur in still image applications.

CORRECTING DISTORTION

[0033] In some embodiments, correcting distortion in an image can occur at any time after the image has been captured. For example, in some embodiments, e.g., where a digital camera is used, distortion is corrected when the image is first taken, and before it is stored. In another embodiment, distortion may be corrected sometime after the image is captured, e.g., before or after some other image processing occurs. In some embodiments, distortion correction can occur at any point where pixel data is available for the image.

[0034] Further, it is understood that embodiments are not limited to correction of still images. For example, in one embodiment, e.g., involving a digital camcorder, distortion is corrected as a video is captured, e.g., frame by frame. In another embodiment, distortion is corrected after a video has been captured, e.g., before or after some other video processing operation occurs. In some embodiments, distortion correction can occur at any point where pixel data is available for a frame of a video.

[0035] As noted above, many different types of distortion may occur. In some embodiments, distortion is introduced by the camera lens. In other embodiments, distortion is introduced by other factors, e.g., the sensor used to capture data. It is understood that embodiments are well suited to applications involving many different types and sources of distortion. While the exemplary embodiments presented below focus on correcting specific types of distortion, e.g., barrel or pincushion distortion, it is understood that other embodiments may be utilized to correct other types of distortion, e.g., chromatic distortion.

[0036] With reference now to Figure 3, a flowchart 300 of an exemplary method of correcting distortion in an image is depicted, in accordance with one embodiment. Although specific steps are disclosed in flowchart 300, such steps are exemplary. That is, embodiments of the present invention are well suited to performing various other (additional) steps or variations of the steps recited in flowchart 300. It is appreciated that the steps in flowchart 300 may be performed in an order different than presented, and that not all of the steps in flowchart 300 may be performed.

[0037] In step 310, the distortion introduced by a lens is determined. In different embodiments, the step is accomplished in different ways. For example, in one embodiment, the lens being tested is used to capture an image of a known reference picture, such as a rectangular grid like image 200. The captured reference image can then be compared to the known reference picture, e.g., on a pixel per pixel basis. In some

embodiments, this comparison is used to generate an equation describing the distortion introduced by the lens. In other embodiments, this comparison is used to generate a distortion mapping, e.g., an image map that describing how to manipulate the captured reference image, in order to reduce or eliminate the distortion introduced by the lens.

[0038] In different embodiments, distortion information may be collected in different ways. For example, a single two-dimensional distortion map may be generated, e.g., using horizontal and vertical values to individually address every pixel. In another embodiment, several separable one-dimensional "images" may be used. In another embodiment, a one-dimensional radial image may be used.

[0039] In some embodiments, step 310 may be repeated for multiple types of distortion, and/or multiple testing scenarios. For example, in some embodiments, distortion may vary with the focal distance of the lens, such that one equation governs distortion for pictures of close range objects, while a different equation governs distortion for pictures of objects further away from the lens. Further, different distortion may be introduced, depending upon the wavelength of the light, e.g., the color, or upon the general lighting conditions.

[0040] With reference to step 320, the distortion information for that lens is stored. In different embodiments, the storage of the distortion information will vary. For example, in one embodiment, distortion information is stored as a texture or image, such that a

texturing engine can use the distortion information to warp an image and reduce or eliminate distortion.

[0041] Further, in some embodiments, the information stored may vary. In some embodiments, e.g., an embodiment where the equation that governs the distortion for the lens is calculated, the equation is stored. In another embodiment, e.g., where a distortion map is generated, absolute distortion correction information for each pixel may be stored. In one such embodiment, the distortion map corrects for distortion by moving a pixel from one absolute location, in the distorted image, to another absolute location, in the corrected image. In another embodiment, a distortion map is used to store relative location information. For example, the distortion map may indicate that the current pixel should be moved to another location, by specifying a relative distance and/or direction to translate the pixel, rather than providing an absolute location.

[0042] With reference to step 330, a captured image is processed to reduce distortion introduced by the lens. In different embodiments, this step may occur at different times. For example, in one embodiment, distortion correction occurs as soon as an image is captured, e.g., when the data is received from the sensor, and before it is stored. In other embodiments, distortion correction occurs before or after some other image processing.

[0043] In some embodiments, captured images are processed on a per pixel basis. For each pixel, the distortion correction information for that pixel is retrieved, and the distortion correction information and captured image are used to generate a corrected

image. In one embodiment, as described in greater detail below, a pixel shader in a texturing engine is configured to generate a corrected pixel. In one such embodiment, the pixel shader retrieves distortion correction information from the distortion correction image, and uses the distortion correction information to retrieve a pixel from the captured image. The pixel shader outputs a corrected pixel, e.g., by outputting the retrieved pixel in a corrected location in the captured image.

DETERMINING DISTORTION

[0044] In different embodiments, different approaches are utilized to determine the distortion in a picture introduced by the equipment used to capture the picture. In some embodiments, the distortion introduced by a lens is determined. In one such embodiment, this determination is performed by comparing an undistorted reference image with the image captured by the lens.

[0045] With reference now to Figure 4, a flowchart 400 of an exemplary method of determining distortion for a lens is depicted, in accordance with one embodiment. Although specific steps are disclosed in flowchart 400, such steps are exemplary. That is, embodiments of the present invention are well suited to performing various other (additional) steps or variations of the steps recited in flowchart 400. It is appreciated that the steps in flowchart 400 may be performed in an order different than presented, and that not all of the steps in flowchart 400 may be performed.

[0046] With reference now to step 410, the lens being tested is used to take a picture of a known reference image. In different embodiments, the nature of the reference image may vary. For example, when determining barrel or pincushion distortion, it may be useful to utilize a grid image, such as image 200. In another embodiment, where color distortion is being determined, it may be useful to utilize several images, so as to test different colors and wavelengths of light. Moreover, in some embodiments, the same lens may be used to take pictures of multiple reference images, so as to test for the same type of distortion under differing circumstances, e.g., under different lighting conditions, or at different distances from the image.

[0047] With reference to step 420, correction, or calibration, information for the lens being tested is gathered. In some embodiments, the lens is calibrated by correcting distortion present in the captured image. In one such embodiment, the captured image is corrected "by hand," e.g., a user manipulates the pixels in the captured image, with reference to the reference image, to correct for distortion introduced by the lens. In other embodiments, the captured image is corrected in other ways, e.g., the captured image is manipulated by an off-line program, with reference to the reference image, to correct the distortions introduced by the lens.

[0048] With reference to step 430, the calibration information for the lens is stored. In some embodiments, this calibration information is stored as one or more textures, which a texturing engine can then use to correct distortions introduced by the lens in the future images. As noted previously, the nature of the calibration information may vary, across

different embodiments. For example, in some embodiments, calibration information consists of absolute pixel locations, e.g., to indicate where a particular pixel in the captured image should be located in the corrected image. In some other embodiments, calibration information consists of relative pixel locations, e.g., information to indicate that the current pixel should be moved to a new location, relative to its current location. Moreover, in different embodiments, calibration information may comprise different types of images, e.g., a two-dimensional image, or multiple one-dimensional separable images, or a single one-dimensional radial image.

GENERATING A CORRECTED IMAGE

[0049] In some embodiments, calibration or correction information can be used by a texturing engine to program one or more pixel shaders, in order to generate a corrected version of an image. As noted previously, in different embodiments, the nature of the calibration information may vary; accordingly, in different embodiments, the specifics of how the calibration information is utilized may vary.

[0050] With reference now to Figure 5, a flowchart 500 of an exemplary method of generating a corrected image is depicted, in accordance with one embodiment. Although specific steps are disclosed in flowchart 500, such steps are exemplary. That is, embodiments of the present invention are well suited to performing various other (additional) steps or variations of the steps recited in flowchart 500. It is appreciated that the steps in flowchart 500 may be performed in an order different than presented, and that not all of the steps in flowchart 500 may be performed.

[0051] With reference now to step 510, in one embodiment, a texturing engine obtains a correction value for a destination pixel in a corrected image from a calibration texture. In one such embodiment, the corrected image is generated on a pixel by pixel basis. In this embodiment, the pixel shader is responsible for determining the contents of one particular destination pixel in the corrected image. In some embodiments, the pixel shader obtains a correction value corresponding to the destination pixel, by accessing a calibration texture image. In some such embodiments, this correction value corresponds to a pixel location in a captured image.

[0052] In other embodiments, e.g., where color distortion is being corrected, correction values may comprise color "shift" information. For example, in one embodiment, correction values may indicate how to manipulate the RGB values of a pixel, in order to correct for color distortion introduced by the lens.

[0053] With reference now to step 520, in one embodiment, the texturing engine obtains a source pixel value from a location in a captured image. In some embodiments, the location of the source pixel value is indicated by the correction value obtained from the calibration texture. In several such embodiments, distortion correction involves "moving" pixels in the captured image to different locations, when generating the corrected image. In one such embodiment, this relocation of pixels allows correction of warping distortion effects, e.g., barrel or pincushion distortion.

[0054] In other embodiments, e.g., where color distortion is being corrected, distortion correction involves adjusting the RGB values of the pixel in the captured image, in order to correct for color distortion introduced by the lens.

[0055] With reference now to step 530, the correction value and the source pixel value are utilized to generate a corrected pixel. In some embodiments, the pixel shader uses the correction value to obtain a source pixel value, and writes that source pixel value to the correct location in the corrected image, e.g., to correct for warping distortion. In some other embodiments, the pixel shader uses the correction value to modify some value in the source pixel value, and writes that modified value to the current location in the corrected image, e.g., to correct for color distortion.

[0056] In some embodiments, the steps in flowchart 500 are repeated for every pixel in the corrected image. In some embodiments, e.g., wherein the GPU or texturing engine has multiple pixel shaders available, it may be possible to process multiple pixels simultaneously. In other embodiments, it may be desirable to process each pixel serially.

[0057] It is appreciated that in some embodiments, the method of flowchart 400 may be applied to correct for multiple types of distortion. In some embodiments, several types of distortion may be corrected simultaneously, e.g., both color distortion and warping distortion may be corrected for a single pixel at once, by accessing one or more calibration textures when processing the pixel. In other embodiments, each type of distortion may require a separate processing application, e.g., a captured image may first

be corrected for warping distortion, and the resulting corrected image may then be further corrected to reduce color distortion. Moreover, in some embodiments, various combinations of these approaches may be utilized.

[0058] It is further appreciated that the method described by a flowchart 500 may be modified. For example, while the method described in flowchart 500 works on a per pixel basis relative to the corrected image, in some embodiments the method may be applied in a manner that works on a per pixel basis relative to the source image.

PIXEL CORRECTION

[0059] With reference now to Figure 6, an example of corrected image generation is depicted, in accordance with one embodiment. While Figure 6 depicts certain specific, enumerated features and elements, it is understood that embodiments are well suited to applications involving different, additional, or fewer features or elements.

[0060] Figure 6 depicts a corrected image 610, a calibration texture 620, and a source image 630. In the depicted embodiment, corrected image 610 is being generated from a source image 630. Calibration texture 620 is used to correct a warping distortion introduced by the lens which captured source image 630. If, for example, a mobile phone camera is used to capture source image 630, barrel distortion may be introduced by the inexpensive lens arrangement used in such cameras.

[0061] In order to generate a corrected pixel in corrected image 610, e.g., pixel 615, a correction value is obtained. In the depicted embodiment, the correction value is obtained from calibration texture 620. Calibration texture 620 indicates that the correct value for pixel 615 is at location 625 in source image 630. In some embodiments, calibration texture 620 may provide the absolute location, e.g., coordinates, for location 625. In other embodiments, calibration texture 620 may provide the relative location of location 625, e.g., as indicated by arrow 621, position 625 is three pixels to the left of the location of pixel 615.

[0062] The texturing engine uses the correction value from calibration texture 620 to obtain the correct pixel value from source image 630. As shown, pixel 635 is "moved" when generating the corrected image, e.g., the contents of pixel 635 in source image 630 are written to pixel 615 in corrected image 610, as indicated by arrow 641.

[0063] Again, it is understood that while the example depicted in Figure 6 describes an approach which focuses on generating the corrected image on a per pixel basis from the corrected image, other embodiments are well suited to other approaches. For example, in one embodiment, in order to generate a corrected image, the pixels in the source image may be processed first. In such embodiment, when pixel 635 is processed, the texturing engine may use the correction information in calibration texture 620 to output pixel 615 in the proper location in corrected image 610. Moreover, in some embodiments, e.g., where a GPU or texturing engine includes multiple pixel shaders, it may be possible to process multiple pixels simultaneously.

[0064] Further, it is understood that while the example depicted in Figure 6 describes an approach which corrects for warping distortion, e.g., barrel distortion, embodiments are well suited to correcting other types of distortion. For example, if color distortion is to be corrected, a calibration texture may include color correction information, rather than positional correction information. In such an embodiment, when generating a corrected pixel, the texturing engine may retrieve the source pixel from the source image, and use the correction information to modify the RGB values of the pixel, before writing a corrected pixel. Moreover, in some embodiments, multiple types of distortion may be corrected.

SYSTEM FOR REDUCING DISTORTION

[0065] With reference now to Figure 7, a system 700 for reducing distortion in captured imagery is depicted, in accordance with one embodiment. While system 700 is depicted as incorporating specific, enumerated features, it is understood that embodiments are well suited to applications involving additional, fewer, or different features or elements.

[0066] Further, in different embodiments, system 700 may be incorporated into different devices. For example, in one embodiment, system 700 is incorporated into a digital camera. In another embodiment, system 700 may be incorporated into a camcorder. In another embodiment, system 700 may be incorporated into a different mobile optical device, e.g., a cell phone having camera functionality. Further, in some embodiments, the image capturing components of system 700 may be incorporated into a separate

device from the distortion reduction and/or display elements of system 700, e.g., a digital camera used to capture a picture, and a desktop or laptop computer used to reduce distortion and display the image.

[0067] In the depicted embodiment, system 700 includes lens 710, sensor 715, graphics processing module 720, storage medium 740, and display 750. As shown, lens 710 incorporates one or more optical elements for focusing light and capturing an image. Sensor 715, in the depicted embodiment, receives the light focused through lens 710. In some embodiments, such as a digital camera, sensor 715 also converts the perceived image into pixel values. As shown in Figure 7, this captured image can be stored on storage medium 740. Storage medium 740, in different embodiments, may comprise any of a wide variety of storage devices, e.g., internal flash memory, a removable storage device, or any other suitable storage medium.

[0068] In the depicted embodiment, graphics processing module 720 is included in system 700. In different embodiments, the purpose and functionality of graphics processing module 720 may vary. For example, in some embodiments, e.g., where system 700 is incorporated into a mobile phone, graphics processing module 720 may be used to display information and render graphics for output to display 750. In the depicted embodiment, graphics processing module 720 includes GPU 725. GPU 725, in some embodiments, is used for a variety of functions, e.g., rendering three-dimensional objects. As shown, GPU 725 includes a number of pixel shaders 727. A pixel shader, in some

embodiments, is used by GPU 725 to calculate the appropriate pixel value for a particular pixel when displaying an image.

[0069] Graphics processing module 720 is also shown as incorporating a buffer 730.

Buffer 730, in some embodiments, is used for storing and retrieving information used by graphics processing module 720 and GPU 725 in displaying an image. In different embodiments, buffer 730 may take different forms. For example, in some embodiments, buffer 730 is composed of random access memory (RAM). In the depicted embodiment, portions of buffer 730 are shown as being allocated for storing a source image 731, a warping image 733, and a corrected image 735.

[0070] System 700 may be utilized to reduce distortion in a captured image or video. In one embodiment, the distortion introduced by lens 710 is predetermined. One approach to calculating this distortion is described above, with reference to flowchart 400, e.g., using the lens to capture a picture of a reference image, and calculating the correction necessary in the reference image. This correction information can then be stored, e.g., as warping image 733.

[0071] When system 700 later captures an image, the captured image can be processed by graphics processing module 720 to reduce the distortion introduced by lens 710. In the depicted embodiment, the captured image is stored as source image 731. GPU 725 is then utilized to generate a corrected image, corrected image 735. One approach to such a corrected image generation is described above, in flowchart 500.

[0072] In some embodiments, where multiple types of distortion may be corrected, it may be necessary to select between several possible warping images. In several such embodiment, graphics processing module 720, or a system processor (not shown), may use additional considerations or data associated with the captured image, in order to select the appropriate correcting texture. For example, if two warping images are available for correcting distortion introduced by lens 710 under two different lighting conditions, the lighting conditions observed by sensor 715 at the time the image was captured may be utilized to select between the two correcting textures.

[0073] Embodiments of the present invention are thus described. While the present invention has been described in particular embodiments, it should be appreciated that the present invention should not be construed as limited by such embodiments, but rather construed according to the following claims.

CLAIMS:

What is claimed is:

1. A method of reducing image distortion, comprising:
 - determining distortion information associated with a lens system;
 - storing said distortion information;
 - processing a captured image taken with said lens system, with reference to said distortion information; and
 - generating a corrected image comprising reduced image distortion compared to said captured image.
2. The method of Claim 1, wherein said determining comprises:
 - taking a calibration picture of a reference image using said lens system; and
 - calculating said distortion information by comparing said calibration picture with said reference image.
3. The method of Claim 2, further comprising:
 - repeating said taking and said calculating for a plurality of reference images, to determine said distortion information for a plurality of distortion types.
4. The method of Claim 1, wherein said generating comprises:
 - extracting a correction value for a destination pixel in said corrected image from said distortion information;
 - obtaining a source pixel value from said captured image;

modifying said source pixel value with reference to said correction value to generate said destination pixel; and

repeating said extracting, said obtaining, and said modifying for each of a plurality of destination pixels in said corrected image.

5. The method of Claim 4, performed by a shader of a graphics processing unit (GPU).

6. The method of Claim 1, wherein said storing comprises:

writing said distortion information as a correction image.

7. The method of Claim 6, wherein said correction image comprises a plurality of correctional pixel data.

8. The method of Claim 7, wherein said correctional pixel data comprises position translation information.

9. The method of Claim 8, wherein said processing comprises:

obtaining a source pixel value from said captured image;

translating said source pixel value from a first location in said captured image to a second location in a corrected image, with reference to said position translation information; and

repeating said obtaining and said translating for a plurality of destination pixels in said corrected image.

10. A system for reducing lens distortion in a captured image, comprising:
- a buffer for storing information; and
 - a texturing engine for generating pixel data, wherein said texturing engine is configured to access a correction image stored in said buffer, and use said correction image to generate a corrected image from said captured image.
11. The system of Claim 10, wherein said texturing engine comprises a plurality of pixel shaders.
12. The system of Claim 11, wherein each of said plurality of pixel shaders is configured to generate a corrected pixel in said corrected image.
13. The system of Claim 12, wherein said plurality of pixel shaders are configured to generate said corrected pixel by obtaining a correction value from said correction image for said corrected pixel, obtaining a source pixel from said captured image, and modifying said source pixel with reference to said correction value to generate said corrected pixel.
14. The system of Claim 10, further comprising a camera with a lens system for providing said captured image, and wherein said correction image comprises a two dimensional image.

15. The system of Claim 10, wherein said correction image comprises a plurality of separable one-dimensional images.
16. The system of Claim 10, wherein said correction image comprises a one-dimensional radial image.
17. A computer-readable medium having computer-readable program code embodied therein for causing a computer system to execute a method of warping a captured image to reduce distortion therein, said method comprising:
- generating distortion correction information;
 - storing said distortion correction information as a warping texture;
 - loading said warping texture into a graphics processing unit (GPU); and
 - producing a modified image by using said GPU to warp said captured image with reference to said warping texture.
18. The computer-readable medium of Claim 17, wherein said distortion correction information comprises absolute pixel correction information.
19. The computer-readable medium of Claim 17, wherein said distortion correction information comprises relative pixel correction information.
20. The computer-readable medium of Claim 17, wherein said using said GPU comprises:

assigning a pixel shader to generate a destination pixel in said modified image;
using said pixel shader and said destination pixel to obtain a correction value
corresponding to said destination pixel from said warping texture;
using said to associate her and said correction value to obtain a source pixel from
said captured image; and
outputting said source pixel as said destination pixel.