US006177956B1

(12) United States Patent
Anderson et al.

(10) Patent No.: US 6,177,956 B1
(45) Date of Patent: *Jan. 23, 2001

(54) SYSTEM AND METHOD FOR CORRELATING PROCESSING DATA AND IMAGE DATA WITHIN A DIGITAL CAMERA DEVICE

(75) Inventors: **Eric C. Anderson**, San Jose; **Mike M. Masukawa**, Los Gatos, both of CA (US)

(73) Assignee: **FlashPoint Technology, Inc.**, San Jose, CA (US)

(*) Notice: This patent issued on a continued prosecution application filed under 37 CFR 1.53(d), and is subject to the twenty year patent term provisions of 35 U.S.C. 154(a)(2).

Under 35 U.S.C. 154(b), the term of this patent shall be extended for 0 days.

(21) Appl. No.: **08/735,705**

(22) Filed: **Oct. 23, 1996**

(51) **Int. Cl.**^{7} **H04N 5/76**

(52) **U.S. Cl.** **348/231**; 348/220; 348/207; 348/222

(58) **Field of Search** 386/107, 117; 348/207, 220, 221, 231, 232, 233, 239, 576, 577, 222; 358/906, 909.1; 707/1, 200, 100, 104; 345/131, 132, 509, 515, 192, 193; H04N 5/225

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,065,246	*	11/1991	Takemoto et al.	348/232
5,153,729	*	10/1992	Saito	348/232
5,402,170		3/1995	Parulski et al.	348/211
5,475,428		12/1995	Hintz et al.	348/263
5,475,441		12/1995	Parulski et al.	348/552
5,477,264		12/1995	Sarbadhikari et al.	348/231
5,493,335		2/1996	Parulski et al.	348/233
5,496,106	*	3/1996	Anderson	348/255
5,633,678	*	5/1997	Parulski et al.	348/231
5,806,072	*	9/1998	Kuba et al.	348/231

OTHER PUBLICATIONS

Martyn Williams, Review–NEC PC–DC401 Digital Still Camera, AppleLink Newbytes, Mar. 15, 1996, pp. 1–3.

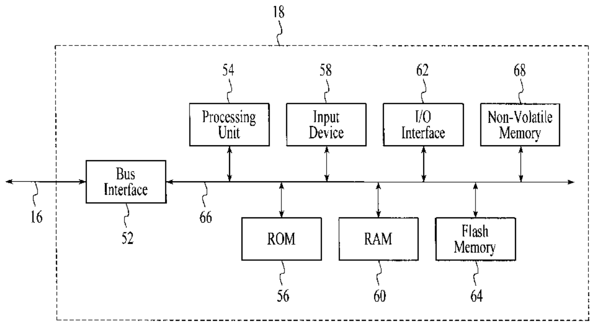
* cited by examiner

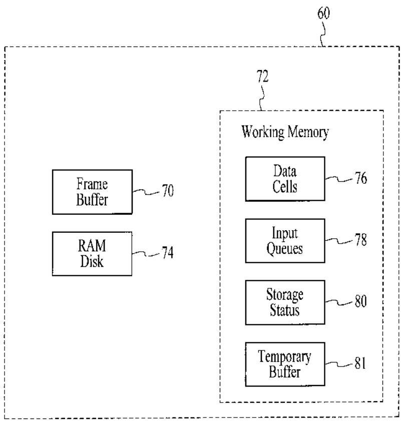
Primary Examiner—Tuan Ho
(74) *Attorney, Agent, or Firm*—Sawyer Law Group LLP

(57) **ABSTRACT**

A system and method for correlating processing data and image data within a digital camera device comprises a capture device for gathering image data, a data cell manager for building a data cell containing processing data and for linking the data cell to the image data, and a processor device for processing and compressing the image data by using the processing data stored within the data cell.

18 Claims, 11 Drawing Sheets





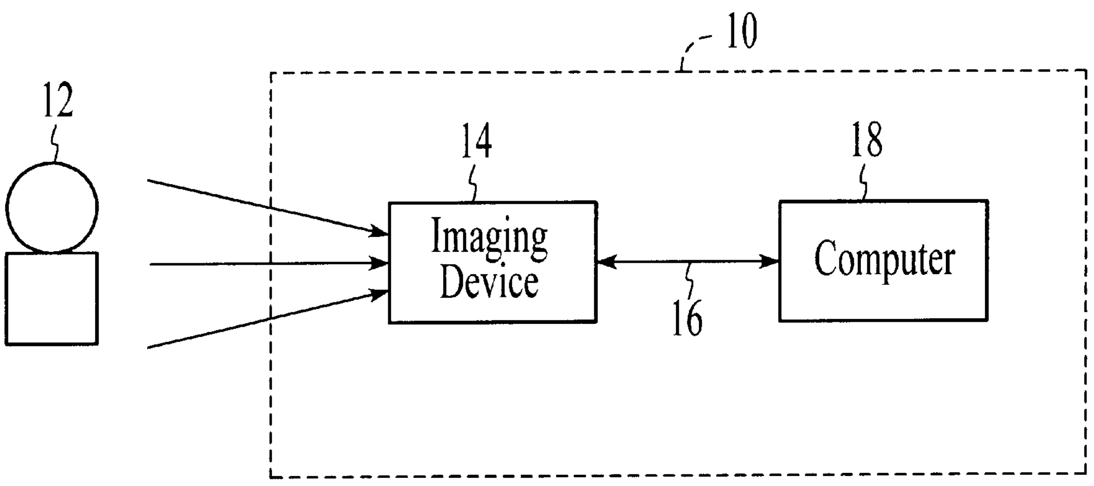


FIG. 1

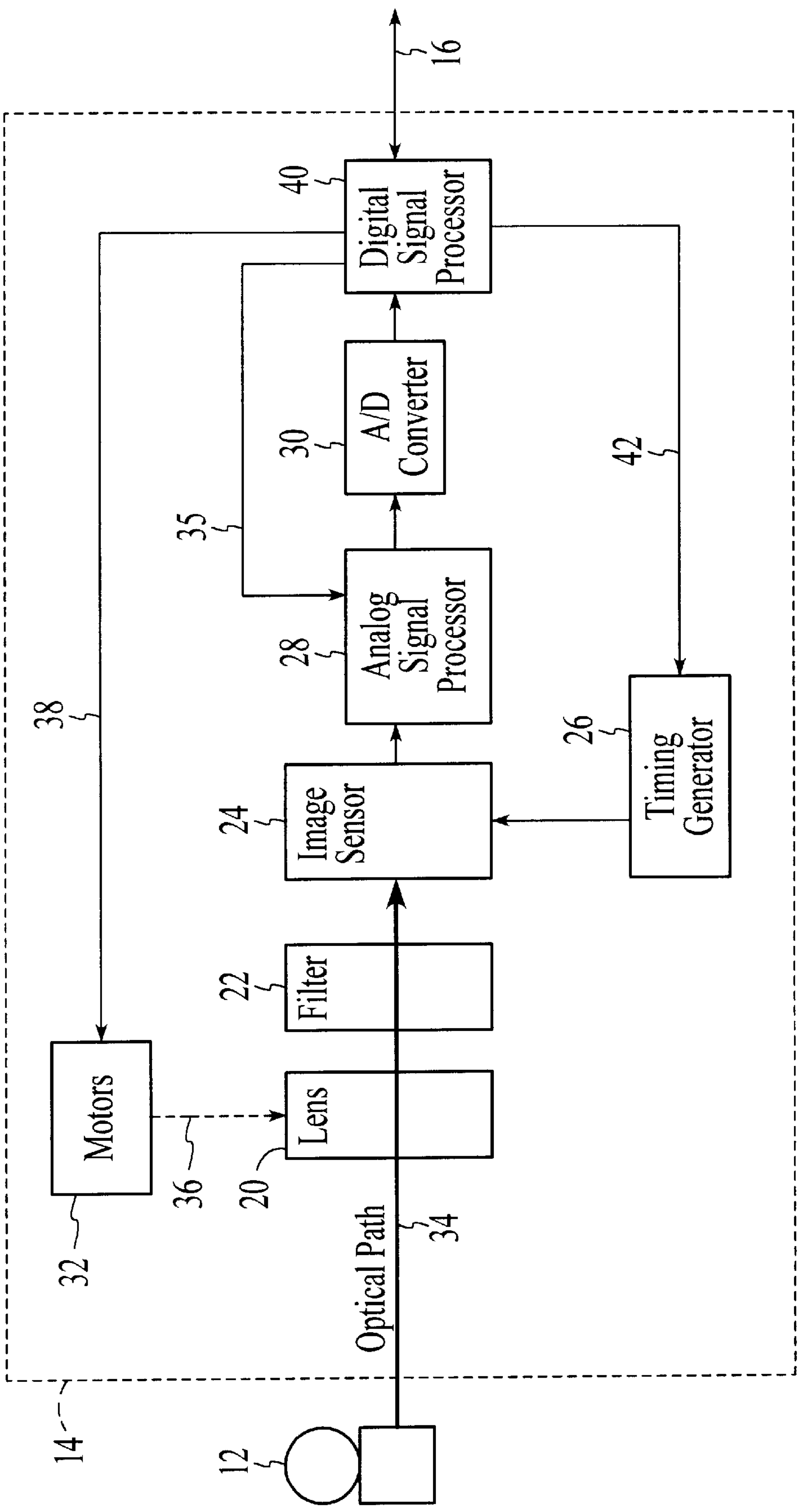

16
40
Digital Signal Processor
30
A/D Converter
42
35
28
Analog Signal Processor
38
26
Timing Generator
24
Image Sensor
22
Filter
Motors
Lens
36
20
32
Optical Path
34
14
12

FIG. 2

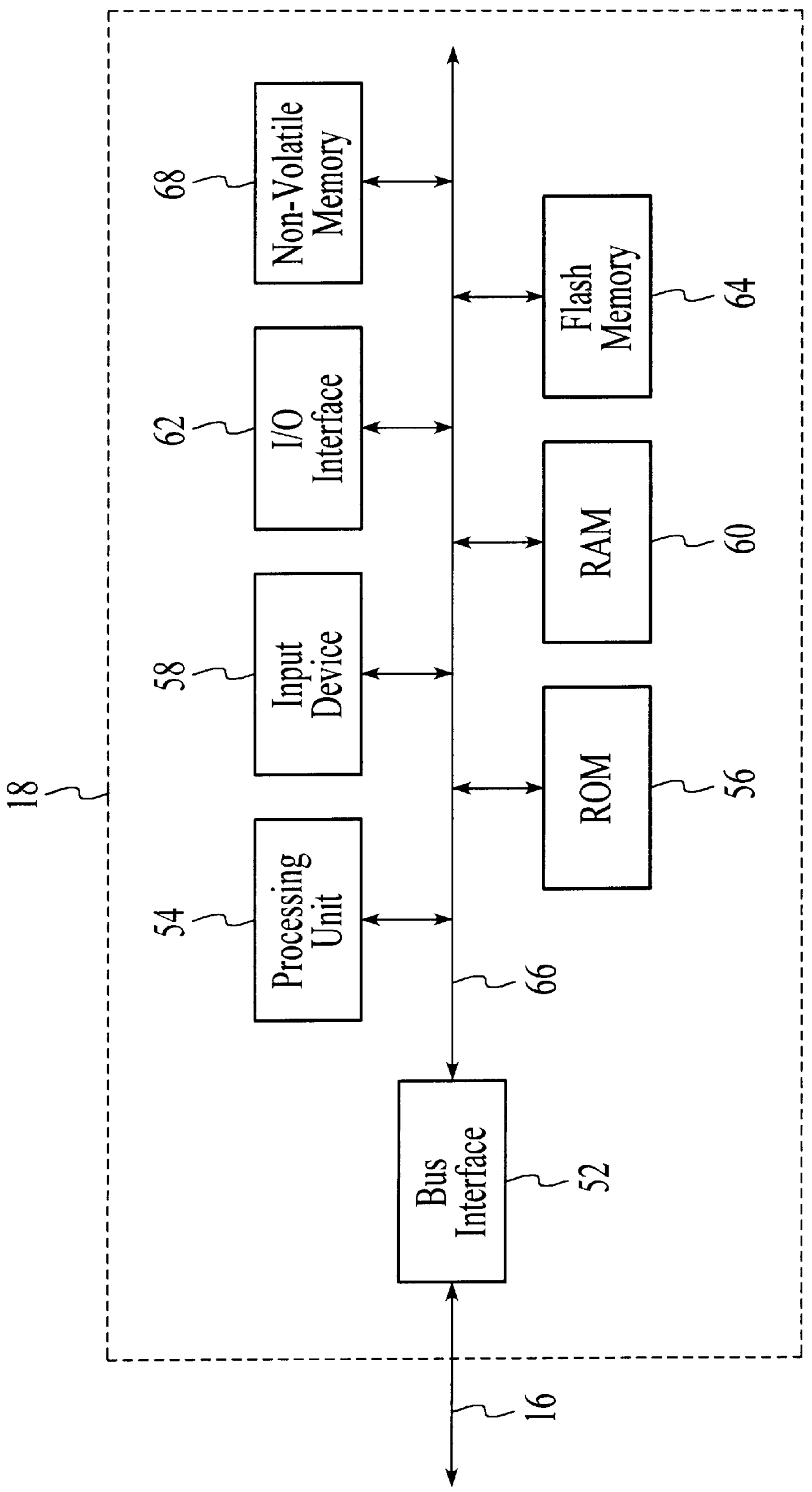

18
68
Non-Volatile Memory
62
I/O Interface
58
Input Device
54
Processing Unit
Flash Memory
64
RAM
60
ROM
56
66
Bus Interface
52
16

FIG. 3

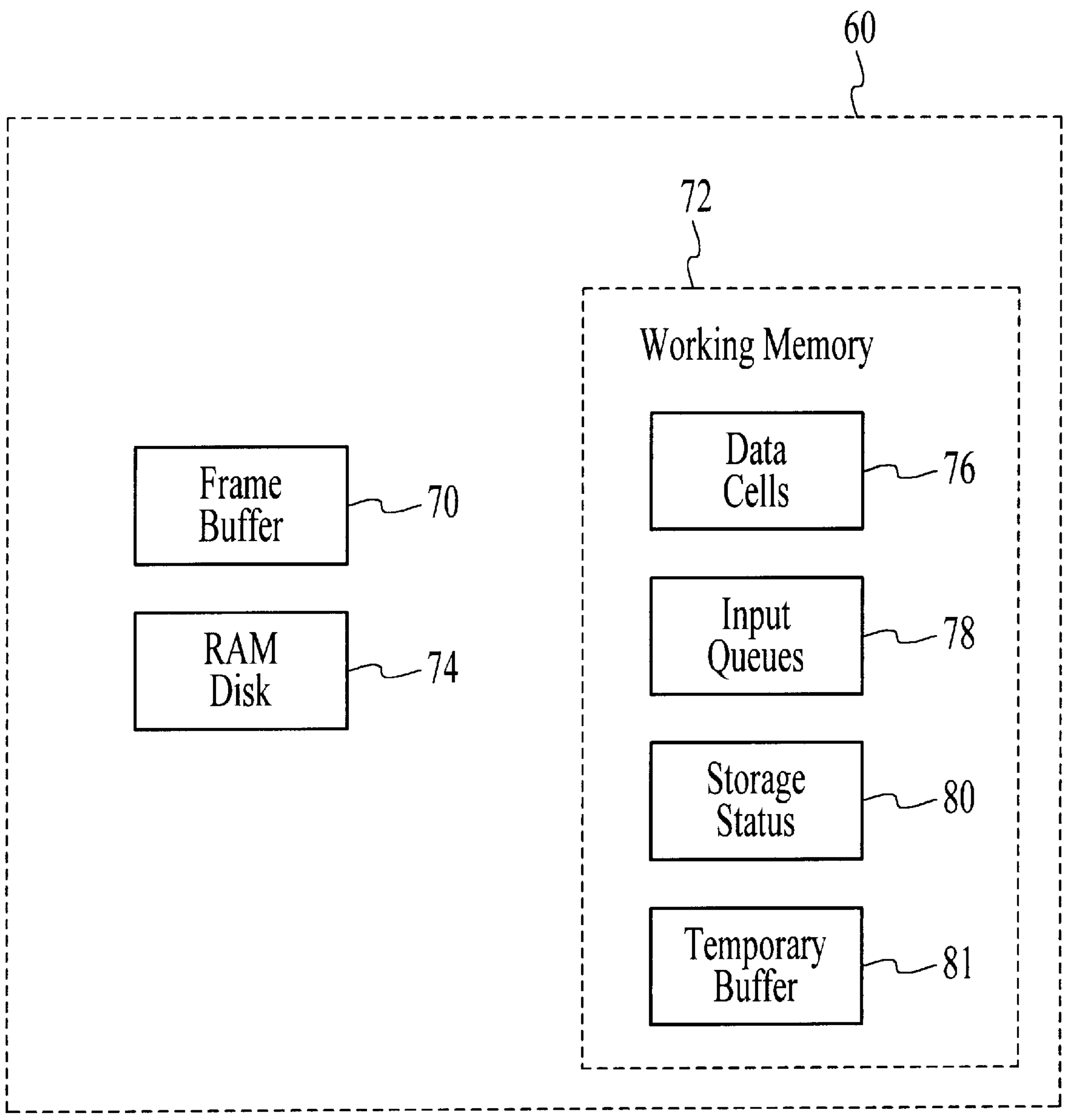

60
72
Working Memory
Frame Buffer
70
RAM Disk
74
Data Cells
76
Input Queues
78
Storage Status
80
Temporary Buffer
81

FIG. 4

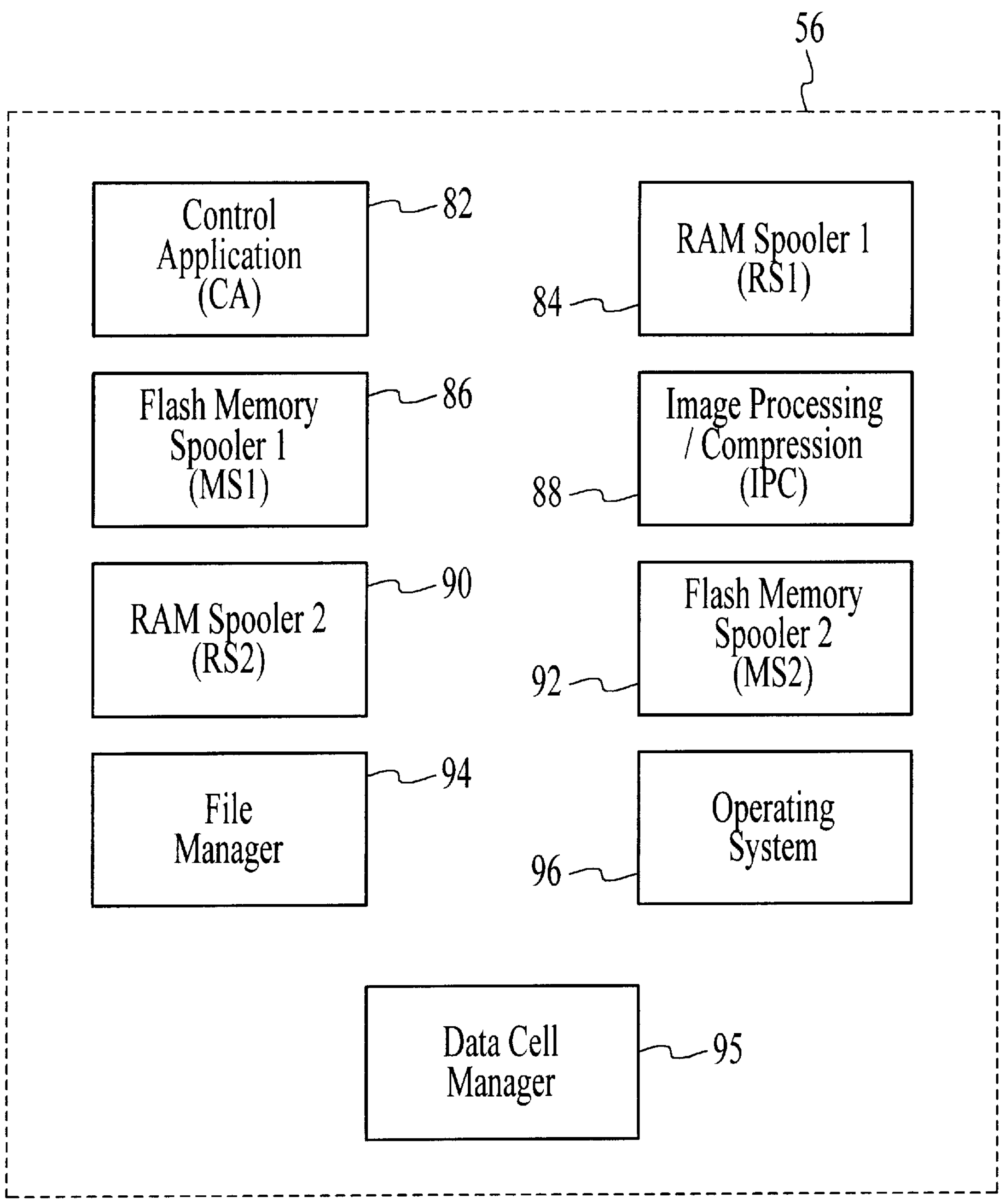

56
Control Application (CA)
82
RAM Spooler 1 (RS1)
84
Flash Memory Spooler 1 (MS1)
86
Image Processing / Compression (IPC)
88
RAM Spooler 2 (RS2)
90
Flash Memory Spooler 2 (MS2)
92
File Manager
94
Operating System
96
Data Cell Manager
95

FIG. 5

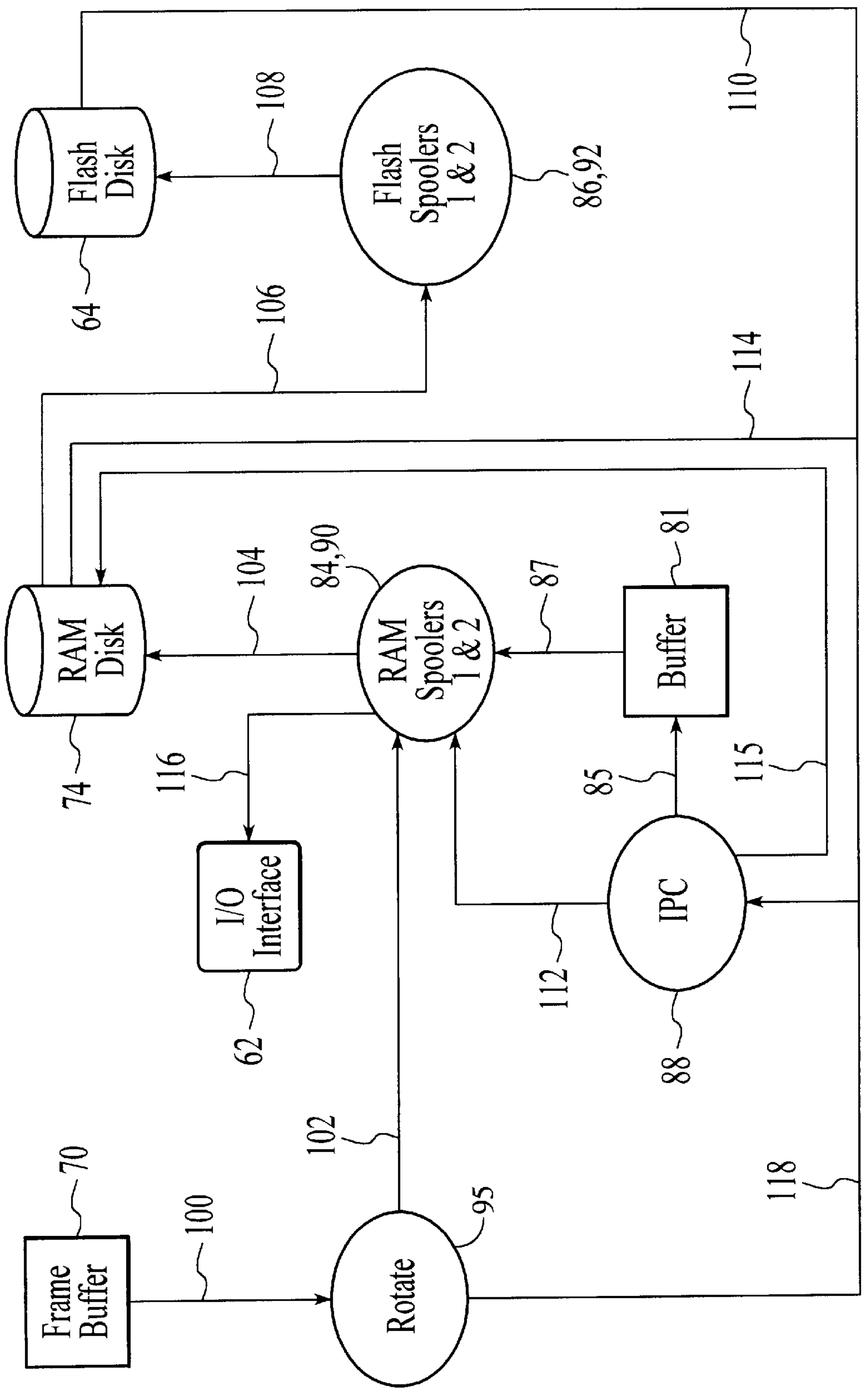

Flash Disk
64
108
Flash Spoolers 1 & 2
86,92
110
106
114
RAM Disk
74
104
84,90
RAM Spoolers 1 & 2
87
Buffer
81
116
I/O Interface
62
85
115
IPC
112
88
Frame Buffer
70
100
Rotate
95
102
118

FIG. 6

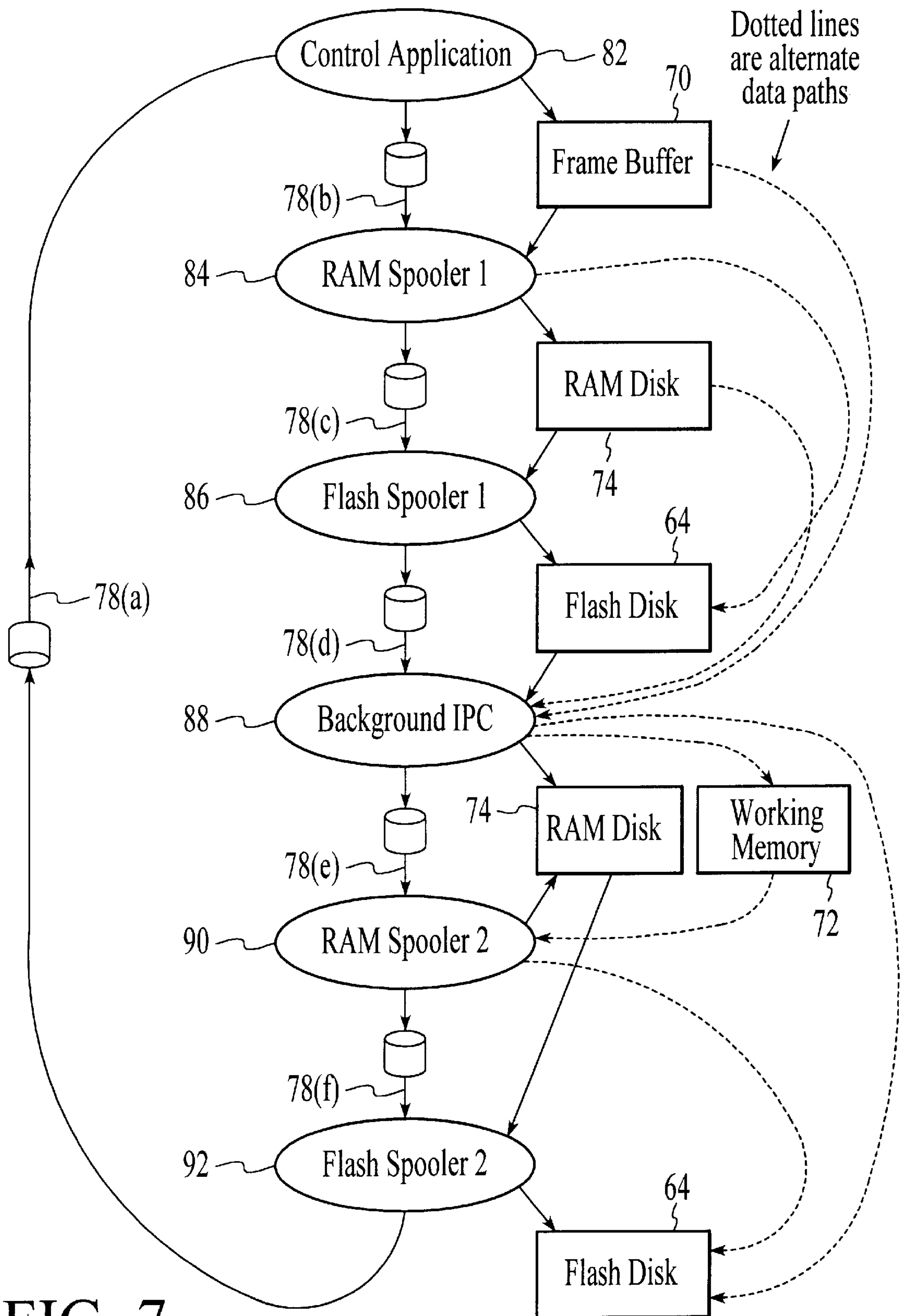

Control Application
82
Dotted lines are alternate data paths
70
Frame Buffer
78(b)
84
RAM Spooler 1
RAM Disk
78(c)
74
86
Flash Spooler 1
64
Flash Disk
78(a)
78(d)
88
Background IPC
74
RAM Disk
Working Memory
78(e)
72
90
RAM Spooler 2
78(f)
92
Flash Spooler 2
64
Flash Disk

FIG. 7

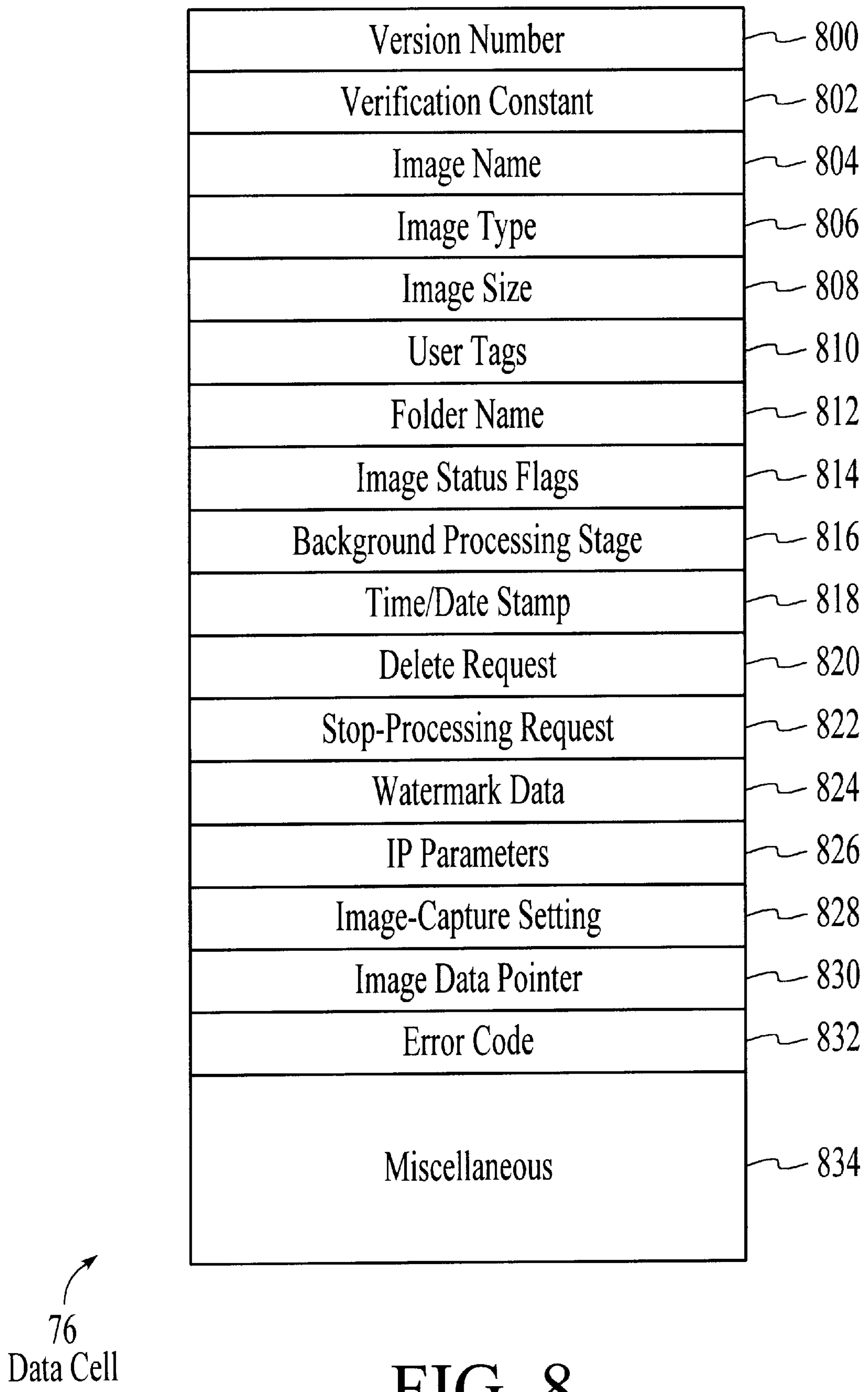

Version Number
800
Verification Constant
802
Image Name
804
Image Type
806
Image Size
808
User Tags
810
Folder Name
812
Image Status Flags
814
Background Processing Stage
816
Time/Date Stamp
818
Delete Request
820
Stop-Processing Request
822
Watermark Data
824
IP Parameters
826
Image-Capture Setting
828
Image Data Pointer
830
Error Code
832
Miscellaneous
834
76
Data Cell

FIG. 8

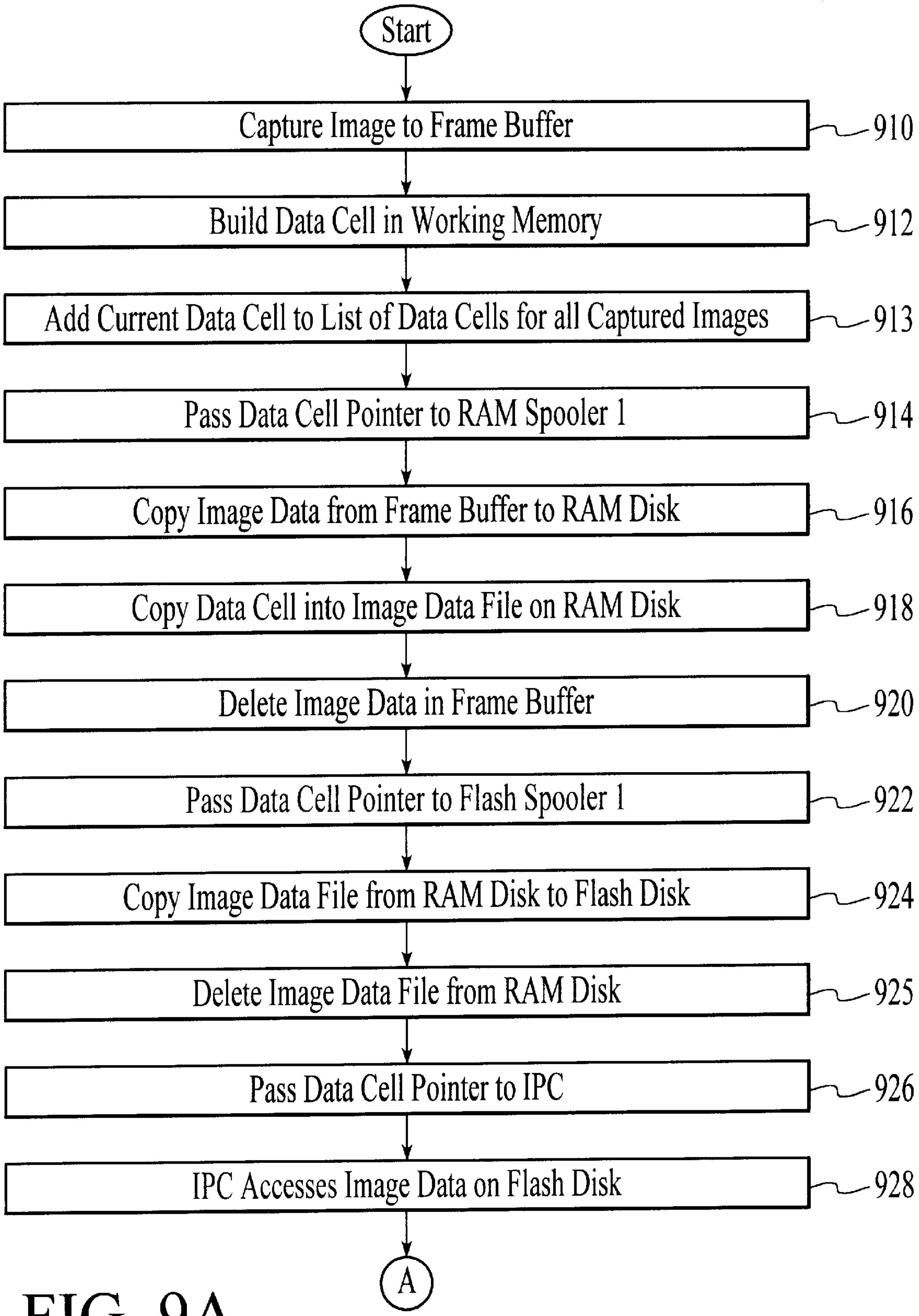

Start
Capture Image to Frame Buffer
910
Build Data Cell in Working Memory
912
Add Current Data Cell to List of Data Cells for all Captured Images
913
Pass Data Cell Pointer to RAM Spooler 1
914
Copy Image Data from Frame Buffer to RAM Disk
916
Copy Data Cell into Image Data File on RAM Disk
918
Delete Image Data in Frame Buffer
920
Pass Data Cell Pointer to Flash Spooler 1
922
Copy Image Data File from RAM Disk to Flash Disk
924
Delete Image Data File from RAM Disk
925
Pass Data Cell Pointer to IPC
926
IPC Accesses Image Data on Flash Disk
928
A

FIG. 9A

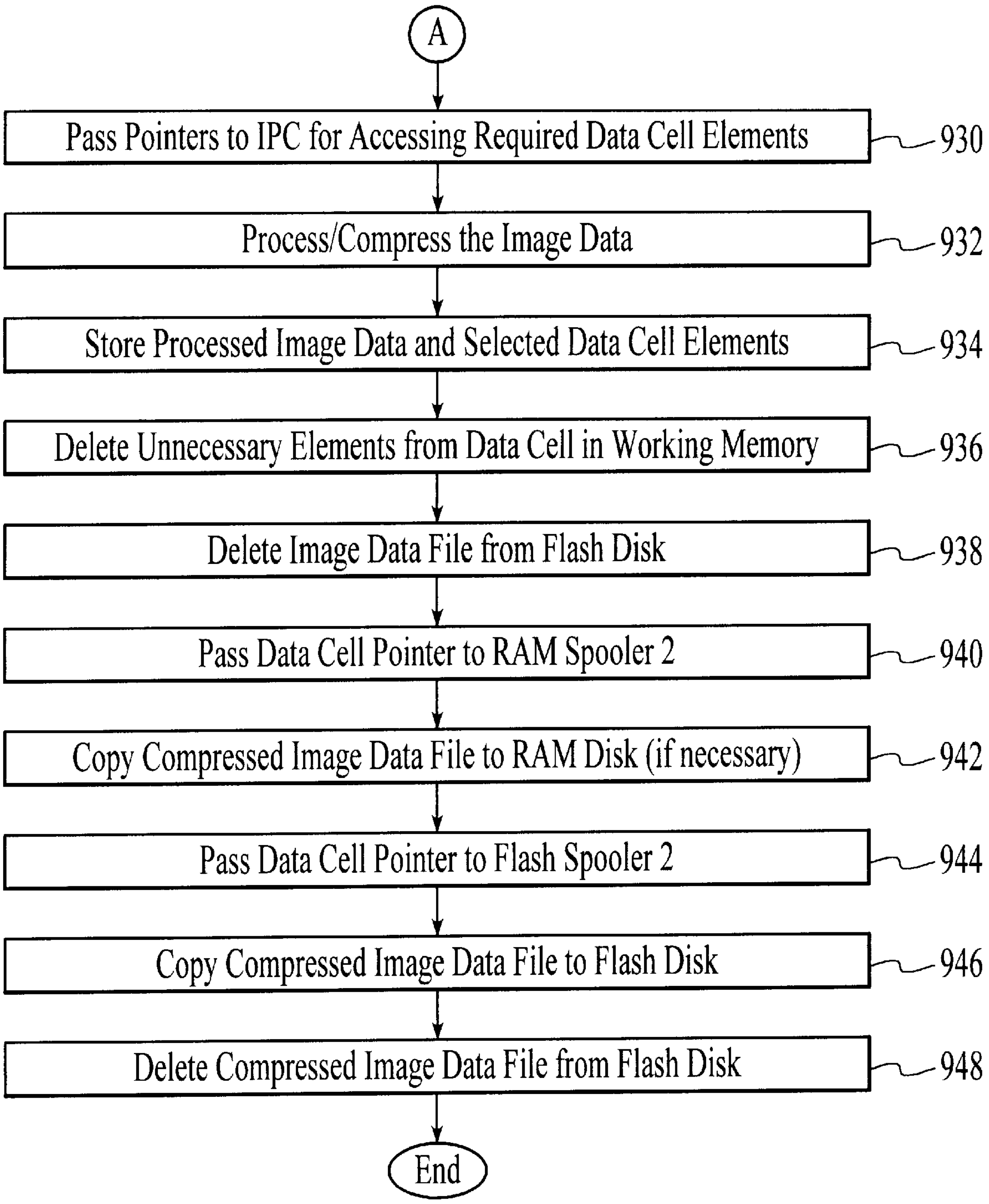

A
Pass Pointers to IPC for Accessing Required Data Cell Elements
930
Process/Compress the Image Data
932
Store Processed Image Data and Selected Data Cell Elements
934
Delete Unnecessary Elements from Data Cell in Working Memory
936
Delete Image Data File from Flash Disk
938
Pass Data Cell Pointer to RAM Spooler 2
940
Copy Compressed Image Data File to RAM Disk (if necessary)
942
Pass Data Cell Pointer to Flash Spooler 2
944
Copy Compressed Image Data File to Flash Disk
946
Delete Compressed Image Data File from Flash Disk
948
End

FIG. 9B

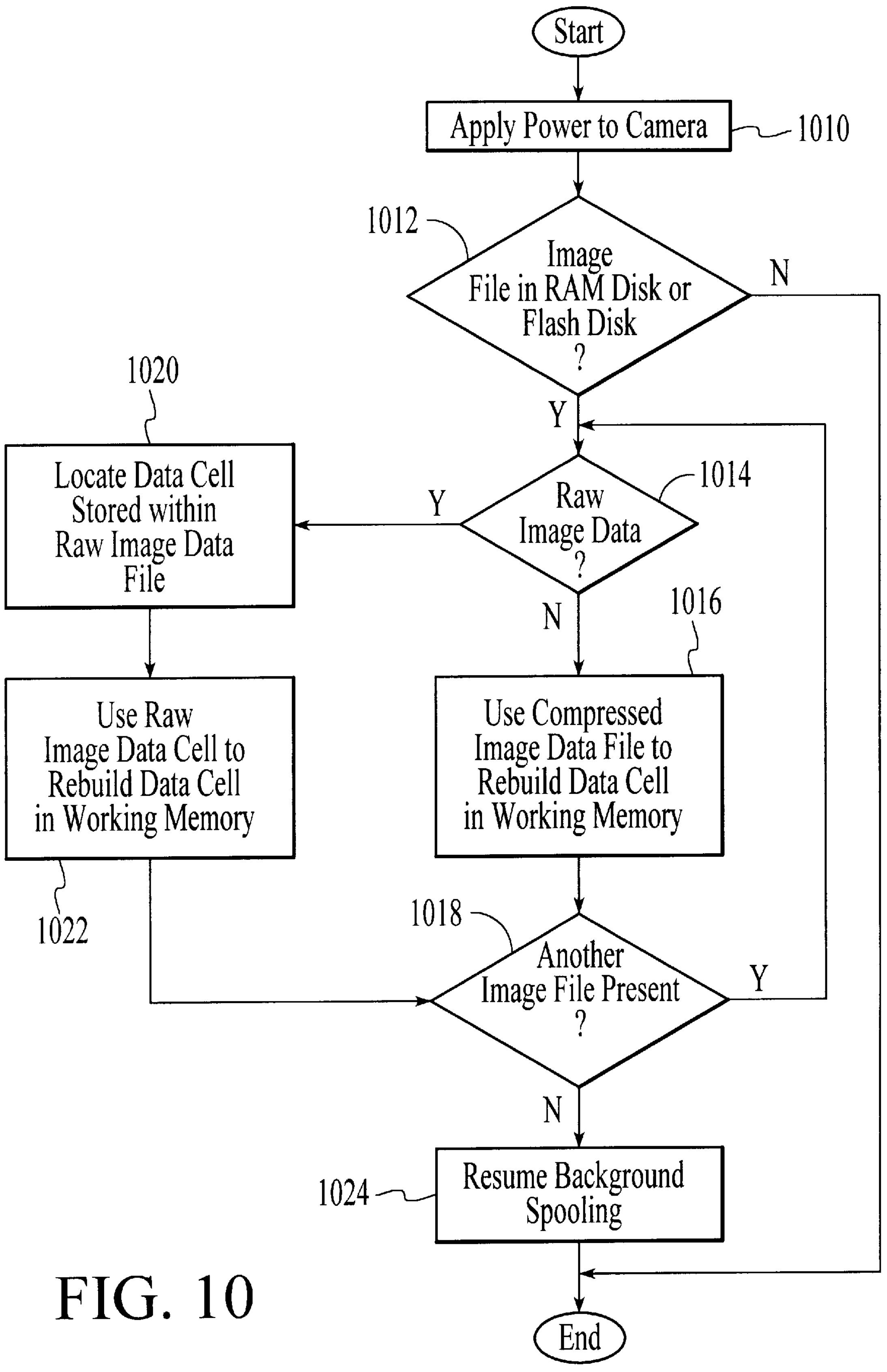

Start
Apply Power to Camera
1010
1012
Image File in RAM Disk or Flash Disk ?
N
Y
1020
Locate Data Cell Stored within Raw Image Data File
Y
Raw Image Data ?
1014
N
1016
Use Raw Image Data Cell to Rebuild Data Cell in Working Memory
Use Compressed Image Data File to Rebuild Data Cell in Working Memory
1022
1018
Another Image File Present ?
Y
N
1024
Resume Background Spooling
End

FIG. 10

SYSTEM AND METHOD FOR CORRELATING PROCESSING DATA AND IMAGE DATA WITHIN A DIGITAL CAMERA DEVICE

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates in general to digital camera technology and also relates more particularly to a system and method for correlating processing data and image data within a digital camera device.

2. Description of the Background Art

An important digital still-camera performance feature is the number of captured images that can be stored in the camera's finite memory. To maximize the image-carrying capacity of digital still-cameras, it is desirable to compress the images prior to storage. Conventional digital cameras typically perform image processing on the captured raw image data and then use a high-quality image compression routine (such as JPEG) to compress the image data.

Furthermore, digital cameras may frequently be required to capture and concurrently process multiple successive sets of captured image data. Each captured image, however, has important related information which may be needed during the image processing and compression operations, as well as subsequently. Examples of such processing and compression data might include camera settings (e.g., focus, aperture, and white-balance information), time and date of image capture and image processing parameters.

Each captured image potentially has a different set of relevant processing information. Therefore, each captured image within a digital camera may require a separate and unique set of processing data. Furthermore, to permit effective access to these unique sets of processing and compression data, each of the sets of captured image data and the corresponding processing data must be linked together. Efficient access to the processing data at the appropriate time is thus an important feature of modern digital cameras. Therefore, what is needed is an improved system and method for correlating processing data and image data within a digital camera device.

SUMMARY OF THE INVENTION

The present invention is a system and method for correlating processing data and image data within a digital camera device. In the present invention, an imaging device captures an image in response to an image capture request and responsively produces corresponding raw image data which is temporarily stored into a frame buffer. A data cell manager then builds a corresponding data cell containing various types of processing data which the data cell manager links to the captured raw image data. The processing data may include information such as image-capture settings, image size, user tags and image-processing parameters. The data cell is preferably stored in working memory within the camera DRAM.

A first RAM spooler then typically transfers the raw image data into an individual image data file within a RAM disk in the camera DRAM. Next, the data cell manager makes a copy of the data cell in working memory and places the copy into the image data file stored in the RAM disk for recovery purposes. The image data in the frame buffer is then deleted to allow a camera user to capture another image.

A first flash spooler next transfers the raw image data file from the RAM disk to a flash memory which preferably is a removable flash disk. An image processor device then accesses, processes and compresses the raw image data using the corresponding processing data stored in the data cell. The image processor device may then directly store the compressed data into a compressed image data file on the RAM disk, or alternately, a second RAM spooler may store the compressed image data into a compressed image data file on the RAM disk. The cell manager then stores selected necessary processing data from the corresponding data cell into the compressed image file. The cell manager also deletes unnecessary processing data from the data cell stored in working memory. A second flash spooler then transfers the compressed image data file from the RAM disk to the flash memory.

The data cell thus allows specific camera settings which exist at image capture time to be effectively saved and linked to the corresponding image data, thereby permitting subsequent changes of the camera settings without losing those camera settings previously saved in the data cell. The present invention also allows the camera device to recover from disruptive events such as power failures which threaten to damage the captured image data. Following a disruptive event, the data cell manager may locate the copy of the data cell which is stored in the image data file and then use this copied data cell to rebuild the original data cell stored within working memory in the camera DRAM. Once the original data cell has been reconstructed, the camera may then successfully complete the processing, compression and storage operations for the captured image data.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a block diagram showing a digital camera device according to the present invention;

FIG. **2** is a block diagram showing a preferred embodiment of the FIG. **1** imaging device according to the present invention;

FIG. **3** is a block diagram showing a preferred embodiment of the FIG. **1** computer of the present invention;

FIG. **4** is a block diagram showing a preferred embodiment of a Random Access Memory (RAM) of the FIG. **3** computer;

FIG. **5** is a block diagram showing a preferred embodiment of a Read Only Memory (ROM) of the FIG. **3** computer;

FIG. **6** is a block diagram showing a preferred embodiment of the FIG. **1** camera device according to the present invention;

FIG. **7** is a block diagram showing priority levels of preferred processes and corresponding image data paths;

FIG. **8** is a block diagram of the preferred embodiment for an exemplary data cell according to the present invention;

FIG. **9**A is the initial portion of a flowchart showing the operation of the present invention using the data cell of FIG. **8**;

FIG. **9**B is the final portion of a flowchart showing the operation of the present invention using the data cell of FIG. **8**; and

FIG. **10** is a flowchart showing preferred method steps for using the present invention to recover from a disruptive event within a digital camera device.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

The present invention comprises a system and method for correlating processing data and image data within a digital camera device and includes an imaging device for capturing image data, a data cell manager for building a data cell containing processing data and for linking the data cell to the captured image data, and a processor device for processing and compressing the captured image data by using the processing data stored within the data cell.

Referring now to FIG. **1**, a block diagram of a preferred embodiment of a camera **10** is shown. Camera **10** may be used to capture a set of image data representing an object **12**. Camera **10** preferably comprises an imaging device **14**, an external bus **16** and a computer **18**. Imaging device **14** is optically coupled to object **12** and electrically coupled via external bus **16** to computer **18**. Once a photographer has focused imaging device **14** on object **12** and, using a capture button or some other means, instructed camera **10** to capture an image of object **12**, computer **18** commands imaging device **14** via external bus **16** to capture raw image data representing object **12**. The captured raw image data is transferred over external bus **16** to computer **18** which performs various image processing functions on the image data before storing it in its internal memory. External bus **16** also passes various status and control signals between imaging device **14** and computer **18**.

Referring now to FIG. **2**, a block diagram of a preferred embodiment of imaging device **14** is shown. Imaging device **14** preferably comprises a lens **20** having an iris, a filter **22**, an image sensor **24**, a timing generator **26**, an analog signal processor (ASP) **28**, an analog-to-digital (A/D) converter **30**, a digital signal processor (DSP) **40**, and one or more motors **32**.

In operation, imaging device **14** captures an image of object **12** via reflected light impacting image sensor **24** along optical path **34**. Image sensor **24** responsively generates a set of raw image data representing the captured image **12**. The raw image data is then routed through ASP **28**, A/D converter **30** and DSP **40**. DSP **40** has outputs coupled to lines **35**, **38** and **42** for controlling ASP **28**, motors **32** and timing generator **26**. From DSP **40**, the raw image data passes over external bus **16** to computer **18**.

Referring now to FIG. **3**, a block diagram of a preferred embodiment of computer **18** is shown. Computer **18** comprises a bus interface **52**, a processing unit **54**, a read-only memory (ROM) **56**, an input device **58**, a random access memory (RAM) **60**, an I/O interface **62**, a flash memory **64** and a non-volatile memory **68** coupled together via an internal bus **66**. In the preferred embodiment, computer **18** is embedded as part of camera **10** using a conventional architecture. However, those skilled in the art will recognize that in an alternate embodiment, computer **18** may be a discrete computer system.

Bus interface **52** is preferably a bi-directional first-in, first-out interface for receiving the raw image data and imaging device **14** control signals passed between computer **18** and DSP **40**. Interface **52** has data lines coupled to both external bus **16** and internal bus **66**. Processing unit **54** executes programming instructions stored in ROM **56** and RAM **60** to perform various operations. ROM **56** stores a set of computer readable program instructions which control how processing unit **54** accesses, transforms and outputs the image data. While ROM **56** is employed as a conventional non-volatile memory device for practicing the present invention, those skilled in the art will recognize that in alternate embodiments ROM **56** could be replaced with a functionally equivalent computer useable medium such as a compact disk and drive, a floppy disk and drive, or a flash memory.

Input device **58** preferably comprises a series of control buttons which generate signals translated by processing unit **54** into an image capture request, an operating mode selection request, and various control signals for imaging device **14**. In an alternate embodiment in which computer **18** is a discrete computer system, input device **58** also includes a keyboard and mouse-type controller.

I/O Interface **62** is coupled to internal bus **66** and has an external port connector for coupling computer **18** with a host computer (not shown) for downloading image data stored in RAM **60** and/or flash memory **64**. At the user's choice or when camera **10** is completely filled with image data, I/O Interface **62** enables the image data to be down-loaded, thus freeing up storage space for future sets of image data.

Flash memory **64** serves as an additional image data storage area and is preferably a non-volatile device, readily removable and replaceable by a user. Thus, a user who possesses several flash memories **64** may replace a full flash memory **64** with an empty flash memory **64** to effectively expands the picture taking capacity of camera **10**. In the preferred embodiment of the present invention, flash memory **64** is a flash disk. Non-volatile memory **68** stores an image counter whose current value becomes an identifier for each new set of image data captured by camera **10**. The counter is preferably incremented each time a new image is captured. In the preferred embodiment, non-volatile memory **68** is either an EEPROM or a battery-backed SRAM.

Referring now to FIG. **4**, a block diagram of a preferred embodiment of RAM **60** is shown. RAM **60** is comprised of a frame buffer **70**, a working memory **72** and a RAM disk **74**. Frame buffer **70** preferably comprises a dedicated space of contiguous memory suitable for storing the raw image data generated by image sensor **24**. In alternate embodiments, frame buffer **70** may be memory space allocated within working memory **72**. The function of frame buffer **70** is to store the most recently captured set of raw image data until computer **18** either stores the raw image data in RAM disk **74** or transfers it to an image processing unit.

RAM disk **74** is a memory area within RAM **60** organized in a "sectored" format similar to that of conventional hard disk drives. The RAM disk **74** function is to store image data. RAM disk **74**, in conjunction with flash memory **64**, sets the maximum image holding capacity of camera **10**. Once both flash memory **64** and RAM disk **74** have been filled with compressed image data, the insertion of a new flash memory **64** or down-loading the image data via I/O interface **62** will enable camera **10** to continue capturing new images.

Working memory **72** is comprised of data cells **76**, input queues **78**, storage status **80** and temporary buffer **81**. Data cells **76** are data structures and each data cell **76** is uniquely associated with particular captured image data. A data cell **76** is comprised of a plurality of data cell elements which are further described below in conjunction with FIG. **8**. Input queues **78** are data structures comprised of a plurality of data cell "pointers" each corresponding to data cells **76**. In the preferred embodiment, input queues operate on a first-in/first-out basis.

Storage status **80** is a data structure describing the remaining available memory in both RAM disk **74** and flash memory **64**. Storage status **80** contains the following four conditional variables: "RAM Disk Raw File Space," "RAM Disk Compressed File Space," "Flash Memory Raw File Space" and "Flash Memory Compressed File Space." Each of the four conditional variables is set to one of three values: FULL, ALMOST FULL or OK. If the variable is set to "OK," then space is available for that particular file type (i.e., a raw file or a compressed file) on that particular storage resource (i.e., RAM disk **74** or flash memory **64**). If the variable is set to "ALMOST FULL" then space is not currently available for that particular file type on that particular storage resource, but there will be space in the future. If the variable is set to "FULL" then, absent an increase in available space on storage resources (due, for example, to downloading data or replacing storage units), no space is available for that particular file type on that particular storage resource, nor will space be available in the future. Temporary buffer **81** of working memory **72** is provided for temporarily storing data and/or program code.

Referring now to FIG. **5**, a block diagram of a preferred embodiment of ROM **56** is shown. ROM **56** preferably contains code for processes **82** through **96**, including a control application (CA) **82**, a RAM spooler **1** (RS1) **84**, a flash memory spooler **1** (MS1) **86**, image processing/compression (IPC) **88**, a RAM spooler **2** (RS2) **90**, a flash memory spooler **2** (MS2) **92**, a file manager **94**, a data cell manager **95** and an operating system **96**. In alternate embodiments, the FIG. **5** processes **82** through **96** may be stored in various computer memory types other than ROM **56**.

A "spooler" is herein defined as a routine for transferring data from one process or device to a second process or device. RAM spooler **1** (**84**) transfers raw image data into RAM disk **74**, and flash memory spooler **1** (**86**) transfers raw image data into flash memory **64**. RAM spooler **2** (**90**) transfers compressed image data into RAM disk **74** or to I/O interface **62**, and flash memory spooler **2** (**92**) transfers compressed image data into flash memory **64**.

Control application **82** preferably comprises program instructions for controlling the operation of camera **10** which are executed using processing unit **54**. For example, control application **82** controls data cell manger **95** to create and maintain data cells **76**. Image processing/compression **88** compresses the raw image data to maximize the image-carrying capacity of camera **10**, and also processes the raw image data to permit readily displaying the captured image data on a host computer. In the preferred embodiment, processes **82** through **96** are comprised of a series of software steps implemented on top of a multithreaded operating system and may therefore run in parallel operation. Data cell manager **95** controls and coordinates data cells **76** and is further discussed below in conjunction with FIGS. **7** through **10**.

Referring now to FIG. **6**, a block diagram of the preferred embodiment for camera **10** is shown. In FIG. **6**, frame buffer **70** receives and stores raw image data previously captured by imaging device **14**. Frame buffer **70** then provides the raw image data via line **100** to rotate process **95**.

Process **95** rotates the captured image if necessary and then transfers control of the raw image data to RAM spooler **1** (**84**) using line **102**. Alternately, if RAM disk **74** is full, rotate process **95** may transfer control of the raw image data directly to image processing/compression (IPC) **88** using line **118**. If RAM spooler **1** (**84**) receives control of the raw image data, it then stores the raw image data into RAM disk **74** using line **104**.

Flash spooler **1** (**86**) may then access the raw image data from RAM disk **74** via line **106** and store it into flash memory **64** using line **108**. Alternately, if flash memory **64** is full, RAM disk **74** may provide the raw image data directly to IPC **88** using line **114**. If flash spooler **1** (**86**) stores the raw image data into flash memory **64**, then IPC **88** typically accesses the stored raw image data using line **110** and processes the raw data to responsively obtain compressed image data.

IPC **88** may bypass RAM spooler **2** (**90**) and store the compressed data directly to RAM disk **74** via line **115**, or alternately, if RAM disk **74** is temporarily full, IPC **88** may write the compressed data to temporary RAM buffer **81** via line **85**. RAM spooler **2** (**90**) may then access the compressed image data via line **87** and write the accessed data into RAM disk **74** via line **104**. RAM spooler **2** (**90**) may also download the compressed image data to I/O interface **62** using line **116**. Once the compressed data is in RAM disk **74**, flash spooler **2** (**92**) then accesses the data via line **106** and writes the compressed data into flash memory **64**.

The present invention may thus process and store a sequence of captured images received from imaging device **14**. Although the above example traces the typical data path for a single captured image, the present invention may readily operate on a plurality of captured images progressing through various stages of camera **10**. Therefore, multiple sets of image data may exist simultaneously within computer **18**. The current processing stage for a specific set of image data is preferably indicated by flags located in the image data's unique data cell **76**.

Referring now to FIG. **7**, a block diagram of priority levels for processes **84** through **92** of the preferred embodiment is shown. Background processes **84** through **92** are preferably allotted processing unit **54** time depending on their priority level. This priority level is related to the goal of rapidly emptying frame buffer **70** to enable rapid capture of successive sets of image data.

Control application **82** transfers raw image data from imaging device **14** to frame buffer **70** and may supersede any of background processes **84** through **92**. The background process with the highest priority is RAM spooler **1** (**84**) which moves raw image data out of frame buffer **70** to RAM disk **74**. The second highest priority is flash memory spooler **1** (**86**) which moves raw image data out of RAM disk **74** to flash memory **64**. The third highest priority is Image Processing/Compression **88** which accesses raw image data and responsively processes and compresses the image data before storing it as compressed image data into RAM disk **74**, or if RAM disk **74** is full, into temporary RAM buffer **81** of working memory **72**. The fourth highest priority is RAM spooler **2** (**90**) which, if necessary, may move compressed image data out of working memory **72** into RAM disk **74**. The lowest priority is flash memory spooler **2** (**92**) which moves the compressed image data out of RAM disk **74** into flash memory **64**. Those skilled in the art will recognize that either a greater or a lesser number of priority levels than the preferred five may be used in the present invention. Also, alternate embodiments may establish different criteria for routing the captured image data, depending upon memory resources available and/or the maximum image capture rate desired. File manager process **94** and operating system process **96** are not assigned specific priority levels since they either operate in the background or under interrupt conditions.

Processes **82** through **92** preferably each has a respective input queue **78**(*a*) through **78**(*f*) which operates on a first-in/first-out basis. If one of processes **82** through **92** has a data cell **76** pointer in its input queue, then only that process can access and perform operations on the image data associated with that particular data cell **76**. The data cell pointers are passed between processes **82** through **92** in a specific order until the original raw image data has been fully processed, compressed and stored in a memory resource.

The priority level scheme introduced above may "block" one or more processes **84** through **92** even though a data cell **76** pointer is in its input queue **78**. For example, since moving raw image data out of frame buffer **70** has the highest priority, if a user repeatedly captures images in rapid succession, RAM spooler **1** (**84**) will continue to operate until RAM disk **74** becomes filled with raw image data. While RAM spooler **1** (**84**) is operating, all of the other lower priority processes **86** through **92** will be "blocked" (i.e., idled), even though some of the lower priority processes **86** through **92** may still have data cell **76** pointers in their input queues **78**. This blocking of lower priority processes applies to all priority levels. For example, operation of flash memory spooler **1** (**86**) will block image processing/compression **88**, RAM spooler **2** (**90**) and flash memory spooler **2** (**92**), and operation of image processing/compression **88** will block RAM spooler **2** (**90**) and flash memory spooler **2** (**92**), and so on, until the image data has been fully processed, compressed and stored in memory. Furthermore, if a lower priority ROM process is currently operating and a higher priority ROM process requires processing unit **54**, then the lower priority ROM process is immediately blocked until the higher priority ROM process has completed its operations.

Referring now to FIG. **8**, a block diagram of the preferred embodiment for an exemplary data cell **76** is shown. In the preferred embodiment, data cell manager **95** allocates a structure and then builds a separate data cell **76** to correspond with each set of captured image data. An exemplary data cell **76** typically includes data cell elements **800** through **834**, however various additional data cell elements may alternatively be included by data cell manager **95**.

Version number **800** indicates which version of data cell **76** is presently in use, so that data cell manager **95** can work with more than one version of data cell **76**. Verification constant **802** is a known constant value used as a check by data cell manager **95** to verify the validity of the data comprising data cell **76**. Image name **804** identifies the particular captured image which corresponds to data cell **76**. Image name **804** is of the preferred form "IMXXXXXX," where "XXXXXX" is the image number. In the preferred embodiment, the image number "XXXXXX" is not reset, so when images are down-loaded to a host computer, the down-loaded image names **804** will not conflict with image names **804** of image files previously downloaded to the host computer. However, in an alternate embodiment the image number "XXXXXX" could be reset each time image data is down-loaded from camera **10**. Also the "IM" in the image identifier may be replaced with "IO."

Image type **806** specifies the format of the captured image. The image type **806** is typically in the form "YYY," is preferably either CFA or JPG which both refer to sets of compressed image data. Image size **808** includes information about the height and width (number of pixels) of the corresponding captured image. Image processing and compression (IPC) **88** uses image size **808** to correctly process a given set of captured image data. User tags **810** include a variety of tags which may be set by a camera **10** user. For example, a user may set a specific user tag **810** to identify whether a particular captured image is a time-lapse image.

Folder name **812** contains the name of the specific folder in which a particular captured image file resides. Image status flags **814** contain information about how much image

processing has been performed on the corresponding image data (e.g., whether the image data is raw or compressed data, and whether the image data is in frame buffer **70**, RAM **60**, temporary buffer **81** or flash disk **64**). The image status flags **814** store either a "Raw Image Data In Frame Buffer" flag, a "Raw Image Data In RAM Disk" flag, a "Compressed Image Data In RAM Disk" flag, a "Raw Image Data In Flash memory" flag, a "Compressed Image Data In Flash memory" flag, or a "Compressed Image Data In Temporary Buffer" flag. Background processing stage **816** indicates the current stage of the image data within the background spooling processes **84** through **92** (FIG. **7**).

Time/date stamp **818** contains data identifying the time and date that the image data was captured by camera **10**. If selected, delete request **820** causes the corresponding image file to be deleted from camera **10**. If selected, stop-processing request **822** causes camera **10** to complete the current process and then to temporarily suspend further processing of the corresponding image data. Watermark data **824** selects a particular watermark image and also specifies where the selected watermark is placed on the captured image.

Image processing (IP) parameters **826** contain information which IPC **88** uses during processing of the corresponding image data. For example, IP parameters **826** may include the compression level and color depth for a particular captured image. Image-capture settings **828** may include the various camera **10** settings which existed when the corresponding image data was captured. For example, image-capture settings **828** may include camera **10** focus values, shutter speed, aperture, white-balance settings and exposure values.

Image data pointer **830** is a pointer to identify the location of the captured image data which corresponds to data cell **76**. Error code **832** stores information to indicate whether processing of the image data was successful for each of the background spooling processes **84** through **92**. Miscellaneous **834** contains a variety of "housekeeping" information used by data cell manager **95** to control and coordinate the function of exemplary data cell **76**.

Referring now to FIG. 9A, the initial portion of a flowchart showing the preferred operation of the present invention is shown. Initially, camera **10** captures **910** a selected image and stores the captured image data into frame buffer **70**. Data cell manager **95** responsively builds **912** a data cell **76** in working memory **72** as described above in conjunction with FIG. **8**. Data cell manager **95** then adds **913** the data cell **76** to a data cell list identifying data cells **76** for all captured images. To subsequently access a given captured image, data cell manager **95** typically identifies the data cell **76** for the given image and then locates the corresponding image data file. Next, data cell manager **95** generates a pointer to the location of data cell **76** in working memory **72**, and then passes **914** the generated data cell **76** pointer to RAM spooler **1** (**84**) by placing the data cell **76** pointer in the RAM spooler **1** (**84**) input queue **78**(*b*).

RAM spooler **1** (**84**) then copies **916** the captured image data from frame buffer **70** to RAM disk **74** to create an image data file. Next, data cell manager **95** makes **918** a copy of the data cell **76** located in working memory **72** and places the data cell **76** copy into the newly-created image data file on RAM disk **74**. RAM spooler **1** (**84**) then deletes **920** the image data from frame buffer **70**.

Next, data cell manager passes **922** the generated data cell **76** pointer to flash spooler **1** (**86**) by placing the data cell **76** pointer in the flash spooler **1** (**86**) input queue **78**(*c*). Flash spooler **1** (**86**) then copies **924** the image data file from RAM disk **74** to flash disk **64** and deletes **925** the image data file from RAM disk **74**. Next, data cell manager **95** passes **926** the data cell **76** pointer to image processing/compression (IPC) **88** by placing the data cell **76** pointer in the IPC **88** input queue **78**(*d*). IPC **88** then accesses **928** the image data stored in flash disk **64** to begin the processing and compressing operations. The FIG. **9**A method then proceeds to FIG. **9**B.

Referring now to FIG. **9**B, the final portion of a flowchart showing the operation of the present invention is shown. Continuing the process steps of FIG. **9**A, data cell manager **95** passes **930** to IPC **88** a number of pointers for locating specified data cell **76** elements. In the preferred embodiment, the specified data cell **76** elements typically may include IP parameters **826**, image size **808**, watermark data **824** and image-capture settings **828**. IPC **88** responsively uses these received pointers to locate and access those specified data cell **76** elements which are needed to effectively process and compress the captured image data.

IPC **88** then advantageously uses this information accessed from data cell **76** to process and compress **932** the captured image data. Next, IPC **88** stores **934** the processed and compressed image data into an image data file in RAM disk **74**, if space is available. If no space is currently available in RAM disk **74**, IPC **88** temporarily stores the compressed image data into another available memory location, such as temporary buffer **81** in working memory **72**. In step **934**, data cell manager **95** also stores selected data cell **76** elements into the compressed image data file created and stored by IPC **88**. Data cell manager **95** thus modifies the copy of data cell **76** that was placed into the raw image data file during step **918**. In the preferred embodiment, the selected data cell **76** elements which data cell manager **95** incorporates into the compressed image data file typically may include image name **804**, image type **806**, image size **808**, user tags **810**, folder name **812**, time/date stamp **818**, IP parameters **826**, watermark data **824** and image-capture settings **828**.

Data cell manager **95** then deletes **936** unnecessary elements from data cell **76** in working memory **72** to conserve storage space within DRAM **60**. The deleted elements have become unnecessary since IPC **88** has already used them to successfully complete the processing and compression operations and since any other relevant elements have been stored in the compressed image file. In the preferred embodiment, the unnecessary data cell **76** elements deleted from data cell **76** in working memory **72** typically include IP parameters **826**, image size **808**, watermark data **824** and image-capture settings **828**. In alternate embodiments, the deleted data cell **76** elements may further include image status flags **814**, background processing stage **816**, time/date stamp **818**, delete request **820** and stop-processing request **822**.

Next, IPC **88** deletes **938** the raw image data file from flash disk **64**, including the copy of the data cell **76** which data cell manager **95** placed into the raw image data file during step **918**. Data cell manager then passes **940** the generated data cell **76** pointer to RAM spooler **2** (**90**) by placing the data cell **76** pointer in the RAM spooler **2** (**90**) input queue **78**(*e*) RAM spooler **2** (**90**) then copies **942** the compressed image data file to RAM disk **74** if IPC **88** was unable to store the compressed image data file to RAM disk **74** in **17** step **934** above.

Data cell manager then passes **944** the generated data cell **76** pointer to flash spooler **2** (**92**) by placing the data cell **76** pointer in the flash spooler **2** (**92**) input queue **78**(*f*). Flash spooler **2** (**92**) then copies **946** the compressed image data file to flash disk **64** and deletes **948** the compressed image data file from flash disk **64**.

FIGS. **9**A and **9**B illustrate the preferred operation of the present invention using a single captured image and corresponding data cell **76**. The present invention, however, typically operates to capture, process and store a series of captured images. Using multi-tasking and task-priority techniques, the present invention may effectively handle multiple captured images at various processing stages within camera **110**. Therefore, the process steps of FIGS. **9**A and **9**B may advantageously be repeated for each captured image in accordance with the present invention.

Referring now to FIG. **10**, a flowchart showing preferred method steps for using the present invention to recover from a disruptive event within camera **10** is shown. Disruptive events may comprise a variety of occurrences which endanger captured image data within camera **10**, including a power failure within camera **10** or removal of flash disk **64** while a captured image is being processed.

Following a particular disruptive event, a camera **10** user initially remedies the disrupting factor and then applies **1010** power to camera **10**. Data cell manager **95** then determines **1012** whether an image data file was stored in RAM disk **74** or flash disk **64** prior to the intervening disruptive event. In the event of a power failure within camera **10**, image data files on RAM disk **74** are protected through the use of backup batteries.

If no image data files were present in RAM disk **74** or flash disk **64** prior to the intervening disruptive event, then the FIG. **10** process ends. However, if image data files were present in RAM disk **74** or flash disk **64**, then data cell manager **95** determines **1014** whether the image data files contained raw or compressed image data. If the image data file contained compressed image data, then data cell manager **95** accesses and uses **1016** that compressed image data file to rebuild the data cell **76** in working memory **72**.

If, however, the image data file contained raw image data, then data cell manager **95** locates **1020** the copy of the data cell **76** stored within the raw image data file and then uses **1022** that raw image data cell **76** to rebuild the data cell **76** within working memory **72**. After data cell manager **95** rebuilds the current data cell **76** within working memory **72**, then data cell manager **95** determines **1018** whether another image file is present in RAM disk **74** or flash disk **64**. If another image file is present, then the FIG. **10** process returns to step **1014** to rebuild the data cell **76** which corresponds to the additional image file. After data cell manager **95** rebuilds all data cells **76** within working memory **72**, camera **110** may then resume **1024** normal background spooling processes **84** through **92** to process and store the captured image data.

The present invention has been described above with reference to certain preferred embodiments, however those skilled in the art will recognize that various modifications may be provided. Furthermore, while the present invention has been discussed above as applied to digital cameras, those skilled in the art will also recognize that the current apparatus and method may also be applied to various other devices. These and other variations upon the preferred embodiment are provided for by the present invention, which is limited only by the following claims.

What is claimed is:

1. A digital imaging system capable of correlating processing data and image data, said digital imaging system comprising:

a capturing device responsive to an image capture request for receiving image data for a first captured image and first processing data including settings of said capturing device at image capture time;

a manager device coupled to said capturing device for building a data cell containing said first processing data and for linking said data cell to said image data; and a processing device coupled to said capturing device for processing said image data using said first processing data within said data cell, wherein said capturing device is operable for receiving image data for a second captured image using processing data that is different than said first processing data while said processing of said image data for said first captured image is being performed using said first processing data.

2. The digital imaging system of claim **1** wherein said manager device is operable for deleting a selected portion of said processing data from said data cell after said processing device has finished processing said image data.

3. The digital imaging system of claim **2** wherein said processing device is operable for storing said image data into a memory device after said manager device has deleted said selected portion of said processing data from said data cell.

4. The digital imaging system of claim **1** wherein said manager device is operable for making a copy of said data cell and for appending said copy of said data cell to said image data.

5. The digital imaging system of claim **4** wherein said manager device is operable for rebuilding said data cell after a disruptive event using said copy of said data cell appended to said image data.

6. The digital imaging system of claim **1** wherein said processing of said image data includes compressing said image data.

7. A method for correlating processing data and image data in a digital imaging system, said method comprising the steps of:

a) receiving image data for a first captured image and first processing data using a capturing device, wherein said first processing data includes settings of said capturing device at image capture time;

b) building a data cell with a manager device, wherein said data cell contains said first processing data;

c) linking said data cell to said image data;

d) receiving image data for a second captured image and second processing data that is different from said first processing data; and

e) processing said image data for said first captured image using said first processing data within said data cell can be while said capturing device is receiving said image data for said second captured image.

8. The method of claim **7** further comprising step f) of deleting a selected portion of said processing data from said data cell after said step e).

9. The method of claim **8** further comprising the step of storing said image data into a memory device after said step f).

10. The method of claim **7** further comprising the steps of:

making a copy of said data cell; and

appending said copy of said data cell to said image data.

11. The method of claim **10** further comprising the step of rebuilding said data cell after a disruptive event using said copy of said data cell appended to said image data.

12. The method of claim **7** wherein said step e) comprises the step of compressing said image data.

13. A computer-readable medium comprising program instructions for correlating processing data and image data in a digital imaging system, wherein said program instructions, when executed by a computer system coupled to said digital imaging system, cause said digital imaging system to implement the steps of:

a) receiving image data for a first captured image and first processing data using a capturing device, wherein said first processing data includes settings of said capturing device at image capture time;

b) building a data cell with a manager device, wherein said data cell contains said first processing data;

c) linking said data cell to said image data;

d) receiving image data for a second captured image and second processing data that is different from said first processing data; and

e) processing said image data for said first captured image using said first processing data within said data cell while said capturing device is receiving said image data for said second captured image.

14. The computer-readable medium of claim **13** wherein said program instructions, when executed, cause said digital imaging system to implement step f) of deleting a selected portion of said processing data from said data cell after said step e).

15. The computer-readable medium of claim **14** wherein said program instructions, when executed, cause said digital imaging system to implement the step of storing said image data into a memory device after said step f).

16. The computer-readable medium of claim **13** wherein said program instructions, when executed, cause said digital imaging system to implement the steps of:

making a copy of said data cell; and

appending said copy of said data cell to said image data.

17. The computer-readable medium of claim **16** wherein said program instructions, when executed, cause said digital imaging system to implement the step of rebuilding said data cell after a disruptive event using said copy of said data cell appended to said image data.

18. The computer-readable medium of claim **13** wherein said step e) comprises the step of compressing said image data.

* * * * *